

WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego



ROZPRAWA DOKTORSKA

por. mgr inż. Jerzy K. Kowalski

WPLÝW STEROWANIA MINISKOKOWEGO NA CHARAKTERYSTYKI NAPĘDU DYSKRETNEGO

Promotor
płk doc. dr hab. inż. Tadeusz WRÓBEL

WARSZAWA – 1988

WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego

ROZPRAWA DOKTORSKA

por. mgr inż. Jerzy K. Kowalski

WPŁYW STEROWANIA MINISKOKOWEGO NA CHARAKTERYSTYKI NAPĘDU DYSKRETNEGO

Promotor
pik doc. dr hab. inż. Tadeusz WRÓBEL

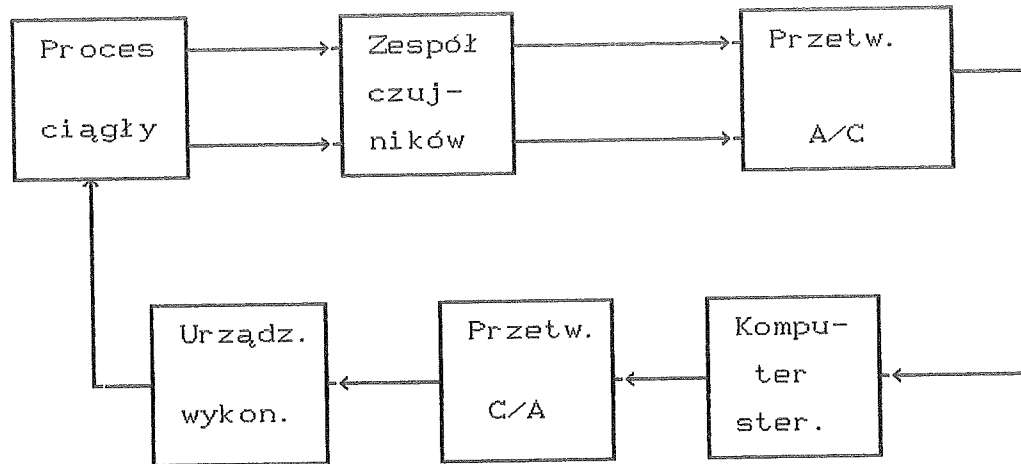
WARSZAWA – 1988

1. GENEZA ORAZ CEL I ZAKRES PRACY

1.1 ROZWÓJ SILNIKÓW SKOKOWYCH

Silniki skokowe w szerokich zastosowaniach przyjęły się w pełni dopiero w połowie naszego stulecia, po wynalezieniu tranzystora w czasie bujnego rozwoju układów półprzewodnikowych. Ekspansja elektroniki i techniki cyfrowej spowodowała powstanie pewnych w działaniu i dostępnych elementów półprzewodnikowych. Uzyskiwanie przebiegów impulsowych o różnych kształtach i dużej mocy nie stanowiło już poważniejszego problemu technicznego. Na tej podstawie zostały opracowane różnorodne elektroniczne komutatory wielofazowe o częstotliwości regulowanej w bardzo szerokich granicach. Jednocześnie zastosowanie komputerów do sterowania różnymi systemami automatyki stworzyło zapotrzebowanie na typowe elektromechaniczne elementy wykonawcze charakteryzujące się dokładnością i niezawodnością pracy przy możliwie niskich kosztach instalacyjnych i eksploatacyjnych. Dotychczas stosowane elementy wykonawcze o działaniu ciągłym i rozbudowanym systemie sprzężeń zwrotnych okazały się nieprzystosowane do współpracy z komputerami. Przetwarzanie sygnałów cyfrowych na analogowe i odwrotnie znacznie komplikuje systemy regulacji, a także wymusza stosowanie drogich i skomplikowanych układów sprzężeń zwrotnych. Na rys. 1.1 został przedstawiony typowy system

regulacji automatycznej sterowany komputerowo.



Rys. 1.1

System regulacji automatycznej sterowany komputerowo

W wielu przypadkach ogromna szybkość pracy komputerów nie mogła być wykorzystana ze względu na niedoskonałość elementów przekształcających sygnały. Klasyczny napęd synchronizowany o działaniu ciągłym nie mógł już zadowolić projektantów i użytkowników układów regulacji automatycznej. Stała się oczywistą potrzeba stworzenia elementów wykonawczych bezpośrednio przetwarzających sygnały cyfrowe z komputera na mechaniczne przemieszczenie proporcjonalne do liczby tych sygnałów. Takimi elementami okazały się silniki skokowe zapewniające przekształcenie sygnałów cyfrowych w dyskretne przesunięcie kątowe lub liniowe. Zwrócono się znów

do tych stosowanych już od drugiej połowy XIX wieku maszyn w zupełnie innym ujęciu. To co początkowo stanowiło wadę silnika skokowego, a mianowicie dyskretny ruch wirnika, zostało wykorzystane jako podstawa działania silnika w nowych układach wykonawczych. Właśnie możliwość kwantowania ruchu oraz ustalenie jednoznaczności między liczbą i częstotliwością impulsów sterujących, a liczbą i szybkością elementarnych przesunięć wału spowodowała, że silnik skokowy stał się przedmiotem badań i poszukiwania nowych rozwiązań. Początkowo prace badawcze związane były z dążeniem do uzyskania skokowo wirującego elektromagnesu, który nie miałby szybko zużywających się części mechanicznych, równoważnego silnikowi z mechanizmem zębatkowym, będącym w tym okresie swego rodzaju wzorem. Ze względu na łatwe sterowanie i niechęć konstruktorów do rozbudowy komutatorów elektronicznych, długi czas skupiano się na budowie różnych konstrukcji silników jednofazowych sterowanych impulsami unipolarnymi. Nie zdawano sobie jeszcze sprawy, że na tej drodze nie uda się uzyskać ani dużej szybkości działania, ani niezbędnego zapasu stabilności. Wysokie wymagania co do szybkości działania i stabilności stawiane przez współczesną technikę otwartym układom sterowania z silnikami skokowymi, wymusiły opracowanie wielofazowych silników skokowych o lepszych parametrach pracy. Jednym z pierwszych silników tego typu, który znalazł szerokie zastosowanie, był trójstojanowy silnik skokowy z wirnikiem reluktancyjnym. Następnie trójstojanowe silniki skokowe zostały zastąpione jednopakietowym silnikiem trójfazowym z uzębionym wirnikiem

reluktancyjnym. Stosowane były dwie główne modyfikacje takiego silnika:

- silniki synchroniczne reluktancyjne z gładkim stojanem;
- silniki synchroniczne reduktorowe o nierównej liczbie otwartych żłobków stojana i zębów wirnika.

Do prac opisujących reluktancyjne silniki skokowe należą [42], [2], [69], [74], [19], [59].

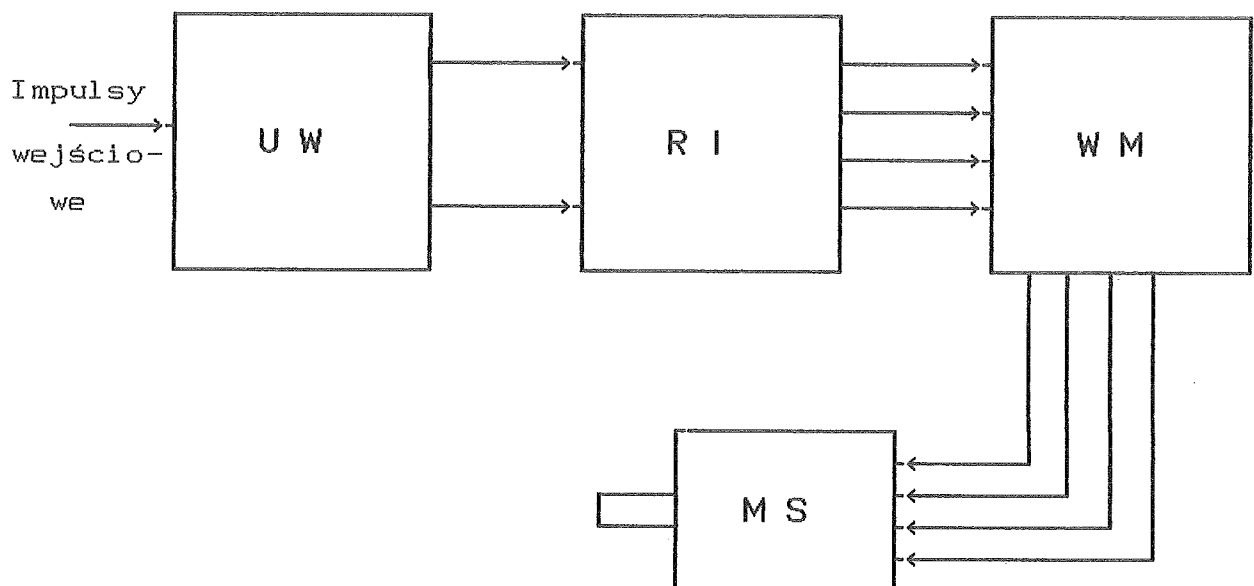
Kolejny etap rozwoju wielobiegunowych silników skokowych wiązał się z konstrukcją czteropasmowych silników reluktancyjnych. Silniki tego typu, przy zasilaniu impulsami unipolarnymi i komutacji cztero- lub ośmiotaktowej osiągają dużą stabilność, znaczną szybkość działania i pozwalają uzyskać duże momenty z jednostki objętości. Dalsze doskonalenie silników skokowych polegało na pojawieniu się silników o magnesach trwałych (o wirniku czynnym) [84], [87], [22], [57], a następnie ,najbardziej doskonałych hybrydowych silników skokowych, które łączą reluktancyjny wirnik z zastosowaniem magnesu trwałego na stojanie lub wirniku [55], [29], [46], [30], [67]. Klasyfikacją i porównaniem różnych typów silników zajęto się w publikacji [123].

W miarę gromadzenia wiedzy, wyników doświadczeń i rozwoju teorii, stało się oczywiste, że każda maszyna elektryczna rozwijająca moment synchronizujący, może być wykorzystana jako silnik skokowy. Zostało dowiedzione, że wszystkie specyficzne cechy silnika skokowego są wynikiem wyłącznie zasilania sygnałami impulsowymi, a nie zależą od

wybranego rozwiązania konstrukcyjnego. Prace nad nowymi konstrukcjami silników skokowych są nieodłącznie związane z rozwojem nowych sposobów i układów sterowania. Układy sterowania, ich właściwości wpływają w zasadniczy sposób na własności napędu dyskretnego, dlatego przy wyborze konstrukcji silnika skokowego i ustaleniu jego optymalnych parametrów powinno się uwzględnić wpływ obciążenia i właściwości układu sterowania.

1.2 NAPĘDY SKOKOWE

Silnik skokowy jest zasadniczym elementem napędów skokowych. Najbardziej rozpowszechnionym jest otwarty napęd skokowy którego struktura pokazana jest na rys. 1.2.



Rys. 1.2. Otwarty napęd skokowy

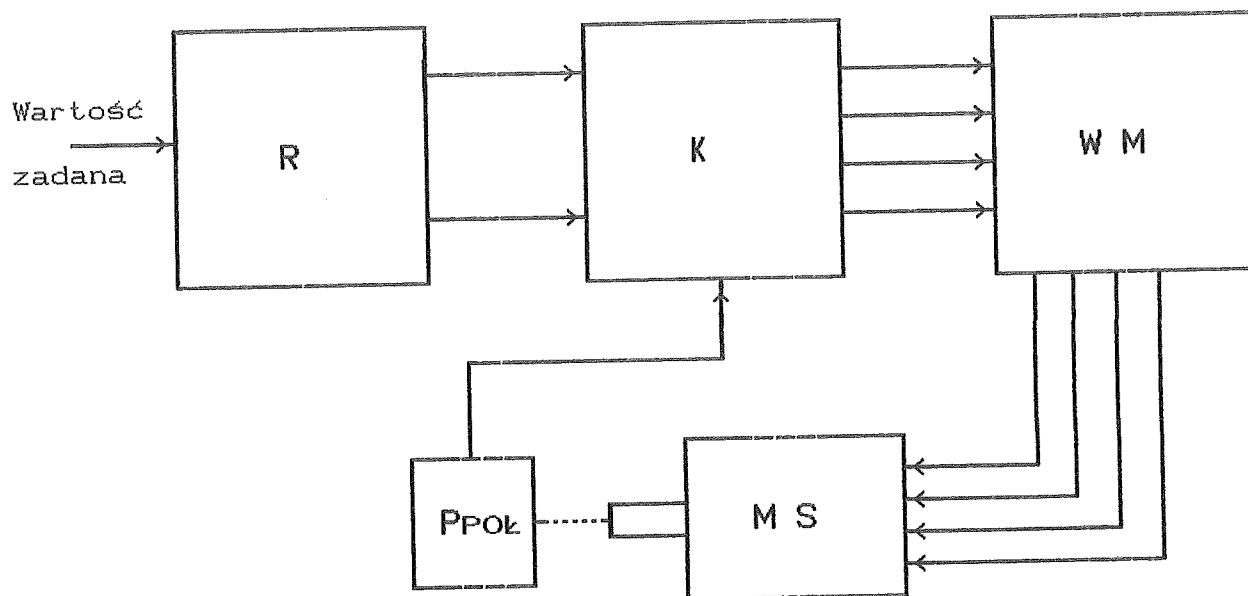
Składa się on z elektronicznego bloku sterowania BS i silnika skokowego MS. Blok przetwarzania przekształca ciąg impulsów wejściowych w wielofazowy układ napięć do zasilania uzwojeń pasmowych silnika skokowego. Blok sterowania składa się zazwyczaj z układu wejściowego UW, elektronicznego rozdzielacza impulsów RI oraz wielokanałowego wzmacniacza mocy WM. Elektroniczny blok sterowania jest zasilany z sieci prądu stałego i pełni rolę statycznego przetwornika częstotliwości. Otwarty napęd skokowy przetwarza informację wejściową zawartą w liczbie i częstotliwości impulsów wejściowych. Zmiana kształtu lub amplitudy impulsów wejściowych w pewnych określonych granicach nie powoduje zakłóceń w normalnej pracy napędu. Prędkość obrotowa i sumaryczne przesunięcie wirnika są proporcjonalne do częstotliwości i liczby impulsów wejściowych. Przy braku impulsów wejściowych następuje przerwanie komutacji i silnik zatrzymuje się rozwijając duży statyczny moment synchronizujący, który umożliwia ustalenie położenia napędu. Dyskretny napęd w układzie otwartym można traktować jako synchroniczno-impulsowy napęd nadażny, który zapewnia szeroki zakres regulacji częstotliwości (od zera) oraz cyfrowe zadawanie drogi z dokładnym ustaleniem współrzędnych końcowych.

W niektórych publikacjach [82], [103], [99], [14] za główną zaletę silnika skokowego uważa się możliwość jego pracy w układzie otwartym, tymczasem istnieje wiele rozwiązań zamkniętych układów sterowania silnikami skokowymi o znacznie

lepszych możliwościach dynamicznych niż w przypadku układu otwartego [71], [26], [49], [3].

W razie potrzeby uzyskania dużych dokładności odwzorowania położenia przy jednoczesnej dużej szybkości działania należy stosować zamknięty napęd skokowy. Schemat funkcjonalny takiego układu napędowego jest przedstawiony na rys. 1.3.

Oprócz omówionych w poprzednim przykładzie elementów występują ponadto bloki dodatkowe - przetwornik położenia wału silnika PP, rejestr zliczający R i przełącznik elektroniczny K. Rejestr zliczający zapamiętuje polecenia przekazane przez komputer sterujący lub urządzenie programujące, zadawane w postaci cyfrowego odwzorowania przyrostu drogi. Maksymalna dopuszczalna niezgodność stanów wejścia i wyjścia w chwili zadania polecenia jest określona pojemnością rejestru.



Rys. 1.3. Zamknięty napęd skokowy

Gdy w rejestrze znajduje się informacja, pętla sprzężenia zwrotnego jest zamknięta i silnik zaczyna obracać się przyspieszając tak jak maszyna prądu stałego. Przetwornik położenia umieszczony na wale silnika spełnia rolę komutatora (w stosunku do silnika) i generatora impulsów (w odniesieniu do rejestru), odpowiadających przesunięciu wału. Po odliczeniu zadanej liczby skoków, otwiera się pętla sprzężenia zwrotnego i silnik zatrzymuje się bez wybiegu. Zamknięty napęd skokowy można sterować bezpośrednio kodem cyfrowym lub podobnie jak w układzie otwartym ciągiem impulsów. Pętla sprzężenia zwrotnego nadaje napędowi zamkniętemu zasadnicze cechy napędu prądu stałego, takie jak: szybkość działania w stanach nieustalonych, dużą sprawność i tłumienie kołysań własnych. Równocześnie zachowane są zalety napędu skokowego - niewystępowanie wybiegu wirnika, tzn. pewne zatrzymywanie się w położeniu końcowym oraz brak kumulacji błędów. W ostatnim okresie coraz szersze zastosowanie znajdują nowe metody sterowania silników skokowych pozwalające znacznie poprawić parametry napędu skokowego. Należą do nich m.in.:

- komutacja niesymetryczna;
- forsowanie impulsowe;
- zasilanie ze źródła prądowego;
- praca start-stopowa;
- praca miniskokowa.

Ostatnia z podanych metod umożliwia znaczne zmniejszenie wartości skoku silników na drodze elektronicznej. Jest to cenna właściwość w wielu zastosowaniach. Opracowanie metod szerszego wykorzystania tych nowych metod sterowania przy zastosowaniu elementów elektronicznych nowych generacji (mikroprocesorów, układów wielkiej skali integracji) pozwoliłoby zwiększyć obszar zastosowania silników skokowych, co dałoby polepszenie parametrów użytkowych nowych urządzeń przy jednoczesnym zwiększeniu niezawodności i obniżce kosztów produkcji i eksploatacji.

13 CEL I ZAKRES PRACY

Zbadanie wpływu zastosowania sterowania miniskokowego na parametry mechaniczne napędów skokowych pozwoliłoby określić zakres zastosowań tej nowej metody sterowania. Ma to duży wpływ na rozwój konstrukcji napędów skokowych i pozwoliłoby w przyszłości określić modyfikacje silników skokowych, ich konstrukcji mechanicznej i elektrycznej, które pozwolą na poprawę parametrów pracy miniskokowej.

Celem pracy jest między innymi zbadanie wpływu sterowania miniskokowego na charakterystyki mechaniczne silnika skokowego, oraz określenie podstawowych wymagań dotyczących sposobów generowania funkcji sterujących i zasilania silnika.

Praca obejmuje rozpoznanie literaturowe w zakresie napędów dyskretnych ze szczególnym uwzględnieniem sterowania miniskokowego, konstrukcję programu symulacji komputerowej

napędu sterowanego miniskokowo, badania eksperymentalne napędu zasilanego z odpowiedniego sterownika miniskokowego (opracowanego i wykonanego przez autora w ramach pracy) oraz analizę wyników i porównanie ich z symulacją komputerową.

W ramach pracy można sformułować tezę, że zastosowanie sterowania miniskokowego umożliwia zwiększenie liczby skoków na obrót wału silnika (ok. 100 razy), zwiększenie maksymalnej częstotliwości pracy napędu, eliminację drgań wirnika i rezonansu mechanicznego bez znaczącego pogorszenia parametrów mechanicznych napędu.

2. STEROWANIE I KONSTRUKCJE NOWOCZESNYCH SILNIKÓW SKOKOWYCH

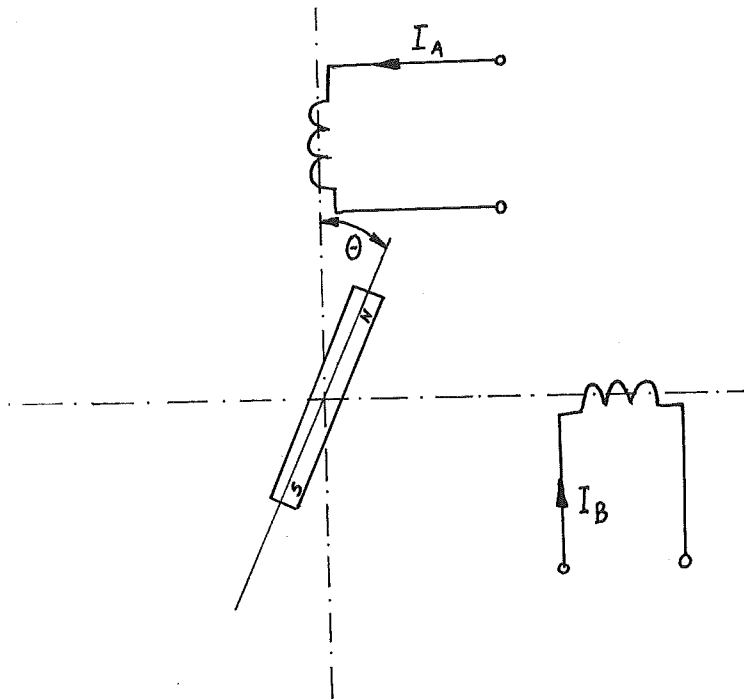
2.1. PRACA MINISKOKOWA SILNIKA SKOKOWEGO

Zastosowanie pracy miniskokowej do silnika skokowego wymaga zastosowania specjalnego układu sterowania umożliwiającego wygenerowanie dyskretnego sygnału sterującego w pełnym zakresie napięć wyjściowych wzmacniacza mocy. Klasyczne układy sterowania silnikami skokowymi opisano w literaturze książkowej dostępnej w Polsce [63], [10], [78] oraz w licznych publikacjach czasopiśmienniczych [90], [33], [11], [23], [113], [36], [5], [95], [86], [91], [89], [104], [8], [106], [79], [110], [9], [6], [7], [75], [31], [52], [1], [41], [97]. Rozwiązania sterowników dla silników skokowych opisane są również w literaturze patentowej [P-1] - [P-13], [P-14] - [P-15], [P-16] - [P-22], [P-23] - [P-26], [P-27] - [P-31], [P-32] - [P-33]. Klasyczny układ sterowania silników skokowych generuje sygnały o dwóch poziomach napięcia lub prądu, odpowiadających maksymalnemu i minimalnemu poziomowi źródła. Przy takim sterowaniu wirnik silnika przyjmuje położenia dyskretne określone przez jego konstrukcję. Istotne dla rozważań zawartych w tej pracy są układy sterowania miniskokowego, będące również tematem wielu prac [18], [125], [73], [21], [107], [92], [111], [37], [34], [85], [96], [93], [70], [12], [15], [119], [P-32].

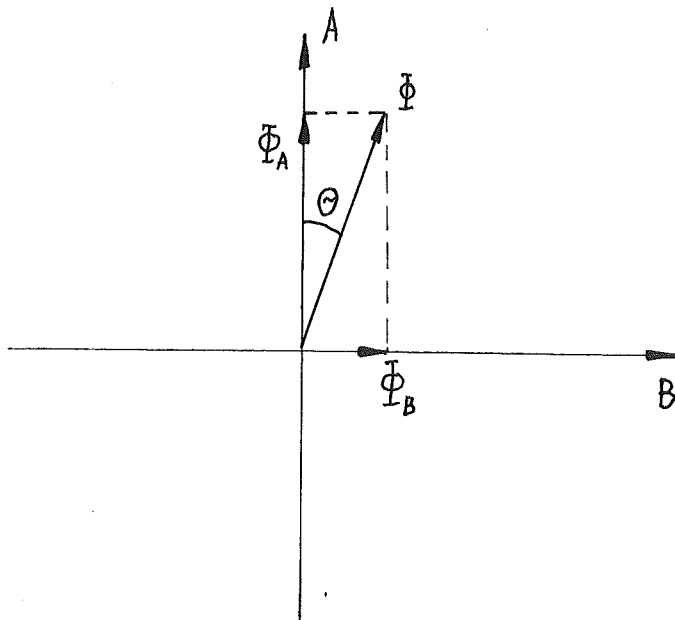
Układ sterowania miniskokowego pozwala uzyskać sygnały o

poziomach pośrednich między maksymalną i minimalną wartością sygnału źródła. Przy zastosowaniu takiego rodzaju wymuszenia prądy płynące w pasmach silnika wytwarzają wektor strumienia magnetycznego którego położenie w przestrzeni określone jest przez wartości tych prądów. Wektor strumienia zajmuje położenia pośrednie pomiędzy pozycjami określonymi przez konstrukcję silnika. Uzyskuje się w ten sposób elektroniczne pomnożenie konstrukcyjnej liczby skoków bazowych przez liczbę miniskoków w skoku bazowym. Ilustracja pracy miniskokowej oraz sygnały zasilające silnik są pokazane odpowiednio na rys. 2.1 i 2.2.

a)



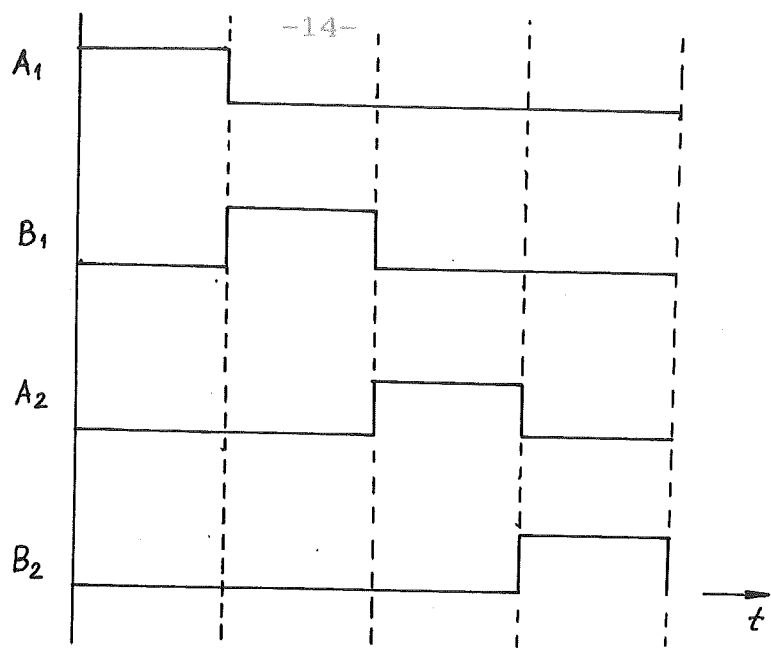
b)



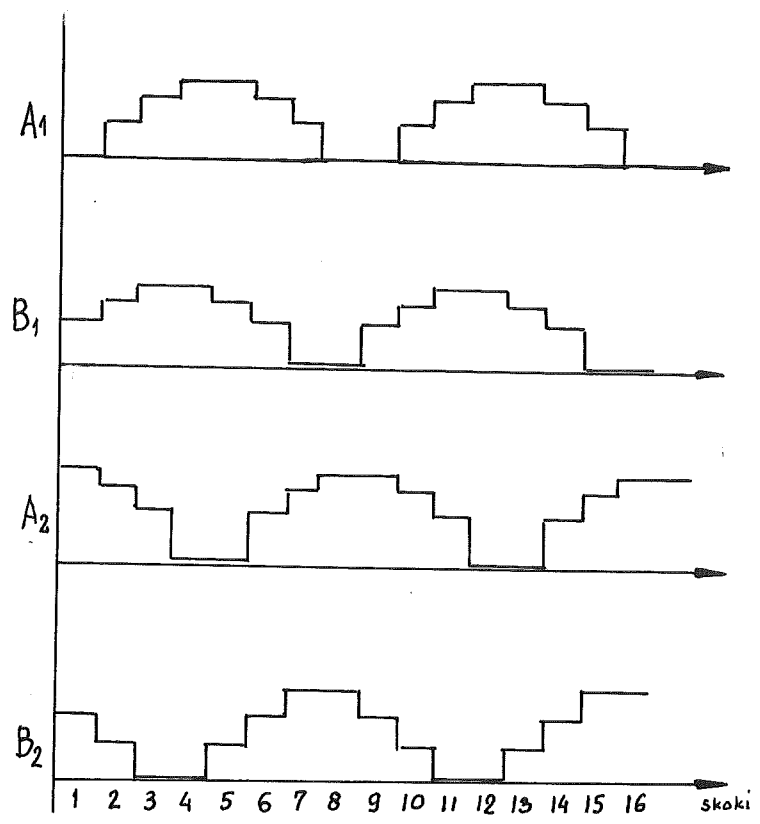
Rys. 2.1. Sterowanie miniskokowe silnika skokowego

- a) jedno z położeń pośrednich wirnika silnika skokowego przy sterowaniu miniskokowym,
- b) wykres wektorowy dla sterowania miniskokowego.

a)



b)



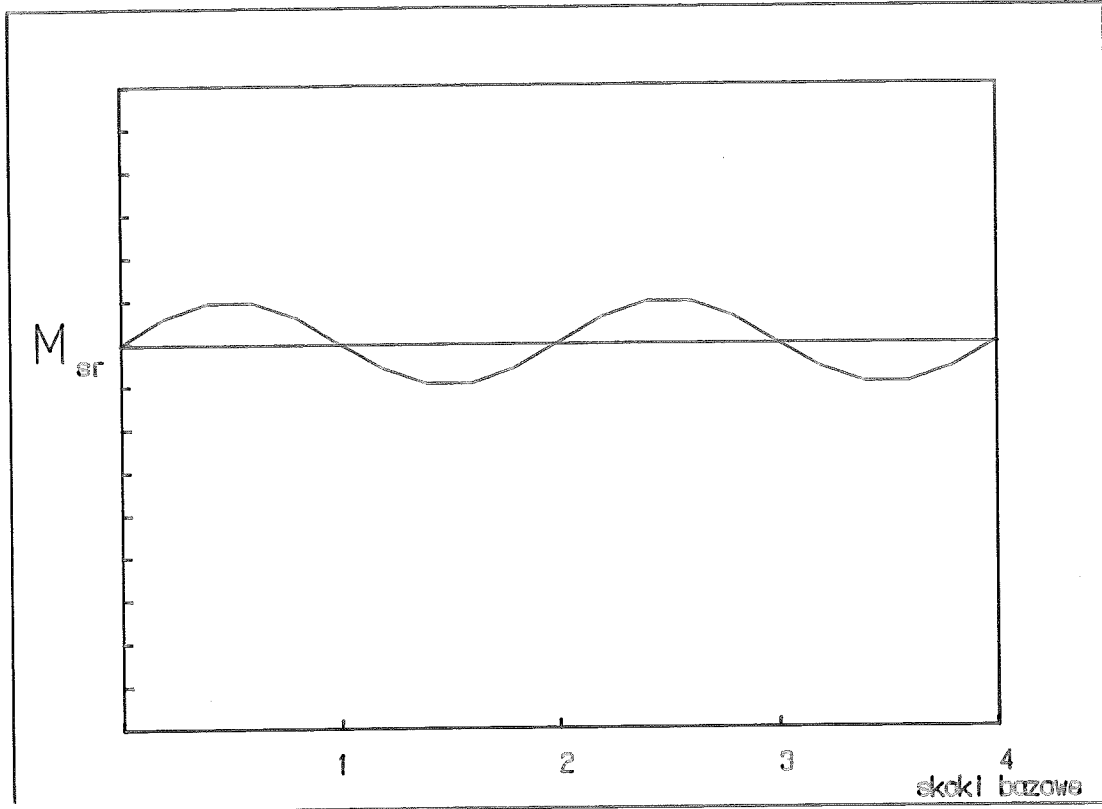
Rys. 2.2 Przebiegi sygnałów zasilających silnik skokowy:
a) sygnały klasycznego sterowania typu 1-2,
b) sygnały sterowania miniskokowego dla podziału N=4.

Przy generowaniu sygnału sterowania miniskokowego niezbędne jest spełnienie następującego warunku:

$$(2.1) \quad \phi_A^2 + \phi_B^2 = \text{const}$$

związane jest to z potrzebą zachowania stałej amplitudy strumienia i w związku z tym stałości maksymalnego momentu synchronizującego w funkcji położenia kąowego wirnika.

W silnikach skokowych reluktancyjnych i hybrydowych występuje niekorzystne zjawisko pulsacji reluktancji obwodu magnetycznego w czasie ruchu wirnika. Przy standardowym sposobie sterowania zjawisko to wpływa na odkształcenie charakterystyki momentu synchronizującego w funkcji kąta obrotu wirnika, a przy sterowaniu miniskokowym powoduje pulsację maksymalnego momentu synchronizującego w ramach jednego skoku bazowego. Bez zastosowania dodatkowych środków zapobiegających zjawisko to powoduje obniżenie maksymalnych momentów obciążenia możliwych do uzyskania w danym silniku. Zjawisko to ilustruje rys. 2.3.



Rys. 2.3. Pulsacja momentu synchronizującego przy pracy miniskokowej.

Do sterowania miniskokowego jest stosowana podstawowa funkcja $\sin(k*\alpha+\theta)$, gdzie k jest numerem miniskoku, a θ przesunięciem fazowym funkcji. Dla silników wielopasmowych stosuje się te funkcje z przesunięciem fazowym zależnym od liczby pasm. Można stosować zasilanie bipolarne lub unipolarne, w zależności od konstrukcji silnika [90], [88], [20], [119].

2.2. KONSTRUKCJE SILNIKÓW SKOKOWYCH

Silniki skokowe ze względu na prostotę konstrukcji, precyzję i łatwość sterowania znajdują coraz szersze zastosowanie w różnych urządzeniach przemysłowych, a w szczególności w sprzęcie komputerowym i robotach przemysłowych, obrabiarkach do dokładnej obróbki metali, w maszynach do pisania i do szycia i innych [121], [111], [92], [76], [12], [112]. Dążąc do zmniejszenia wymiarów silnika przy jednoczesnym poprawieniu jego charakterystyk stosuje się coraz częściej magnesy trwałe bazujące na pierwiastkach ziem rzadkich [122], [54], [100], [24]. Oddzielnym problemem jest doskonalenie układów sterowania i wzmacniaczy mocy. Zastosowanie nowoczesnych elementów półprzewodnikowych oraz mikroprocesorów pozwoliło na wprowadzenie nowego rodzaju sterowania - miniskokowego. Pozwala on na uzyskanie mniejszych skoków przez podzielenie skoku bazowego na określoną liczbę miniskoków.

Typowo stosowane układy miniskokowe pozwalają uzyskać do 125 miniskoków na jeden skok bazowy silnika. Zastosowanie sterowników mikroprocesorowych wymusiło zastosowanie nowych elementów we wzmacniaczach mocy. Wynika to z potrzeby zwiększenia prędkości pracy napędów skokowych. Do wzmacniaczy mocy wprowadza się nowoczesne tranzystory mocy typu MOSFET oraz scalone układy mocy.

W pracach [16], [83], [25] omówiono rozwój silników skokowych w obliczu gwałtownego postępu w dziedzinie nowych materiałów i technologii. Zastosowanie pierwiastków ziem

rzadkich pozwoliło na wielokrotną poprawę parametrów materiałów magnetycznie twardych stosowanych w silnikach skokowych. Na przykład dla stopu SmCo_5 mogą wynosić: $B_r=0.92$ T, $H_c=705$ kA/m, $(BH)_{\max}=167$ kJ/m³. Ostatnie osiągnięcia dotyczą zastosowania materiałów z dodatkiem neodymu-żelaza-boru (NdFeB) o $B_r=1.2$ T, $H_c=860$ kA/m; $(BH)_{\max}=240$ kJ/m³ [80], [81]. Wykazują one jak widać znacznie większe natężenie pola magnetycznego niż magnesy samarowo-kobaltowe. Wykorzystanie magnesów bazujących na neodymie w silnikach o wartości skoku 0.9° pozwoliło na uzyskanie dużej dokładności przy 400 skokach na obrót.

Te "półskokowe" silniki mają taką samą dokładność pozycjonowania jak popularne silniki 1.8° , które mają tylko 200 skoków na obrót. Na rys. 2.4. przedstawiono porównanie wirników różnych rodzajów silników. Oszczędności w objętości maszyny przy tych samych parametrach mechanicznych są znaczące.

Przykładowo można podać parametry silnika D410 firmy Portescap, w którym zastosowano magnes neodymowy. Przy wymiarach zewnętrznych $10.16 \times \phi 40.64$ mm silnik rozwija statyczny moment synchronizujący 91.83 mNm. Dla porównania silnik z konwencjonalnymi materiałami magnetycznymi przy tym samym momencie synchronizującym ma długość 50.8 mm. Wirnik silnika D410 składa się ze 100 par biegunów. Ze względu na małe wymiary wirnika ma on niewielki moment bezwładności i w związku z tym silnik D410 jest znacznie szybszy od silników konwencjonalnych.

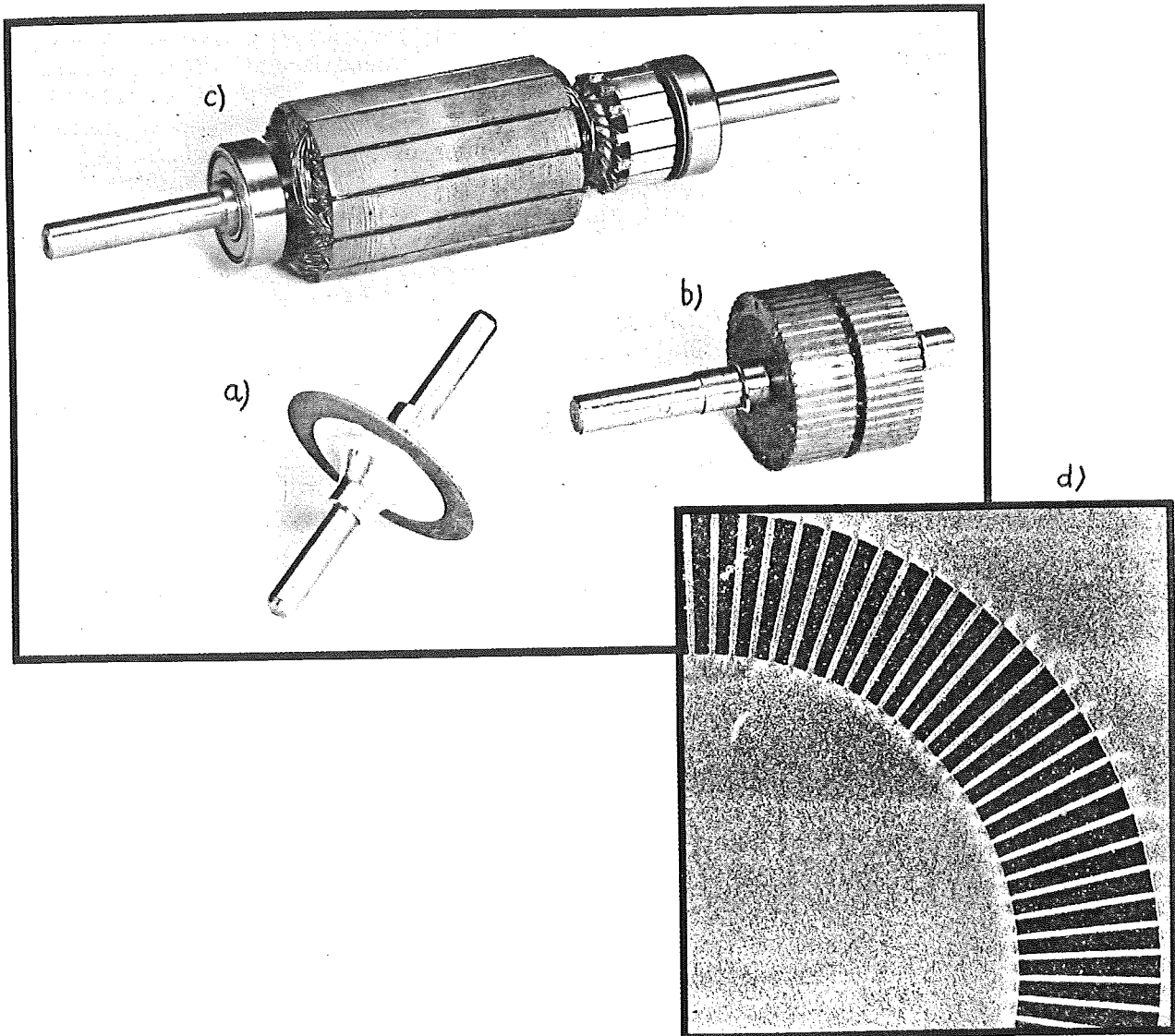
Popularne silniki tarczowe oparte są o magnesy

samarowo-kobaltowe. Silnik PMI firmy Koolmorgen z serii USS-60 ma moment bezwładności $2.4 \cdot 10^{-6}$ Nms² i pozwala osiągnąć przyspieszenie kątowe do 12000 rad/s². W wersji 3.6° ma on następujące wymiary: długość 38.1 mm, średnica 63.5 mm i rozwija statyczny moment synchronizujący 282.6 mNm. Przy zastosowaniu sterowania miniskokowego można podzielić skok bazowy na 100 miniskoków.

Informacje dotyczące nowoczesnych silników skokowych produkowanych m.in. w firmie Texas Instruments Co. można znaleźć w [105].

Ocenia się (wg. Sigma Instruments), że silniki o magnesach trwałych z pierwiastkami ziem rzadkich pozwalają osiągnąć do 50-60% większy moment w porównaniu z silnikami hybrydowymi dotychczas produkowanymi.

Silniki serii 802 tej firmy mają obwód magnetyczny z magnesami ziem rzadkich i alnico. Silnik taki przy średnicy zewnętrznej 86.36 mm i kącie skoku 1.8° rozwija moment synchronizujący 6.357 Nm. Należy zaznaczyć, że dotychczas stosowane wzmacniacze i sterowniki mogą być z powodzeniem zastosowane do nowych silników.



Rys 2.4 Porównanie wirników różnych rodzajów silników
a) wirnik silnika skokowego tarczowego o magnesach z
pierwiastkami ziem rzadkich;
b) wirnik silnika hybrydowego;
c) wirnik silnika prądu stałego;
d) powiększenie wycinka wirnika tarczowego 3.6° ;

Zastosowanie sterowania miniskokowego wraz z małym momentem bezwładności wirnika pozwoliło w znacznym stopniu wyeliminować jego drgania. Jest to dodatkowa korzyść wynikająca z zastosowania nowych materiałów i technologii.

Rozwój zaznaczył się także w konstrukcjach układów sterowania i wzmacniaczy mocy. Opracowany przez firmę Compumotor szereg układów sterowania, dla standardowych silników 1.8° i 125 miniskoków na skok bazowy, pozwala uzyskać 25000 skoków na obrót. Układ sterowania reguluje prąd w pasmach przez modulację szerokości impulsów. Układ pozwala zatrzymać wirnik w każdym ze 125 punktów pomiędzy położeniami bazowymi. Seria M wzmacniaczy mocy zapewnia uzyskanie z odpowiednimi silnikami momentów statycznych od 0.176 do 4.95 Nm przy maksymalnej prędkości obrotowej do 1200 obr/min.

Model 2900 firmy Bodine ma specjalne układy do redukcji drgań. Inne parametry są podobne do wyżej opisanych.

Najnowsze sterowniki konstruowane są w oparciu o specjalistyczne systemy mikroprocesorowe z odpowiednimi programami pracującymi w czasie rzeczywistym. W firmie CyberPark opracowano system modułowy oparty o procesor jednoukładowy Z8; zawiera on szynę sterującą 26 liniową i może sterować do 16 silnikami z lokalnymi sterownikami mocy.

Zastosowanie nowoczesnych elementów elektronicznych znacznie zwiększyło możliwości napędów skokowych i ich efektywność.

Nowoczesne urządzenia produkcyjne w przemyśle półprzewodnikowym wymagają stosowania nowych napędów

skokowych o skoku dochodzącym do 0.0072 stopnia (50000 skoków/obrót). W artykule [29] zostały opisane nowe urządzenia napędów skokowych realizujących takie parametry. Seria 600 serwonapędów firmy Gould Motion Control Div. jest nową generacją napędów skokowych. System ten oparty jest na technice mikroprocesorowej co pozwala swobodnie programować funkcje i charakterystyki napędu. Silnik w tym napędzie może rozwijać maksymalną prędkość obrotową do 3600 obr/min przy maksymalnym momencie obrotowym 0.39 [Nm] oraz impulsowym do 1.15 [Nm]. Wszystkie parametry napędu można programować przy pomocy łącza RS-232C. Układ wyjściowy pracuje z częstotliwością 20 kHz przy zastosowaniu tranzystorów MOSFET w układzie choppera. W ramach tej serii produkowane są różne szeregi sterowników:

- szereg I - zapewnia podstawowe parametry przy najwyższym momencie obrotowym, prędkości i przyspieszeniu;
- szereg LM - liniowych napędów miniskokowych o dokładności 472 sk./mm;
- szereg M - obrotowych napędów miniskokowych o dokładności 50000 sk./obr;
- szereg DH - obrotowych napędów miniskokowych, 5000 sk./obr przy momencie 0.29 [Nm] i częstotliwości przełączania 20 kHz.

W pracy [60] zostały opisane podstawowe parametry najczęściej stosowanych silników skokowych produkcji japońskiej. W ostatnich latach nastąpił gwałtowny wzrost produkcji silników skokowych i w roku 1982 wyprodukowano w Japonii 30 milionów silników. Produkowane są silniki w trzech grupach: reluktancyjne, o magnesach trwałych i hybrydowe. Główne zastosowanie znajdują w drukarkach, stacjach dysków elastycznych i twardych. W tabelach 2.1, 2.2 zostały przedstawione parametry typowych silników o magnesach trwałych i hybrydowych.

Do sterowania takich silników stosuje się coraz częściej specjalizowane układy scalone wielkiej skali integracji, które realizują skomplikowane algorytmy sterowania silnika. Zastosowanie sterowania miniskokowego pozwala na uzyskanie dokładności pozycjonowania w zakresie jednego skoku do $0.5 \mu\text{m}$ lub $1/400$ części skoku podstawowego. Zastosowanie sterowania miniskokowego pozwala także znacznie zmniejszyć drgania wirnika po zatrzymaniu.

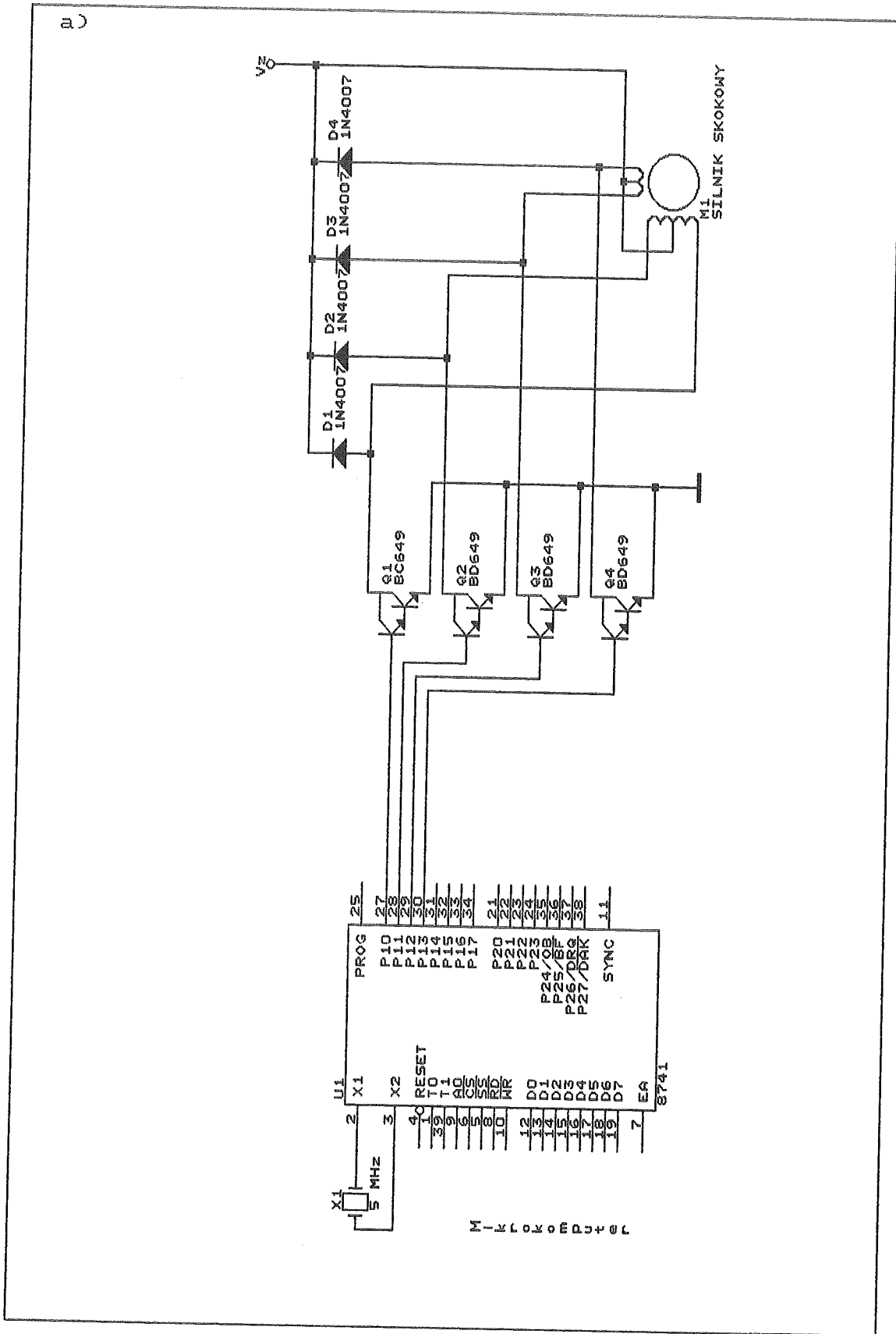
TABELA 2.1. Typowe silniki skokowe o magnesach trwałych
produkcji japońskiej

średnica zewnętrzna [mm]	długość [mm]	wartość skoku [deg]	moment statyczny [mNm]
20	18.3	18	2.3
25	16	15, 18	6 - 11
35	15 - 22	7.5, 15	11 - 18
40	22	7.5, 10, 15, 18	19 - 35
42	15	7.5, 15	15 - 75
51	25	7.5, 15	25 - 30
55	25	7.5, 15	83 - 100
60	43	7.5, 15	150 - 260
65	13.5	1.8	45 - 75

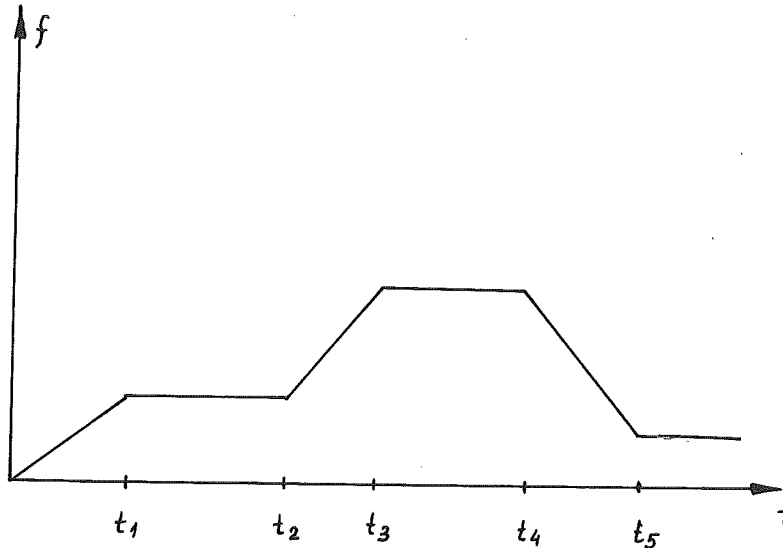
TABELA 2.2. Typowe silniki hybrydowe produkcji japońskiej

średnica zewnętrzna [mm]	długość [mm]	wartość skoku [deg]	moment statyczny [Nm]
50	41	1.8	0.15
57	38 - 102	1.8	0.17-1.1
	39 - 51	0.9	0.18-0.38
	38 - 83	0.72	0.21-0.8
	38 - 83	0.45	0.31-1.1
86	62 - 135	1.8	1.27-3.5
		0.72	1.27-4.0
		0.45	1.60-4.5
106	120 - 240	1.8	3.50-15
110	197	0.72	7.5
106	120 - 240	0.45	4.10-15
39	26	3.6	0.04
	26 - 35	1.8	0.04-0.06
	30	0.9	0.051
42	33	3.6	0.05
	35 - 39	1.8	0.095-0.17
	33	0.9	0.08
46	10 - 12	0.9, 1.8	0.024-0.03
57	13.5	0.9, 1.8	0.072

Wprowadzenie do produkcji tanich jednostrukturalnych mikroprocesorów pozwoliło zastosować je w inteligentnych sterownikach silników skokowych. Takie zastosowanie mikroprocesora Intel 8741A zostało przedstawione w pracy [102]. W przedstawionym układzie mikroprocesor jednostrukturalny jest sterowany przez mikrokomputer i realizuje funkcje sterownika silnika skokowego czteropasmowego. Jedynymi elementami poza mikroprocesorem są tranzystory wzmacniacza mocy, sterowane bezpośrednio przez wyjścia mikroprocesora. Zapisany w pamięci EPROM program sterujący pozwala realizować różne charakterystyki sterowania. Na rys. 2.5.b pokazana jest przykładowa charakterystyka rozpędzania i hamowania silnika. Układ sterownika jest przedstawiony na rys. 2.5.a. Zagadnienia związane ze sterowaniem silnikami skokowymi z zastosowaniem mikroprocesorów poruszono również w pracach [50], [118], [44], [126], [64], [56], [108], [109], [114], [116], [66], [27], [53], [76], [94], [117], [119], [120].



b)



Rys.2.5 Napęd z silnikiem skokowym, sterowany mikroprocesorem
a) Układ sterownika;
b) Przykładowa charakterystyka napędu.

W pracy [62] została opisana nowa seria silników skokowych typu MH produkowanych przez firmę Superior Electric. Silniki te przy pracy półskokowej pozwalają osiągnąć do 20000 imp/s. Uzyskano ten parametr przez optymalizację struktury obwodu magnetycznego w celu redukcji strat. Do sterowania tych silników zaprojektowano serię sterowników USS75 Syncrostep. I tak układ USS75M zawiera układy optymalizacyjne do generacji miniskoków. Specjalny sposób zasilania pasm silnika pozwolił na uzyskanie

sinusoidalnego rozkładu momentu synchronizującego w funkcji kąta (w obrębie jednego skoku). Typowy skok bazowy silnika współpracującego z tym układem wynosi 1.8° , dokładność pozycjonowania 5% i moment synchronizujący 0.7 Nm. W takich silnikach można osiągnąć duże przyspieszenie kątowe do 94000 rad/s^2 przy momencie bezwładności wirnika $12.35 \cdot 10^{-6} \text{ Nms}^2$. Ciągła moc wyjściowa wynosi 80 W, a impulsowy wydatek mocy do 110 kW/s. Produkowane są też napędy skokowe z tej serii o większych momentach wyjściowych jak np.: MH112-FJ8020 gdzie moment synchronizujący wynosi 14 Nm przy zasilaniu jednopasmowym i 18 Nm przy zasilaniu dwupasmowym. Napędy te pozwalają na pracę start-stopową do częstotliwości 4000 Hz przy momencie obciążenia do 11 Nm. Podobne parametry czasowe mają inne napędy serii MH172-FD8030, FD8040 przy momencie obciążenia do 31 Nm. Dla wszystkich napędów dokładność pozycjonowania wynosi minimum $\pm 2\%$.

Problemy pracy miniskokowej i eliminacji drgań poruszane są w pracach [62], [29], [35], [16], [60]. Porównano w nich dotychczas stosowane układy sterowania oraz układy sterowania półskokowego z nowoczesnymi mikroprocesorowymi układami sterowania miniskokowego. W produkowanych obecnie układach sterowania stosuje się zazwyczaj do 125 miniskoków na skok bazowy silnika. Poprzez zastosowanie takiego sterowania wyeliminowano problemy drgań rezonansowych występujących zazwyczaj w zakresie 150-250 Hz.

2.3. SILNIKI SKOKOWE, A INNE RODZAJE SILNIKÓW WYKONAWCZYCH

Porównanie silników hybrydowych oraz silników prądu stałego pozwala stwierdzić że silniki skokowe charakteryzują się podobnymi parametrami dynamicznymi przy nieco mniejszym wydatku mocy z danej objętości lecz pozwalają uzyskać znacznie większą dokładność przy mniej skomplikowanych układach sterowania. Przykładowe zestawienie parametrów napędów skokowych i prądu stałego jest przedstawione w tabeli 2.3.

Za główne zalety zastosowania silników skokowych w napędach, w porównaniu z serwonapędami prądu stałego niektórzy autorzy uważają:

- możliwość niewystępowania sprzężenia zwrotnego;
- możliwość realizowania bardzo małych przemieszczeń (1 skok na dobę) bez stosowania przekładni mechanicznej;
- możliwość synchronizacji wirowania kilku silników skokowych;
- dużą niezawodność;
- stosunkowo niską cenę [99], [14].

Porównanie napędów z silnikami skokowymi z innymi rodzajami napędów elektrycznych zajęto się również w pracach [51], [4], [124].

Silniki prądu stałego (DC) zamieszczone w powyższej tabeli wyposażone są w układy sprzężenia zwrotnego, a silniki skokowe pracowały w układzie otwartym.

Porównywanie silników skokowych z silnikami indukcyjnymi i synchronicznymi nie ma dużego znaczenia ze względu na

całkowicie odmienny obszar zastosowań tych maszyn, choć zdarzają się prace na ten temat [103], [28].

TABELA 2.3. Przykładowe zestawienie napędów skokowych i prądu stałego

Typ silnika	Wymiary [mm]	Liczba pasm	Kąt skoku [deg]	Metoda sterowania	Wydatek mocy [kW/s]	Max. prędk. [1/min]	Moment obr. [N m]
Silnik skokowy hybrydowy	57x51	4	0.9	1-2	1.8	2250	0.14
		10	0.72	4	2.1	4800	0.71
	86x62	4	0.9	1-2	3.0	1930	0.4
		10	0.72	4	3.3	2280	0.42
	86x94	4	0.9	1-2	4.5	1950	0.72
		10	0.72	4	4.5	1440	0.73
Silnik DC z magnesem ferrytowym	50x103				0.7	3000	0.15
	76x125				1.68	3000	0.6
	86x129				2.2	3000	0.83
Silnik DC z magnesem ziem rzadkich	50x78				2.2	3000	0.2
	76x103				3.4	3000	0.6
	86x128				5.7	2500	1.6
Silnik DC bezszczotkowy	52x117				2.9	3000	0.16
	76x123				8.4	3000	0.7

3. MODELE MATEMATYCZNE SILNIKÓW SKOKOWYCH

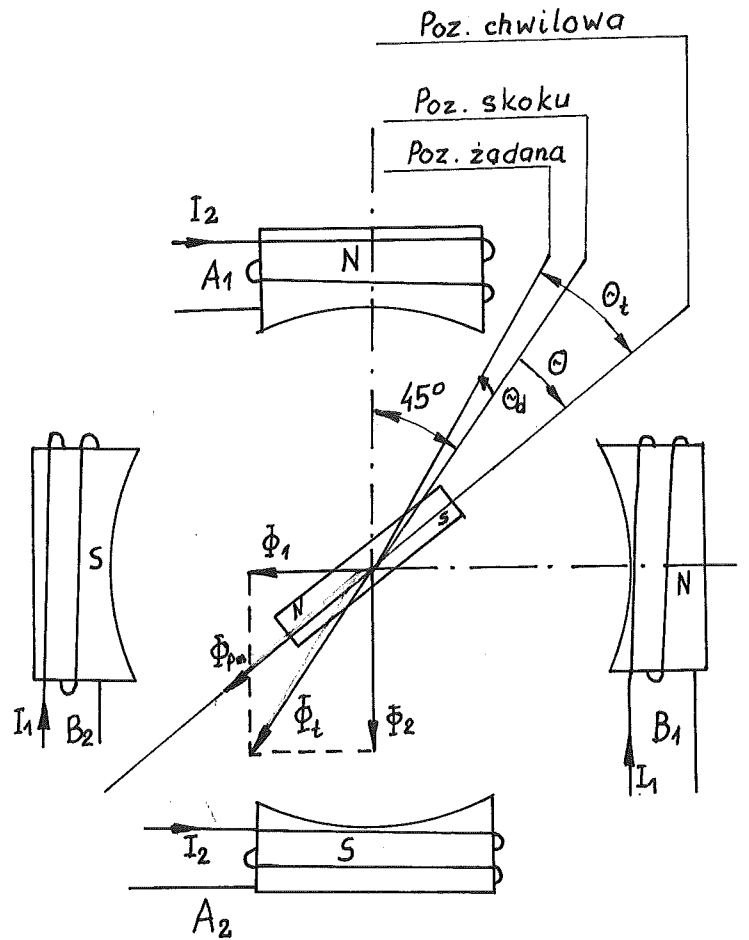
W literaturze można się spotkać z wieloma opisami modeli silników skokowych. Są rozpatrywane modele różnych odmian i konstrukcji silników skokowych. Spotykane są opisy modeli silników skokowych z magnesami trwałymi [20], [61], [65], [84], silników reluktancyjnych [40], [77], [68], [45] oraz silników hybrydowych [63], [43], [17], [115] [48]. Poniżej zostaną przedstawione wybrane opisy modeli silników skokowych, które ilustrują różne podejścia do modelowania tych maszyn.

W pracy [20] został przedstawiony model matematyczny silnika skokowego. Opisany model dotyczy silnika skokowego czterobiegunowego, dwupasmowego, z wirnikiem o magnesach trwałych i kącie skoku równym 90 stopni. Został on przedstawiony na rys. 3.1, gdzie zostały użyte następujące oznaczenia:

- θ - kąt obrotu wirnika względem pozycji odniesienia [rad];
- θ_d - błąd pozycjonowania [rad];
- I_N - nominalny prąd zasilania pasma silnika [A];
- I_1 - prąd w pasmie B [A];
- I_2 - prąd w pasmie A [A];
- Φ_1 - wektor strumienia wytwarzanego przez prąd I_1 w pasmie B [Wb];
- Φ_2 - wektor strumienia wytwarzanego przez prąd I_2 w pasmie A [Wb];

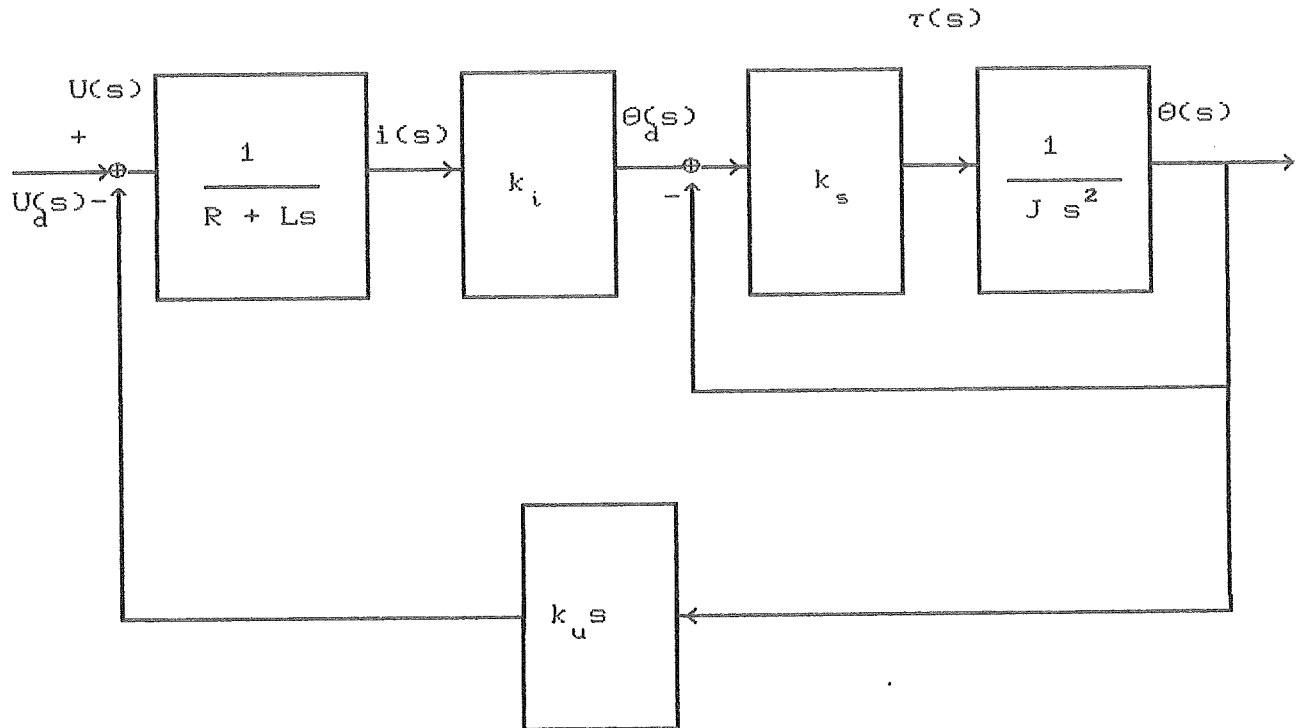
$\Phi_t = \Phi_1 + \Phi_2$ - wypadkowy wektor strumienia [Wb];

Φ_m - wektor strumienia magnesu trwałego [Wb].



Rys. 3.1. Model analogowy silnika skokowego

Opierając się na modelu obwodowym silnika oraz przekształceniu Laplace'a równań silnika utworzono schemat blokowy przedstawiony na rys. 3.2.



Rys. 3.2. Schemat blokowy modelu silnika skokowego

Transformata Laplace'a napięcia zasilającego ma postać:

$$(3.1) \quad U(s) = U_d(s) - U_u(s)$$

gdzie:

$U_d(s)$ - transformata napięcia różnicowego zasilającego pasma silnika;

$U_u(s)$ - transformata napięcia rotacji silnika.

Dodatkowo określono wartości:

$$(3.2) \quad U(s) = (R + sL)i(s)$$

$$(3.3) \quad U(s) = K_u s \theta(s).$$

Na podstawie równań (3.1) i (3.2), (3.3) oraz zależności: $\theta_d(s) = K_i i(s)$ obliczono funkcję transmitancji modelu silnika:

$$(3.4) \quad \frac{\theta(s)}{U(s)} = \frac{\frac{K_i}{R}}{\frac{J\gamma s^2}{K_m} + \frac{Js^2}{K_m} + \left(\gamma + \frac{K_u K_i}{R}\right)s + 1}$$

gdzie:

$\theta(s)$ - wyjściowa funkcja kąta;

$U(s)$ - funkcja sterująca;

J - całkowity moment bezwładności napędu sprowadzony na wał silnika;

R - rezystancja pasma;

K_i - stała prądowa;

K_u - stała napięciowa;

K_m - stała momentu obrotowego;

$\gamma = L/R$ - stała czasowa elektromagnetyczna.

Uzyskana funkcja transmitancji pozwala na obliczenie zachowania się napędu skokowego w zależności od zadanej funkcji sterowania. Mimo zastosowanych w

tworzeniu modelu wielu uproszczeń m.in.: linearyzacji obwodów magnetycznych, nieuwzględnienia nasycenia, prądów wirowych i histerezy, założenia identyczności uzwojeń pasmowych - opisana funkcja transmitancji pozwala analizować pracę silnika w różnych stanach, przy stosunkowo niewielkich błędach.

Również w pracy [61] został opisany model silnika skokowego z magnesami trwałymi za pomocą transmitancji oraz transformaty Laplace'a. Napięcie zasilające pasmo silnika zostało przedstawione w postaci operatorowej:

$$(3.5) \quad U(s) = RI(s) + K_u \theta(s)$$

i na tej podstawie określono prąd płynący przez pasmo (w postaci operatorowej):

$$(3.6) \quad I(s) = (U(s) - K_u \theta(s)) / R.$$

Określono także transformatę Laplace'a momentu obrotowego silnika skokowego:

$$(3.7) \quad M(s) = K_m I(s) = Js^2 \theta(s) + Ds \theta(s)$$

gdzie:

D - współczynnik tłumienia wewnętrznego;

K_m - stała momentu;

J - całkowity moment bezwładności napędu.

Na podstawie równań (3.6) i (3.7) określono transmitancję silnika, jako stosunek transformaty Laplace'a sygnału wyjściowego do wejściowego:

$$(3.8) \quad \frac{\theta(s)}{U(s)} = \frac{\frac{1}{K_e}}{\frac{JR}{(K_m K_e)} s^2 + \frac{DR}{(K_m K_u)} s + 1}$$

Przy następujących założeniach:

$$(3.9) \quad \gamma = \sqrt{\frac{JR}{K_m K_e}}$$

$$(3.10) \quad e = \frac{D}{2J} \gamma$$

można napisać nowe równanie transmitancji:

$$(3.11) \quad \frac{\theta(s)}{U(s)} = \frac{\frac{1}{K_u \gamma^2}}{s^2 + \frac{2s e}{\gamma} + \frac{1}{\gamma^2}}$$

Na podstawie równania (3.11) po przeprowadzeniu przekształceń otrzymano wyrażenie na odpowiedź silnika na zadane wymuszenie:

$$(3.12) \quad U(s) = U_0 / s$$

w postaci:

$$(3.13) \quad \theta(s) = \frac{E_0}{K_u \gamma^2} \frac{1}{s((s+\alpha)^2 + \beta^2)}$$

gdzie:

$$\alpha = e/\gamma$$

$$\beta = \frac{1}{\gamma} \sqrt{1-e^2}$$

Po przeprowadzeniu transformacji odwrotnej Laplace'a uzyskano funkcję odpowiedzi silnika w dziedzinie czasu:

$$(3.14) \quad \theta(t) = \frac{E_0}{K_e J \gamma^2} \left(\frac{1}{(\alpha^2 + \beta^2)} - \frac{e^{(-\alpha t)} \sin(\beta + \arctg(\frac{\beta}{\alpha}))}{\beta \sqrt{\alpha^2 + \beta^2}} \right).$$

Uzyskana funkcja odpowiedzi w dziedzinie czasu stanowi podstawę do wszechstronnej analizy pracy silników

skokowych przy użyciu maszyn analogowych i cyfrowych.

Model silnika skokowego opisany równaniami macierzowymi dla silników hybrydowych dwupakietowych o rozdzielonym uzwojeniu dwupasmowym podano w pracy [63]. Model matematyczny został wyprowadzony przy założeniu, że silnik jest układem liniowym. Nie są uwzględnione: nasycenie magnetyczne, histereza i prądy wirowe w magnetowodzie. Przyjęto ponadto, że uzwojenia stojana wytwarzają pole sinusoidalne i że oba pasma są identyczne. Silnik jest zasilany impulsami bipolarnymi ze źródła napięciowego, wg cyklogramu (1, 2), (2, -1), (-1, -2), (-2, 1),.. ... przez rezystor forsujący o rezystancji R_f . Na rysunku 3.3 przedstawiono model obwodowy rozpatrywanego silnika oraz wykres wektorowy napięć odpowiadający pierwszemu taktowi komutacji. Magnes trwały zastąpiono cewką wzbudzenia. Ponieważ silnik jest układem o czterech stopniach swobody jest opisany trzema równaniami napięciowymi obwodów elektrycznych i jednym równaniem mechanicznym równowagi momentów.

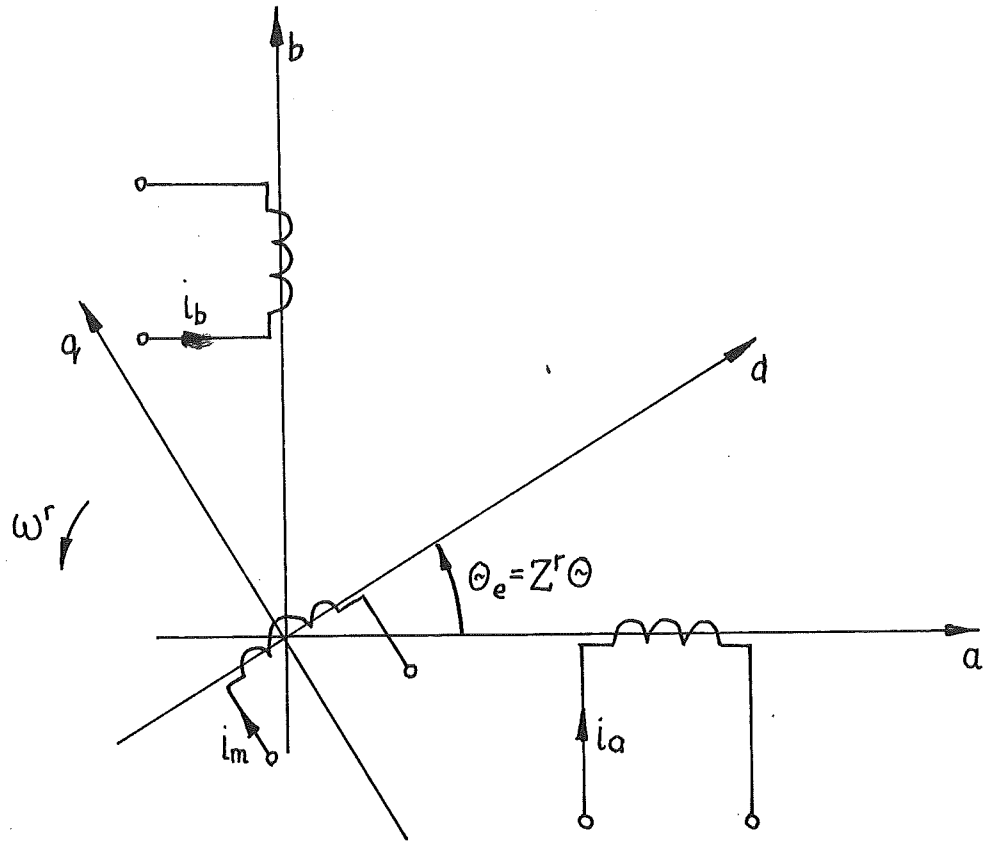
Równania napięciowe zostały wyprowadzone z podstawowych zależności

$$(3.15) \quad u_i = R_i i_i + \frac{d\psi_i}{dt}$$

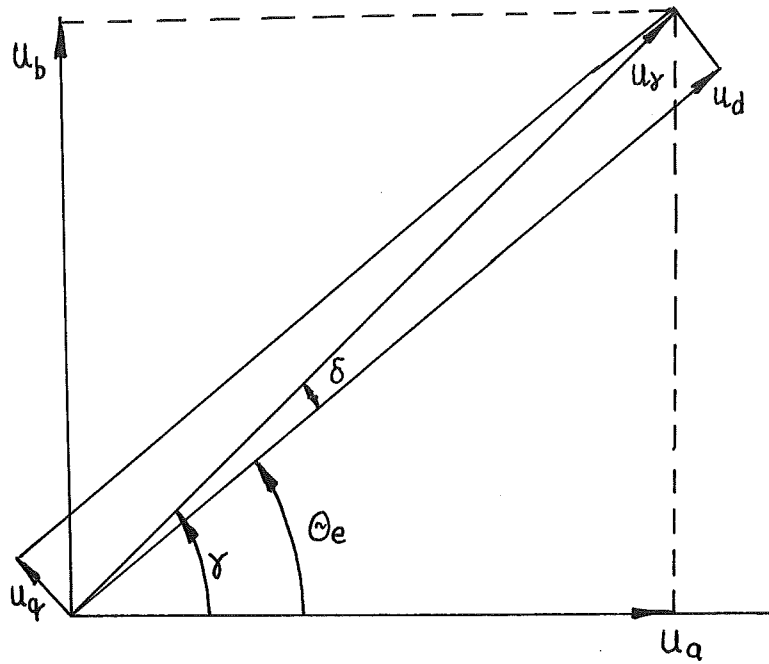
oraz

$$(3.16) \quad \psi_i = \sum L_{im} i_m$$

a)



b)



Rys. 3.3 Schemat obwodowy (a) i wykres wektorowy napięć (b) silnika skokowego

Ze względu na liniowość układu, indukcyjności uzwojeń nie zależą od prądów i są tylko funkcją kąta położenia wirnika

$$(3.17) \quad L_{im} = f(\theta)$$

Równania (3.15) i (3.16) przyjmują postać

$$(3.18) \quad u_i = R_i i_i + \sum_m (L_{im} \frac{d}{dt} i_m + i_m \frac{d}{dt} L_{im})$$

Dla omawianego silnika

$$(3.19) \quad L_{im} = L_{mi} \quad m <> i$$

$$(3.20) \quad \theta_e = Z^r \theta^r$$

Zatem uwzględniając, że prędkość kątowna wirnika wynosi:

$$(3.21) \quad \omega^r = \frac{d\theta^r}{dt} = \frac{1}{Z^r} \cdot \frac{d\theta_e}{dt},$$

a rezystancje obwodów stojana wraz z rezystancjami dodatkowymi

$$(3.22) \quad R_a + R_d = R_b + R_d = R$$

równania napięciowe modelu obwodowego silnika przedstawia się w następującej postaci:

$$(3.23) \quad u_a = R i_a + L_{aa} \frac{di_a}{dt} + L_{ab} \frac{di_b}{dt} + L_{am} \frac{di_m}{dt} + (G_{aa} i_a + G_{ab} i_b + G_{am} i_m) \omega^r$$

$$(3.24) \quad u_b = R i_b + L_{ab} \frac{di_a}{dt} + L_{bb} \frac{di_b}{dt} + L_{bm} \frac{di_m}{dt} + (G_{ab} i_a + G_{bb} i_b + G_{bm} i_m) \omega^r$$

$$(3.25) \quad u_m = R_m i_m + L_{am} \frac{di_a}{dt} + L_{bm} \frac{di_b}{dt} + L_{mm} \frac{di_m}{dt} + (G_{am} i_a + G_{bm} i_b + G_{mm} i_m) \omega^r$$

W równaniach powyższych uwzględniono, że

$$(3.26) \quad i \frac{dL}{dt} = i \frac{dL}{d\Phi} \cdot \frac{d\Phi}{dL} = Z^r i \frac{dL}{d\theta_e} \omega^r = G i \omega^r$$

przy czym indukcyjność rotacji

$$(3.27) \quad G = Z^r \frac{dL}{d\theta_e}$$

Układ równań (3.23), (3.24), (3.25) można wykorzystać do dalszej analizy, jeżeli określi się zależności indukcyjności własnych i wzajemnych uzwojenia silnika od kąta położenia wirnika. Biorąc pod uwagę założenia poczynione na wstępie oraz przyjmując dodatkowo, że bieguny ("duże zęby") stojana nie są uźłobkowane, a uźłobkowany jest tylko wirnik, indukcyjności własne uzwojeń stojana, z dokładnością do podstawowej harmonicznej ich zmienności, można przedstawić następująco:

$$(3.28) \quad L_{aa} = L_o + L_2 \cos 2\theta_e$$

$$(3.29) \quad L_{bb} = L_o - L_2 \cos 2\theta_e$$

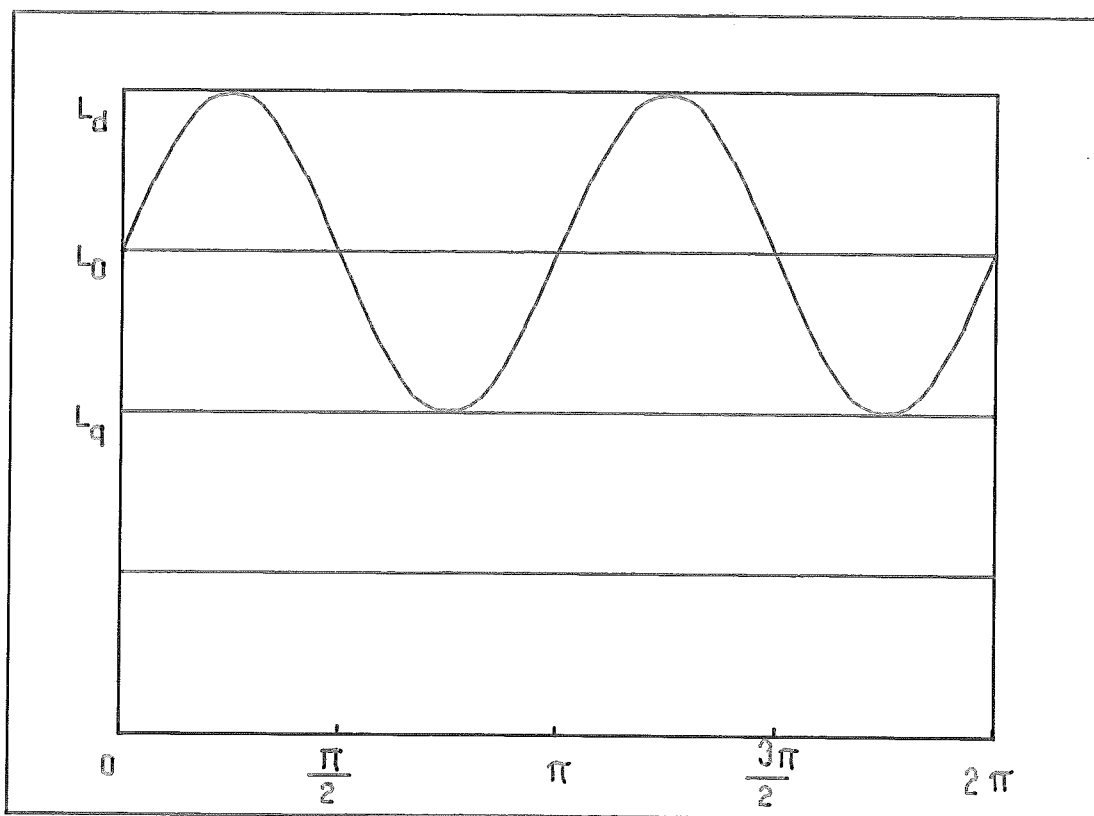
przy czym:

$L_o = (L_d + L_q)/2$ - wartość średnia indukcyjności własnej;

$L = (L_d - L_q)/2$ - amplituda składowej przemiennej.

Ilustruje to zjawisko rys. 3.4. Osie pasm uzwojenia stojana są przesunięte względem siebie w przestrzeni o kąt $\pi/2$. Jednakże - wskutek występowania modulacji permeancji

dla strumienia magnetycznego (spowodowanej ułożkowaniem wirnika), której okres zmienności jest funkcją podwójnego kąta θ_e - osie pasm stojana w stosunku do składowej przemiennej są przesunięte nie o $\pi/2$, a o π - są więc sprzężone.



Rys. 3.4. Zależność indukcyjności własnej uzwojenia stojana od położenia wirnika

Indukcyjność wzajemną pasm stojana można wobec tego wyrazić w postaci:

$$(3.30) \quad L_{ab} = L_{ba} = L_2 \sin 2\theta_e$$

Dla większości magnesów trwałych wzbudzenia, można przyjąć, że ich strumień nie zależy ani od położenia wirnika, ani od oddziaływania prądów stojana. Dlatego dla zastępczej cewki wzbudzenia można z dużym przybliżeniem przyjąć:

$$i_m = \text{const}$$

(3.31)

$$L_{mm} = \text{const.}$$

Natomiast indukcyjności wzajemne stojan wirnik, w rozważanym przypadku identycznych pasm stojana, są określone następującymi wyrażeniami:

$$(3.32) \quad L_{am} = L_{am(\max)} \cos \theta_e$$

$$(3.33) \quad L_{bm} = L_{am(\max)} \sin \theta_e$$

Po uwzględnieniu wyrażeń (3.26)-(3.33), równania (3.23), (3.24), (3.25) przyjmą postać:

$$(3.34) \quad u_a = R i_a (L_o + L_2 \cos 2\theta_e) \frac{di_a}{dt} + L_2 \sin 2\theta_e \frac{di_a}{dt} + \\ + (-2Z^r L_2 i_a \sin 2\theta_e + 2Z^r L_2 i_b \cos 2\theta_e - k_u \sin \theta_e) \omega^r$$

$$(3.35) \quad u_b = R i_b + L_2 \sin 2\theta_e \frac{di_a}{dt} + (L_o - L_2 \cos 2\theta_e) \frac{di_b}{dt} + \\ + (2Z^r L_2 i_a \cos 2\theta_e + 2Z^r L_2 i_b \sin 2\theta_e + k_u \cos \theta_e) \omega^r$$

przy czym:

$$(3.36) \quad k_u = Z^r L_{am(\max)} i_m = \psi_{\max}$$

oznacza stałą napięciową silnika. Można ją wyznaczyć mierząc napięcie na zaciskach stojana indukowane strumieniem magnesów trwałych, przy znanej prędkości obrotowej wirnika. Z punktu widzenia modelu silnika o magnesie trwałym, trzecie równanie (3.25) nic nie wnosi i może być w dalszych rozważaniach pominięte.

Poza równaniami napięciowymi należy ułożyć równanie momentów uwzględniające działanie obciążenia silnika skokowego. Na podstawie zasady d'Alemberta dla najczęściej spotykanej postaci obciążenia, można napisać następujące równanie równowagi momentów silnika:

$$(3.37) \quad J \frac{d\omega^r}{dt} + b\omega^r + M \operatorname{sign}(\omega^r) = M_e$$

w którym :

$b\omega^r$ - moment tarcia lepkiego;

M - moment tarcia suchego (dla $\omega^r > 0$, M ma znak dodatni, dla $\omega^r < 0$ - ujemny);

J - moment bezwładności wirnika i sprowadzony na wał silnika moment bezwładności obciążenia.

Moment elektromagnetyczny M_e można wyznaczyć z zależności znanej z ogólnej teorii elektromechanicznego przetwarzania energii:

$$(3.38) \quad M_e = \frac{P_e}{\omega^r} = 2L_z Z^r i_d i_q + k_u i_q$$

gdzie:

i_d, i_q - prądy w uzwojeniach stojana wyrażone we współrzędnych d-q.

Równania (3.34), (3.35) oraz (3.36) umożliwiają pełną analizę zachowania się silnika skokowego omawianego typu metodami numerycznymi lub analogowymi. Wszystkie potrzebne wartości parametrów można znaleźć w

katalogach silników i ich sterowników lub wyznaczyć metodami pomiarowymi (dotyczy to wartości L_0 , L_2 , k_u). Zawsze są również znane parametry obciążenia.

Przegląd rozwiązań konstrukcyjnych silników skokowych dokonany w pracy [40] pozwala, nie ograniczając generalnej słuszności opisu matematycznego, przyjąć jako uogólniony silnik skokowy m-pasmową maszynę elektryczną z dwustronnie uzębioną szczeliną i uzwojeniami wzbudzonymi impulsowo. Zakładamy jednocześnie, że wszystkie uzwojenia fazowe rozłożone są tylko na stojanie lub tylko na wirniku (w przypadku konstrukcji odwróconej).

Przy przedstawianiu charakteru zjawisk i oddziaływań elektromagnetycznych w silniku skokowym nie uwzględnia się nasycenia obwodu magnetycznego. Ilościowy wpływ nasycenia uwzględnia się natomiast przy obliczeniach obwodu magnetycznego w stanach ustalonych. Wyznaczone przy tym współczynniki nasycenia, indukcyjności i stałe czasowe uzwojeń przyjmuje się jako niezależne od wartości chwilowych prądów w uzwojeniach silnika skokowego będącego w ruchu. Przyjęcie takiego założenia upraszczającego pozwala traktować strumienie skojarzone uzwojeń, występujące w układzie równań opisujących napęd, jako liniowe funkcje prądów:

$$(3.39) \quad \psi_a(\theta, i) = L_{a1}(\theta)i_1 + L_{a2}(\theta)i_2 + \dots + L_{aa}(\theta)i_a$$

$$a=1, 2, 3, \dots, m$$

ze współczynnikami zmieniającymi się periodycznie, zależnymi tylko od położenia kąowego wirnika.

Uwzględniając przyjęte założenia - elektromagnetyczny moment obrotowy można wyrazić za pomocą wartości chwilowych prądów i indukcyjności uzwojeń pasmowych.

$$(3.40) \quad M_e = \frac{dW}{d\theta_m} = p \frac{dW}{d\theta} = \frac{p}{2} \sum_{\alpha=1}^m i_\alpha \frac{d\psi_\alpha}{d\theta} =$$

$$= \frac{p}{2} \sum_{i=1}^m \sum_{\alpha=1}^m i_i i_\alpha \frac{dL_{i\alpha}(\theta)}{d\theta} .$$

Zgodnie z zasadami mechaniki teoretycznej, prądy w uzwojeniach i kąt obrotu wirnika są współrzędnymi uogólnionymi rozpatrywanego układu elektromechanicznego. W celu opisanego tego układu należy ułożyć m równań równowagi elektrycznej obwodów, utworzonych przez uzwojenia, elementy przełączające i źródło zasilania oraz jedno równanie równowagi mechanicznej napędu.

Te równania wyjściowe mają następującą postać:

$$(3.41) \quad R_a i_a + \frac{d\psi_a}{dt} = u_a$$

$$(3.42) \quad J \frac{d^2\theta_m}{dt^2} + M = M_e$$

gdzie:

$a=1, 2, 3, \dots, m$ - numery obwodów elektrycznych
tworzonych przez pasma silnika;

u_a, i_a, ψ_a - wartości chwilowe odpowiednio
napięcia prądu i strumienia
skojarzonego k-tego obwodu
elektrycznego;

R_a - rezystancja k-tego obwodu
elektrycznego;

J - całkowity moment bezwładności
sprowadzony na wał silnika;

M - całkowity moment oporowy
obciążenia i strat biegu
jałowego silnika;

M_e - elektromagnetyczny moment
obrotowy, wyznaczony z (3.37);

$\theta_m = \theta/p$ - mechaniczny kąt obrotu wirnika.

Wobec tego napęd z m-pasmowym silnikiem skokowym wymaga

dla pełnego opisu $m+1$ równań łącznie z równaniem momentów.

Przy zasilaniu dowolnego pasma uzwojenia ze źródła prądowego równanie równowagi elektrycznej degeneruje się ($I_\alpha = \text{const}$), a więc odpada odpowiednia współrzędna. Przy zasilaniu silnika skokowego ze źródła prądowego, co często ma miejsce w praktyce, równania równowagi elektrycznej zanikają całkowicie i układ równań sprowadza się do jednego równania momentów.

Wszystkie prądy wchodzące do równania momentu elektromagnetycznego są wtedy stałe w przedziałach każdego taktu komutacji i moment elektromagnetyczny zależy tylko od położenia wirnika i numeru stanu elektrycznego. Okoliczność tę wykorzystuje się szeroko w praktyce do uproszczenia opisu matematycznego jeśli tylko pozwalają na to charakterystyki źródła zasilania lub obciążenia napędu dużym momentem bezwładności.

W dalszej części będą rozpatrywane napędy skokowe zasilane ze źródeł napięciowych.

Jest to najbardziej ogólny przypadek pozwalający wykryć cechy szczególne różnych odmian silników skokowych, ich możliwości energetyczne i właściwości częstotliwościowe. Należy zwrócić uwagę, że rezystancja obwodu uzwojenia pasmowego zależy od sposobu komutacji. Przy komutacji bipolarnej rezystancję tę można uważać za stałą, w przypadku komutacji unipolarnej jest to dopuszczalne tylko wtedy, gdy

uzwojenie jest zbocznikowane diodą.

W dalszych rozważaniach rezystancje wszystkich obwodów będą uważane za stałe i jednakowe, $R_a = R = \text{const}$, $a=1,2,3,\dots,m$. Uwzględniając powyższe uwagi można teraz napisać równanie (3.40) w postaci rozwiniętej w odniesieniu do rzeczywistych współrzędnych fazowych. W tym celu należy podstawić w równaniach (3.41) wyrażenie (3.39) na strumienie sprzężone, przeprowadzić różniczkowanie i wyrazić moment elektromagnetyczny za pomocą indukcyjności i prądów zgodnie ze wzorem (3.40). Po przeprowadzeniu wymienionych operacji otrzymamy:

$$(3.43) \quad Ri_i + \sum_{k=1}^m L_{ik} \frac{di_k}{dt} + \frac{d\theta}{dt} i_k \frac{dL_{ik}}{d\theta} = u_i$$

$$i=1,2,\dots,m$$

$$(3.44) \quad \frac{J}{p} \frac{d^2\theta}{dt^2} + M = \frac{p}{2} \sum_{j=1}^m \sum_{k=1}^m i_j i_k \frac{dL_{jk}}{d\theta}$$

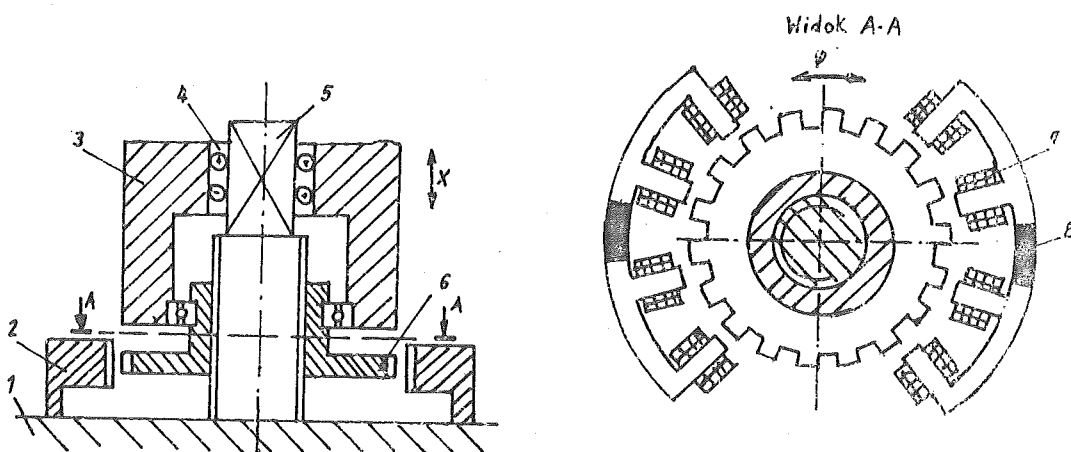
Suma $\sum_{k=1}^m L_{ik} \frac{di_k}{dt}$ określa wartość napięcia indukcji

własnej i wzajemnej w i -tym obwodzie, a suma $\frac{d\theta}{dt} \sum_{k=1}^m i_k \frac{dL_k}{d\theta}$ wartość napięcia rotacji indukowanego przy ruchu wirnika.

Wymuszenia zewnętrzne zadane są w postaci zmieniających się okresowo napięć pasmowych u_b , $b=1,2,3,\dots,m$. Otrzymane równanie różniczkowe nieliniowe o współczynnikach zmiennych okresowo umożliwia analizę wszystkich stanów pracy konkretnego napędu skokowego. Równania te są stosunkowo proste i w większości przypadków nie wymagają dodatkowych przekształceń. Tylko przy opisie silników skokowych o niewielkiej liczbie pasm $m < 4$ i w przypadku niewystępowania sprzężeń między pasmami potrzebne są dodatkowe przekształcenia.

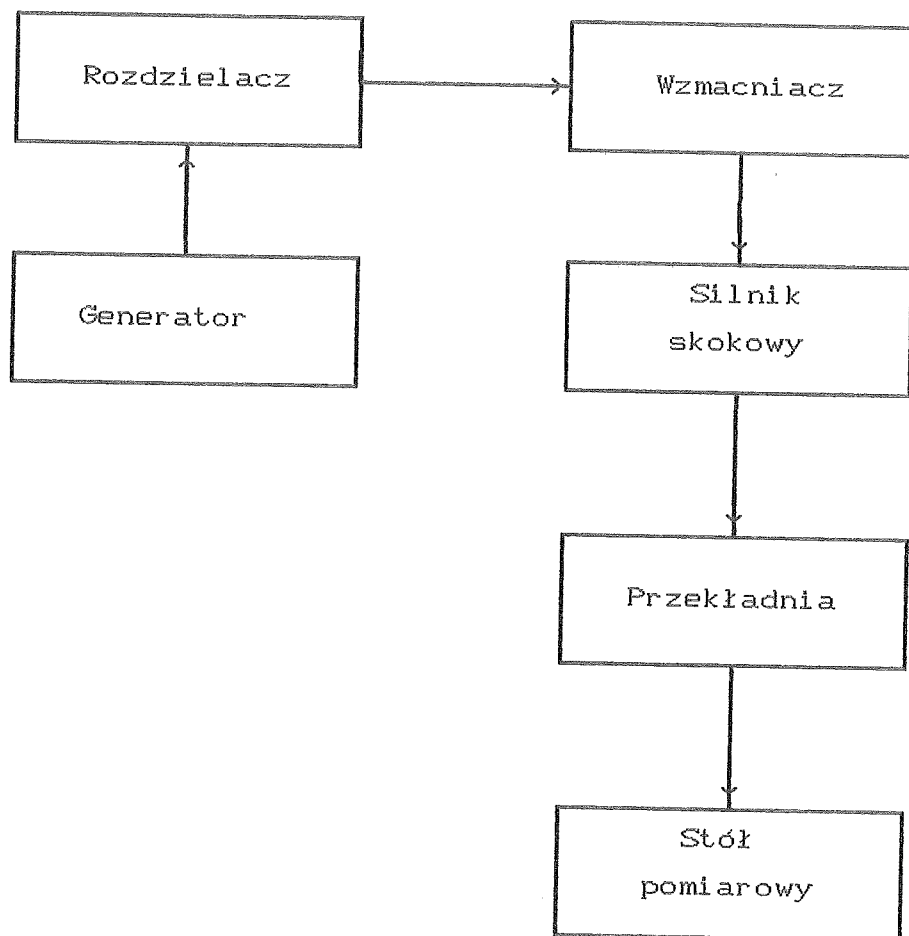
W pracy [77] został przedstawiony model silnika skokowego reluktancyjnego, czteropasmowego zorientowany na blokową analizę numeryczną. Sprowadza się ona do rozbicia modelu na oddzielnie obliczane bloki programów, które są zestawiane w zależności od potrzeb. Podejście takie ułatwia konstrukcję programów analizy numerycznej modelu silnika skokowego. W celu stworzenia modelu zaprojektowano napęd skokowy (rys. 3.5) przeznaczony także do

eksperymentalnej weryfikacji otrzymanych wyników analizy komputerowej.



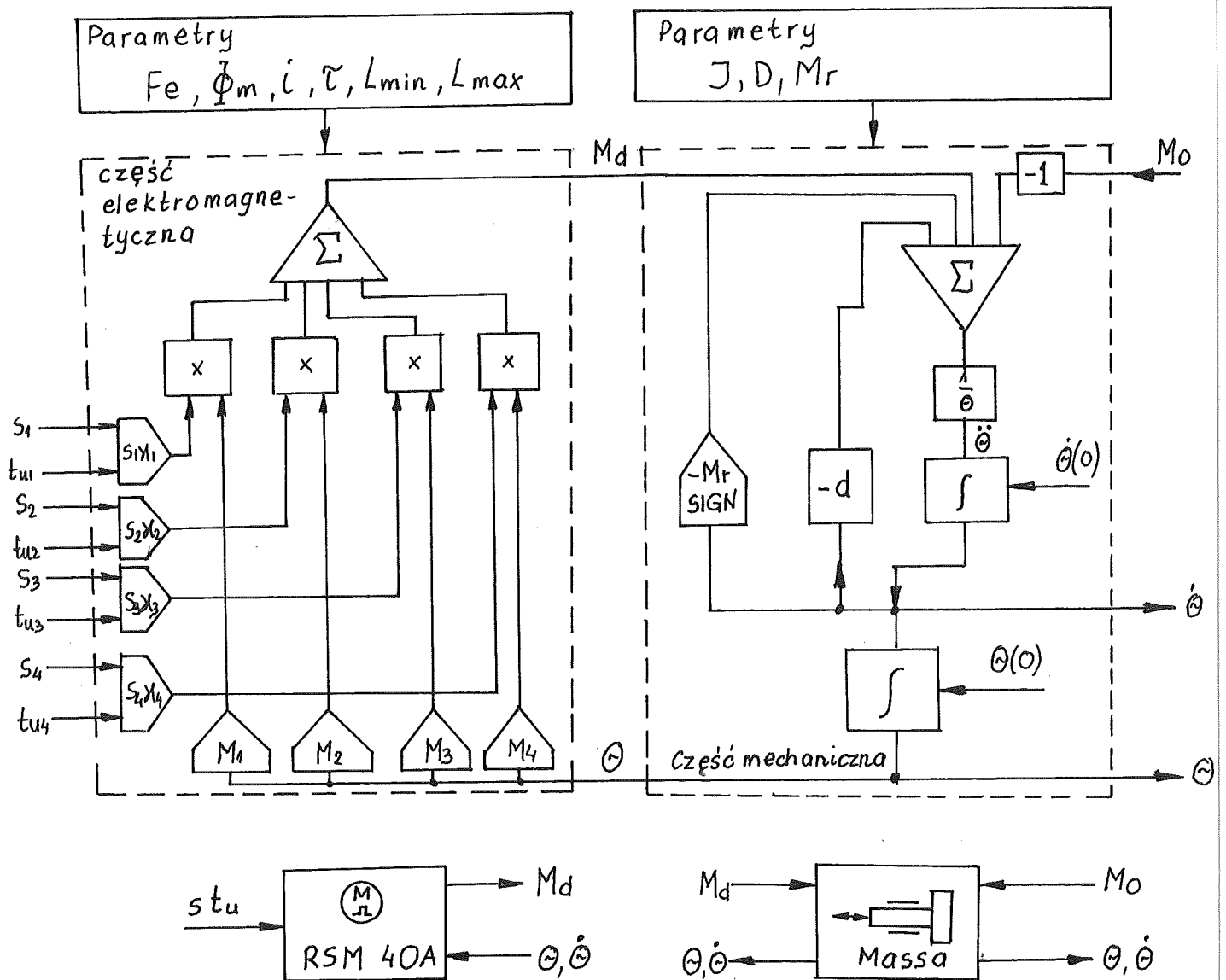
Rys. 3.5. Schemat budowy napędu z silnikiem skokowym:

- 1 - podstawa;
- 2 - jarzmo;
- 3 - stół pozycjonujący;
- 4 - łożysko kulkowe;
- 5 - wał;
- 6 - wirnik;
- 7 - uzwojenie;
- 8 - magnes trwały.



Rys. 3.6. Schemat układu dynamicznego napędu skokowego

Na tej podstawie utworzono schemat systemu dynamicznego napędu skokowego pokazany na rys. 3.6. Model silnika został następnie podzielony na dwie podstawowe części: elektromagnetyczną i mechaniczną. Utworzono dwa równania opisujące te części:



Rys. 3.7. Schemat blokowy modelu silnika skokowego

$$(3.45) \quad M_{de} = \sum_{\alpha=1}^4 \left(- \frac{s_{\alpha} \Phi_m F_e L_2 Z^r}{4L_o} \sin(Z^r \theta - \theta_{\alpha}^r) + \right. \\ \left. + \frac{s_{\alpha} F_e^2 L_2^2 Z^r}{4L_o} \sin(2(Z^r \theta - \theta_{\alpha}^r)) \right)$$

gdzie: Φ_m - strumień magnesu trwałego;

F_e - przepływ jednej cewki;

Z^r - liczba zębów wirnika;

L_2 - amplituda indukcyjności pasma.

$$(3.46) \quad M_{dm} = J \frac{d^2 \theta}{dt^2} + D \frac{d\theta}{dt} + M \text{sign} \left(\frac{d\theta}{dt} \right).$$

Równania te zostały przedstawione w postaci schematu analogowego na rys. 3.7. Do komputerowej analizy utworzonego modelu zostały napisane podprogramy w języku FORTRAN opisujące poszczególne jego elementy:

RSM4DA - silnik skokowy, czterobiegunowy z magnesem trwałym;

RSM4DB - silnik skokowy, czterobiegunowy, reluktancyjny;

MASSE - masa wirująca;

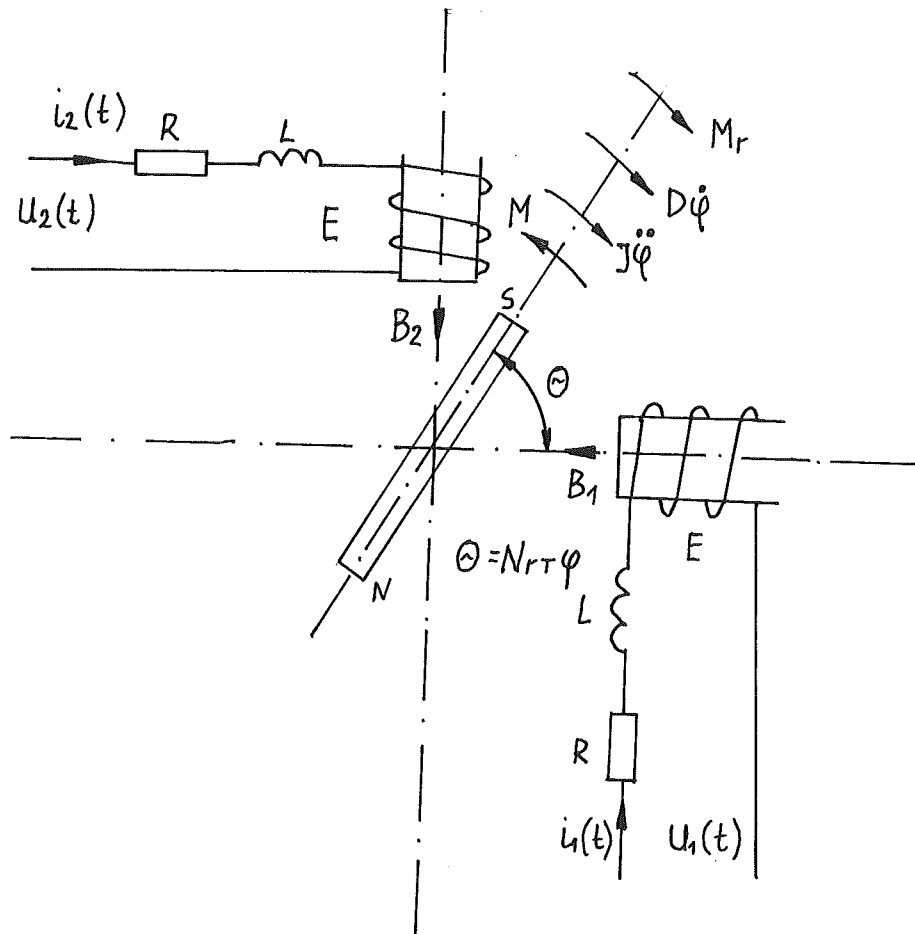
DIMAM - pamięć modelu;

IPV - rozdzielacz impulsów;

Podprogramy te mogą być użyte do analizy numerycznej silników skokowych tej klasy.

W pracy [84] został przedstawiony model silnika skokowego z magnesem trwałym analizowany w zakresie jednego skoku.

Rys. 3.8. przedstawia schemat układu elektromechanicznego modelu. Utworzono dwa równania napięciowe opisujące zależności między prądem, a napięciem w obwodach pasm silnika (3.47), (3.48).



Rys. 3.8. Schemat modelu silnika skokowego z magnesem trwałym

$$(3.47) \quad u_1(t) = Ri_1(t) + L \frac{di_1(t)}{dt} + U_e$$

$$(3.48) \quad u_2(t) = Ri_2(t) + L \frac{di_2(t)}{dt} + U_e$$

gdzie: U_e - napięcie indukowane w uzwojeniach silnika.

Przy założeniu że:

$$(3.49) \quad U_e = \frac{K_e}{Z^r} \frac{d\theta(t)}{dt} \sin \theta(t)$$

można napisać nowe równania różniczkowe napięciowe stojana:

$$(3.50) \quad u_1(t) = Ri_1(t) + L \frac{di_1(t)}{dt} + K_e \frac{d\theta(t)}{dt} \sin \theta(t)$$

$$(3.51) \quad u_2(t) = Ri_2(t) + L \frac{di_2(t)}{dt} + K_e \frac{d\theta(t)}{dt} \cos \theta(t)$$

Następnie utworzono równanie momentów silnika:

$$(3.52) \quad -K_m I_{m1} \sin \theta_o \Delta\theta(t) - K_m I_{m2} \cos \theta_o \Delta\theta(t) + K_m \cos \theta_o \Delta i_2(t) - \\ - K_m \sin \theta_o \Delta i_1(t) = \frac{J}{Z^r} \frac{d^2}{dt^2} \Delta\theta(t) + \frac{D}{Z^r} \frac{d}{dt} \Delta\theta(t) + M(t)$$

oraz przekształcono równania napięciowe:

$$(3.53) \quad \Delta u_1(t) = R \Delta i_1(t) + L \frac{d}{dt} \Delta i_1(t) + \frac{K_e}{Z^r} \sin \theta_o \frac{d}{dt} \Delta\theta(t)$$

$$(3.54) \quad \Delta u_2(t) = R \Delta i_2(t) + L \frac{d}{dt} \Delta i_2(t) + \frac{K_e}{Z^r} \cos \theta_o \frac{d}{dt} \Delta\theta(t)$$

Równania (3.52)-(3.54) stanowią układ równań różniczkowych opisujących zastosowany model silnika skokowego.

Po zastosowaniu transformaty Laplace'a i rozwiązaniu układu równań w dziedzinie operatorowej otrzymano wyrażenie na odpowiedź silnika w zakresie jednego skoku:

$$(3.55) \quad \Delta\theta(s) = \frac{(s + \frac{D}{J}) \Delta\theta(0)}{s^2 + s \frac{D}{J} + \frac{\sqrt{2} K_m I Z^r}{J}}$$

Równanie (3.55) pozwala badać zachowanie się silnika w dziedzinie częstotliwości oraz po przekształceniu odwrotnym Laplace'a również w dziedzinie czasu. W równaniach (3.47)-(3.55) zastosowano dodatkowo następujące oznaczenia:

- $\Delta i_1, \Delta i_2$ - zmiany prądu pasm 1 lub 2 [A];
- $\Delta u_1, \Delta u_2$ - zmiany napięcia zasilającego pasma [V];
- $\Delta\theta(t)$ - zmiany położenia kąтового wirnika [deg];
- D - współczynnik tłumienia wewnętrznego;
- J - całkowity moment bezwładności napędu;
- Z^r - liczba zębów wirnika;
- M - moment obciążenia.

Opisane powyżej modele matematyczne różnych odmian silników skokowych pozwalają na dosyć dokładną analizę numeryczną. Modele te powstały głównie w oparciu o analizę obwodową oraz teorię elektromechanicznego przetwarzania energii.

4. NUMERYCZNE MODELOWANIE SILNIKÓW SKOKOWYCH

Opisane modele silników skokowych stworzono przy pewnych uproszczeniach zjawisk elektromagnetycznych w silniku. Odnosi się to szczególnie do opisanie funkcji momentu elektromagnetycznego silnika w funkcji kąta i prądu zasilającego pasma. Zastosowanie tak opisanych modeli może zadowalać w wielu przypadkach, szczególnie przy rozważaniach teoretycznych, jednak zastosowania praktyczne silników skokowych wymagają wykorzystania takich metod, które pozwalają dokładniej modelować istniejące lub projektowane silniki. Nowoczesne metody numeryczne w połączeniu z wydajnym sprzętem komputerowym umożliwiają badanie bardziej skomplikowanych modeli matematycznych, pozwalają uwzględnić wiele pomijanych dotychczas zjawisk. Dotyczy to przede wszystkim opisanie momentu elektromagnetycznego, występowania histerezy, prądów wirowych oraz nasycenia obwodów magnetycznych. Jednocześnie nowoczesne sterowniki często zasilają silnik ze sterowanego źródła prądowego, co pozwala uprościć równania równowagi elektrycznej silnika. Napędy skokowe wymagają dokładnego projektowania z zastosowaniem wspomaganie komputerowego aby otrzymać zadane parametry napędu przy minimalnych nakładach ekonomicznych. Numeryczne modelowanie silników skokowych pozwala wszechstronnie zbadać zachowanie projektowanego silnika, jego własności w różnych stanach pracy.

4.1. MODEL SILNIKA SKOKOWEGO PRZY PRACY MINISKOKOWEJ.

Przy tworzeniu takiego modelu należy przyjąć pewne założenia określające klasę modelowanych napędów i rodzaj sterownika zasilającego, poniżej są one przedstawione:

- 1) modelowany jest silnik dwupasmowy, hybrydowy, czterobiegunowy (wynika to z powszechnego stosowania tego typu silników w praktyce);
- 2) modelowany silnik jest zasilany z bipolarnego sterowanego źródła prądowego;
- 3) nie uwzględnia się procesów przejściowych podczas przełączania prądów w pasmach silnika.
- 4) obwody magnetyczne pracują w liniowym zakresie charakterystyki, nie uwzględnia się zjawiska histerezy

W.w. uproszczenia nie wpływają znacząco na dokładność modelowania napędu dyskretnego przy sterowaniu miniskokowym. Sterownik z zasilaniem prądowym jest zastosowany w badaniach napędu co pozwala na zastosowanie tego założenia przy modelowaniu. Ograniczenie się do symulacji jednego rodzaju silnika wynika z ograniczeń czasowych i sprzętowych podczas wykonywania pracy. Silnik dwupasmowy z zasilaniem bipolarnym jest analogiem czteropasmowego silnika zasilanego sygnałami unipolarnymi.

Jak wynika z omówionych w poprzednim rozdziale opisów modeli matematycznych silników skokowych, podstawowym problemem przy symulacji komputerowej napędu dyskretnego jest

rozwiązywanie układu równań różniczkowych zwyczajnych, który opisuje badany model. Podstawowe metody rozwiązywania takich układów będą omówione w następnym punkcie.

4.2. NUMERYCZNE METODY ROZWIĄZYWANIA UKŁADU RÓWNAŃ RÓŻNICZKOWYCH ZWYCZAJNYCH.

Rozwiązywanie układów równań różniczkowych jest jednym z podstawowych zastosowań matematyki w technice. Różne metody numerycznego rozwiązywania takich równań można znaleźć w [101], [13] oraz [72]. Poniżej zostaną przedstawione niektóre z tych metod.

Numeryczne rozwiązywanie układów równań różniczkowych zakłada istnienie następującego układu:

$$(4.1) \quad x'_i = f(x_1, x_2, x_3, \dots, x_m; t), \quad i = 1, 2, \dots, m,$$

w którym niewiadome są funkcje $x_1(t)$, $x_2(t)$, ..., $x_m(t)$ jednej zmiennej niezależnej t . Należy określić warunki początkowe:

$$(4.2) \quad x_i(t_0) = x_{i0}, \quad i = 1, 2, \dots, m,$$

które wyrażają fakt, że w danej chwili początkowej t_0 dane są wartości funkcji x_{i0} dla każdej zmiennej

niezależnej. Tak sformułowany problem ma jednoznaczne rozwiązanie w otoczeniu punktu t_0 tylko wtedy gdy funkcje f_i są funkcjami analitycznymi. Równania wyższych rzędów można także sprowadzić do postaci (4.1) przez zmianę współrzędnych opisujących układ.

4.2.1. METODA EULERA PIERWSZEGO RZĘDU.

Na podstawie warunków początkowych znane są wartości zmiennych zależnych $x_i(t_0) = x_{i0}$ dla chwili $t = t_0$. Należy wyznaczyć wartości tych zmiennych w chwili $t = t_1$. Jeżeli przyjmiemy przyrost zmiennej niezależnej $h = t_1 - t_0$ (krok) dostatecznie mały, to pochodną x_i' można będzie zastąpić ilorazem różnicowym $(x_{i1} - x_{i0})/h$. Na tej podstawie z wyrażenia (4.1) wynika że:

$$(4.3) \quad x_{i1} = x_{i0} + hf_i(x_{10}, x_{20}, \dots, x_{m0}), \quad i=1, 2, \dots, m.$$

Obliczone wartości x_{i1} można przyjąć za początkowe dla następnego kroku i powtarzając obliczenia (4.3) obliczyć krok po kroku wartości zmiennych dla dowolnych czasów $t_i = h \cdot i$ (można w takim postępowaniu zmieniać krok całkowania dostosowując go do aktualnego przebiegu funkcji). W taki sposób otrzymuje się wartości funkcji szukanej w postaci tabelarycznej. Błąd metody można ocenić przez zastosowanie rozwinięcia wartości $x_i(t_1)$

według wzoru Taylora

$$x_{i1} = x_{i0} + hf_i(x_{10}, x_{20}, \dots, x_{m0}) + h^2/2! * f'_i(y_{10}, \dots, y_{m0})$$

gdzie: symbolem y_{i0} oznaczono pewną wartość z przedziału $x_{i0} \leq y_{i0} \leq x_{i0}$. Jeżeli M oznacza górną wartość bezwzględną pochodnych f'_i w tym przedziale, to błąd $|e| \leq h^2/2! * M$.

Ponieważ jest on proporcjonalny do h^2 , zapisuje się go symbolicznie $O(h^2)$. Wzór (4.3) można zatem zapisać w postaci:

$$x_{i1} = x_{i0} + hf_i(x_{10}, x_{20}, \dots, x_{m0}) + O(h^2).$$

Z tego wynika, że metoda Eulera ma dokładność pierwszego rzędu, gdyż zachowuje dokładnie człony z pierwszą potęgą h . Obliczony błąd występuje w każdym kroku całkowania, co przy niesprzyjającym układzie liczb może spowodować narastanie tego błędu w trakcie powtórzeń, w sposób lawinowy.

4.2.2. METODA TAYLORA DRUGIEGO RZĘDU.

Metoda Eulera 1-go rzędu jest stosowana coraz rzadziej ze względu na potrzebę stosowania małego kroku oraz możliwość występowania błędów lawinowych. Zamiast niej stosuje się metody wyższych rzędów np. drugiego. W zapisie

"prim" oznacza pochodną względem t , a indeks "a" po przecinku pochodną względem x . Nie wpisane elementy odnoszą się do chwili $t=t_i$. Szukane funkcje x_i można rozwinąć w szereg Taylora, ograniczając się do członów drugiego rzędu,

$$x_{i1} = x_{i0} + \frac{h}{1!} x_{i0}' + \frac{h^2}{2!} x_{i0}''.$$

Korzystając z (4.1) i praw różniczkowania otrzymuje się

$$(4.4) \quad x_{i1} = x_{i0} + hf_i + \frac{h^2}{2} \sum_{\alpha=1}^m f_{i,\alpha} f_{\alpha}, \quad \text{dla } \alpha=1..m.$$

Aby można było zastosować ten wzór należy obliczyć $f_{i,\alpha}$. Można to przeprowadzić w następujący sposób:

$$(4.5) \quad \begin{aligned} k_{1i} &= hf_i(x_{10}, \dots, x_{m0}), \\ k_{2i} &= hf_i(x_{10}, \dots, x_{m0} + \alpha k_{m0}), \\ x_{i1} &= x_{i0} + w_1 k_{1i} + w_2 k_{2i}. \end{aligned}$$

W powyższych równaniach w_1 i w_2 są liczbowymi współczynnikami tak dobranymi, aby (4.5) było identyczne z wyrażeniem (4.4) z dokładnością do członów zawierających h i h^2 . Należy zatem rozwinąć (4.5) w szereg Taylora zachowując tę dokładność

$$k_{zi} = hf_i + \alpha h^2 \sum_i f_{i,\alpha} f'_\alpha.$$

Po podstawieniu tej wartości do wzoru (4.5) otrzymuje się:

$$x_{i1} = x_{i0} + w_1 hf_i + w_2 (hf_i + \alpha h^2 f_{i,\alpha} f'_\alpha).$$

Aby otrzymać wzór identyczny z (4.4), muszą zachodzić następujące związki:

$$(4.6) \quad w_1 + w_2 = 1, \quad w_2 \alpha = 1/2.$$

Z tych dwóch warunków można określić dwie niewiadome przy trzeciej założonej dowolnie.

4.2.3. METODA RUNGEGO-KUTTY-GILLA

Przy zastosowaniu w rozwinięciu w szereg Taylora wyrazów zawierające h , h^2 , h^3 , h^4 , otrzymuje się metodę czwartego rzędu, zwaną metodą Rungego-Kutty-Gilla od nazwisk autorów algorytmu. Metoda ta wymaga czterokrotnego obliczania wartości funkcji w każdym kroku całkowania.

Rozpatruje się równanie

$$y'_i = f(x_1, x_2, x_3, \dots, x_m), \quad i = 1, 2, \dots, m,$$

z warunkami początkowymi (4.2). Należy wybrać krok h i wprowadzić oznaczenia: $x_i = x_0 + i \cdot h$, $y_i = y(x_i)$, a następnie rozpatruje się liczby postaci:

$$\begin{aligned}k_i(1) &= hf(x_i, y_i), \\k_i(2) &= hf(x_i + h/2, y_i + k_i(1)/2), \\k_i(3) &= hf(x_i + h/2, y_i + k_i(2)/2), \\k_i(4) &= hf(x_i + h, y_i + k_i(3)).\end{aligned}$$

Zgodnie z metodą Rungego-Kutty-Gilla wartości y_i funkcji $y(x)$ określa się według wzoru:

$$\begin{aligned}y_i &= y + \Delta y_i, \Delta y_i = 1/6(k_i(1) + 2k_i(2) + 2k_i(3) + k_i(4)), \\i &= 1, 2, \dots, m.\end{aligned}$$

Według tego algorytmu został opracowany podprogram w języku Pascal realizujący rozwiązywanie układu równań różniczkowych zwyczajnych pod nazwą RKG4. Treść tego podprogramu jest przedstawiona poniżej:

```
procedure RKG4(var x1,q1:mat; n:integer; h:real);  
  
*****  
* Procedura rozwiązuje układ równań różniczkowych zwyczajnych *  
* metoda Rungego - Kutty - Gilla 4-go rzędu *  
*****  
  
max liczba równań = 10;  
  
przed pierwszym wywołaniem x[n] zawiera warunki początkowe;
```



```
q[n] = 0;
```

```
h-okresla krok calkowania
```

```
n-okresla liczbe rownan;
```

```
fprim jest procedura obliczajaca pochodne
```

```
xprim[n] w punkcie x[n];
```

```
Procedura jest deklarowana w postaci
```

```
"Procedure fprim(var x,xprim:mat);
```

```
type MatPom=array [1..4] of real;
```

```
const a:MatPom=(0.5,0.292893219,1.707106781,0.1666666667);
```

```
b:MatPom=(0.5,0.292893219,1.707106781,0.3333333333);
```

```
c:MatPom=(0.5,0.292893219,1.707106781,0.5);
```

```
var xprim:mat;
```

```
i,j:integer;
```

```
ak,r:real;
```

```
begin
```

```
for j:=1 to 4 do begin
```

```
fprim(x1,xprim);
```

```
for i:=1 to n do begin
```

```
ak:=h*xprim[i];
```

```
r:=a[j]*ak-b[j]*q1[i];
```

```
x[i]:=x[i]+r;
```

```
q1[i]:=3.0*r+q1[i]-c[j]*ak;
```

```
end;
```

end;

end;

Podprogram RKG4 pozwala rozwiązywać układ do dziesięciu równań różniczkowych zwyczajnych, których rozwiązaniem są funkcje wielu zmiennych (do 10). Przy stosunkowo dużym kroku podprogram ten zapewnia uzyskanie dużej dokładności i szybkości pracy. Analogiczne podprogramy istnieją w innych językach programowania.

4.2.4. METODA MILNE'A

W omówionej poprzednio metodzie Rungego-Kutty-Gila w każdym kroku całkowania należało czterokrotnie obliczać wartość wektora funkcji. Doskonalszy jest algorytm Milne'a który wykorzystuje wartości obliczone w czterech poprzednich krokach, co pozwala zmniejszyć dwukrotnie liczbę obliczeń co znacznie skraca czas obliczeń.

Zasadę metody Milne'a można wyjaśnić na przykładzie równania różniczkowego dla jednej zmiennej:

$$(4.7) \quad \frac{dx}{dt} = f(x)$$

Założono, że dla kolejnych kroków całkowania t_1 , $t_2 = t_1 + h$, $t_3 = t_2 + h$, $t_4 = t_3 + h$ znamy wartości funkcji x_1 , x_2 , x_3 , x_4 oraz wartości pochodnych

$\frac{dx_1}{dt}, \frac{dx_2}{dt}, \frac{dx_3}{dt}, \frac{dx_4}{dt}$. Należy znaleźć wartości x_5 w punkcie $t_5 = t_4 + h$.

Można przybliżyć rozwiązanie w postaci rozwinięcia:

$$(4.8) \quad x_5 = \sum_{i=1}^4 \left(a_i x_i + h b_i \frac{dx_i}{dt} \right) + h b_5 \frac{dx_5}{dt}$$

Współczynniki liczbowe $a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, b_5$ dobiera się tak aby wzór powyższy był słuszny dla rozwinięcia funkcji x przybliżonej wielomianem czwartego stopnia:

$$(4.9) \quad x = c_1 + c_2 t + c_3 t^2 + c_4 t^3 + c_5 t^4$$

Wstawiając do wzorów (4.7) i (4.8) oraz porównując w tożsamościach współczynniki przy jednakowych potęgach, otrzymuje się układ równań do wyznaczenia współczynników a_i i b_i oraz c_i . Jednak równań jest mniej niż niewiadomych. Z tego względu istnieje wiele różnych rozwiązań. Przykładowo:

$$(4.10) \quad x_5 = x_1 + \frac{4}{3} h \left(2 \frac{dx_2}{dt} - \frac{dx_3}{dt} + 2 \frac{dx_4}{dt} \right)$$

$$(4.11) \quad x_5 = x_3 + \frac{1}{3} h \left(\frac{dx_3}{dt} + 4 \frac{dx_4}{dt} + \frac{dx_5}{dt} \right)$$

Wzór (4.10), który pozwala obliczyć $\frac{dx_5}{dt}$ na podstawie wartości x_1 ,

$$\frac{dx_2}{dt}, \frac{dx_3}{dt}, \frac{dx_4}{dt}, \text{ obliczonych wcześniej.}$$

Jest on mniej dokładny niż wzór (4.11),

który po prawej stronie zawiera nieobliczoną jeszcze

wartość $\frac{dx_5}{dt}$.

Dlatego należy postępować w następujący sposób:

1) Oblicza się wartość x_5 na podstawie (4.10),

2) Oblicza się $\frac{dx_5}{dt} = f(x_5)$ na podstawie (4.7),

3) Oblicza się nową dokładniejszą wartość x_5 na podstawie wzoru (4.11).

Jeżeli ta nowa wartość różni się od obliczonej w punkcie 1 należy powtórzyć punkty 2 i 3 aż do uzyskania zadowalającej dokładności.

Gdyby zastosować przybliżenie wielomianem stopnia piątego, zamiast czwartego i gdyby przyjąć, że współczynnik c_6 zmienia się nieznacznie, po etapie 1 można by dokonać jeszcze dodatkowej poprawki:

$$(4.12) \quad x_5^p = x_5^v + \frac{28}{29} (x_4^{(ze\ wzoru\ 5)} - x_4^{(ze\ wzoru\ 4)})$$

Stosowanie tego zabiegu jest tak skuteczne, że w praktyce iterację według etapu 3 wykonuje się tylko raz. W tym przypadku wzór (4.7) stosuje się dwa razy w jednym kroku

całkowania.

Metoda Milne'a jest znacznie szybsza od metody Rungego-Kutty-Gilla lecz wymaga na początku trzykrotnego użycia tej drugiej. Otrzymuje się w ten sposób wartości x_1 , x_2 , x_3 , x_4 .

4.3 PROGRAM ANALIZY NUMERYCZNEJ MODELU SILNIKA SKOKOWEGO.

Przy analizie numerycznej napędu dyskretnego należy wybrać taki model, który zapewnia wymagany stopień dokładności i wybrane uproszczenia. Przy konstruowaniu programu symulacji napędu dyskretnego przy sterowaniu miniskokowym wybrano model matematyczny silnika hybrydowego wg pracy [63] opracowany dla jednostek względnych. Równanie napędu obejmuje jedno równanie różniczkowe rzędu drugiego opisujące model matematyczny silnika. Sterowanie miniskokowe jest uwzględnione w opisie funkcji sterującej.

$$(4.13) \quad J \frac{d^2\theta_e}{d\tau^2} + 2\beta \frac{d\theta_e}{d\tau} + \mu_r \operatorname{sign}\left(\frac{d\theta_e}{d\tau}\right) = \sin \delta;$$

gdzie:

$\tau = \omega_0 t$ czas względny;

$\beta = \beta_e + \beta_m$ całkowity względny wsp. tłumienia
lepkiego;

$$\beta_e = \omega_o \frac{k_U}{2 Z^r U} \quad \text{wzgl. wsp. tłumienia lepkiego elektro-}$$

magnetycznego;

$$\beta_m = \frac{1}{\omega_o} \frac{D}{2J} \quad \text{wzgl. wsp. tłumienia lepkiego zewnętrznego;}$$

$$\mu_r = \frac{M_r R}{k_U U} \quad \text{wzgl. moment obciążenia;}$$

$$\omega_o = \sqrt{\frac{k_U Z^r U}{RJ}} \quad \text{częstotliwość drgań własnych;}$$

$$\delta = \frac{\pi k}{2N} - \theta_e \quad \text{funkcja sterująca;}$$

k- numer miniskoku 0..N;

N- liczba miniskoków w skoku bazowym silnika.

Przy opracowaniu programu analizującego zostały wykorzystane także prace [98], [32], [39], [58], [47], [72], [38].

Na podstawie równania i założeń określono warunki początkowe układu równań:

$$\tau(0) = 0;$$

$$\theta(0) = 0;$$

$$\frac{d\theta}{d\tau} = 0.$$

Do symulacji komputerowej zastosowano następujące podstawienia:

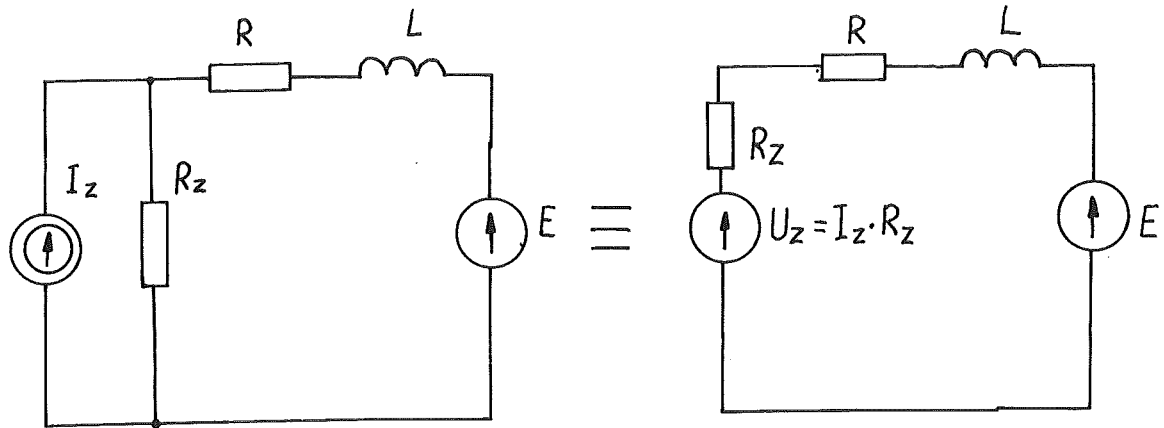
$$\begin{aligned}x[1] &= \tau; \\x[2] &= \theta_e; \\x[3] &= \frac{d\theta_e}{d\tau}.\end{aligned}$$

Równania modelu przedstawiają się następująco:

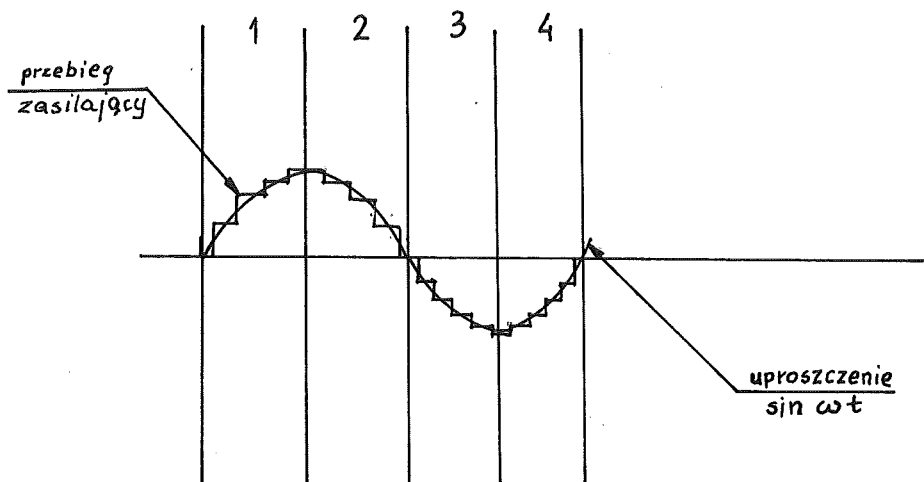
$$\begin{aligned}xprim[1] &= 1; \\xprim[2] &= x[3]; \\xprim[3] &= \frac{\sin\delta - 2\beta x[3] - \mu_r \text{sign}(x[3])}{J}.\end{aligned}$$

Powyższe równania zostały przekształcone do postaci zgodnej z językiem Pascal i zastosowane w programie symulacji komputerowej, którego część obliczeniowa zamieszczona jest w dodatku A. Opisany model pozwala uzyskać charakterystyki czasowe i fazowe napędu dyskretnego przy różnych warunkach pracy.

Do uzyskania charakterystyk momentu w funkcji częstotliwości impulsów sterujących napęd zastosowano metodykę opisaną w pracy [48]. Po przekształceniu obwodu zasilania pasma silnika, tak aby źródło prądowe zastąpić równoważnym źródłem napięciowym, zastosowano wzór dla obliczania charakterystyki momentu silnika dwupasmowego, zasilanego bipolarnie. Schodkowe przebiegi zasilające silnik zostały zastąpione ich rozwinięciem w szereg Fouriera z uwzględnieniem tylko pierwszej harmonicznej. Przekształcenie obwodu pasma silnika oraz przebiegu zasilającego zostało zilustrowane na rys. 4.1 i 4.2.



Rys. 4.1. Przekształcenie obwodu zasilania pasma silnika



Rys. 4.2. Uproszczenie kształtu przebiegów zasilających

Wzór na względny moment rozwijany przez silnik ma postać:

$$\mu = \frac{1}{\sqrt{1 + \omega^2 \cdot \left(\frac{L_o}{R}\right)^2}} - k_r \cdot \frac{\omega \cdot \left(\frac{L_o}{R}\right)}{1 + \omega^2 \cdot \left(\frac{L_o}{R}\right)^2}$$

gdzie $k_r = \frac{k_u R}{Z^r L_o U}$,

$$\omega = \frac{\pi f_i}{2 N}$$

f_i - częstotliwość impulsów ster.
 N - liczba miniskoków w skoku bazowym.

Przedstawiona powyżej zależność pozwoli określić charakterystyki częstotliwościowe napędu i porównać je z wynikami badań.

5. STEROWANIE SILNIKA SKOKOWEGO PRZY PRACY MINISKOKOWEJ

5.1. SYNTEZA FUNKCJI STERUJĄCYCH DO PRACY MINISKOKOWEJ

Sterowanie miniskokowe wymaga zastosowania do zasilania silnika przebiegów o odpowiednim kształcie. Podstawowym wymaganiem jest uzyskanie stałości maksymalnego momentu synchronizującego. Do uzyskania funkcji sterowania dla pracy miniskokowej należy zastosować funkcje spełniające następujący warunek:

$$F^2(t) + G^2(t) = \text{const}$$

gdzie: $F(t)$ i $G(t)$ są funkcjami okresowymi opisującymi przebiegi sterujące pasma silnika.

Takimi funkcjami mogą być przede wszystkim funkcje trygonometryczne sinus i cosinus. Do celów sterowania miniskokowego stosuje się funkcje sinus i cosinus przybliżone krzywą schodkową. Funkcje sterowania dla różnych dwóch silników skokowych przedstawione są poniżej:

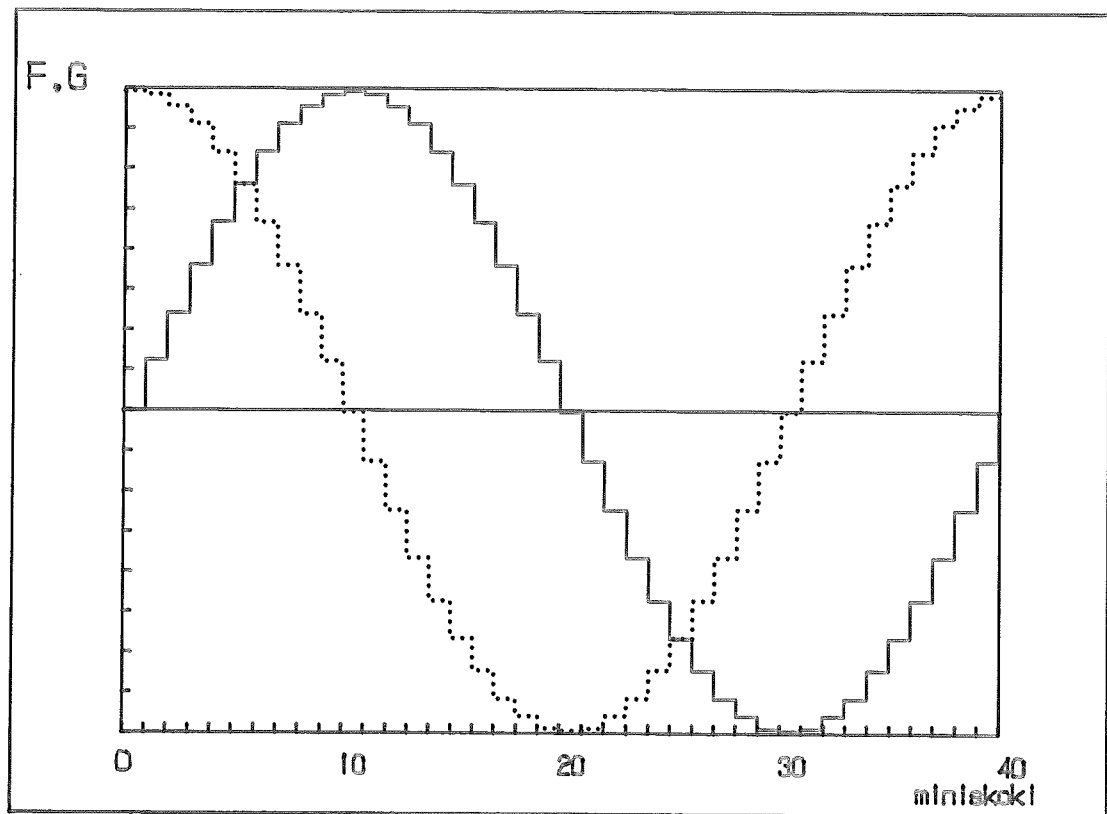
a) silnik dwupasmowy - sterowanie bipolarne

$$F(t) = F_{\max} \sin\left(\frac{\pi k}{2N}\right)$$

$$G(t) = G_{\max} \cos\left(\frac{\pi k}{2N}\right)$$

b) silnik czteropasmowy - sterowanie unipolarne

$$F_1(t) = F_{\max} \left(0.5 + 0.5 \sin\left(\frac{\pi k}{2N}\right) \right)$$
$$G_1(t) = G_{\max} \left(0.5 + 0.5 \sin\left(\frac{\pi k}{2N}\right) \right)$$



Rys. 5.1.

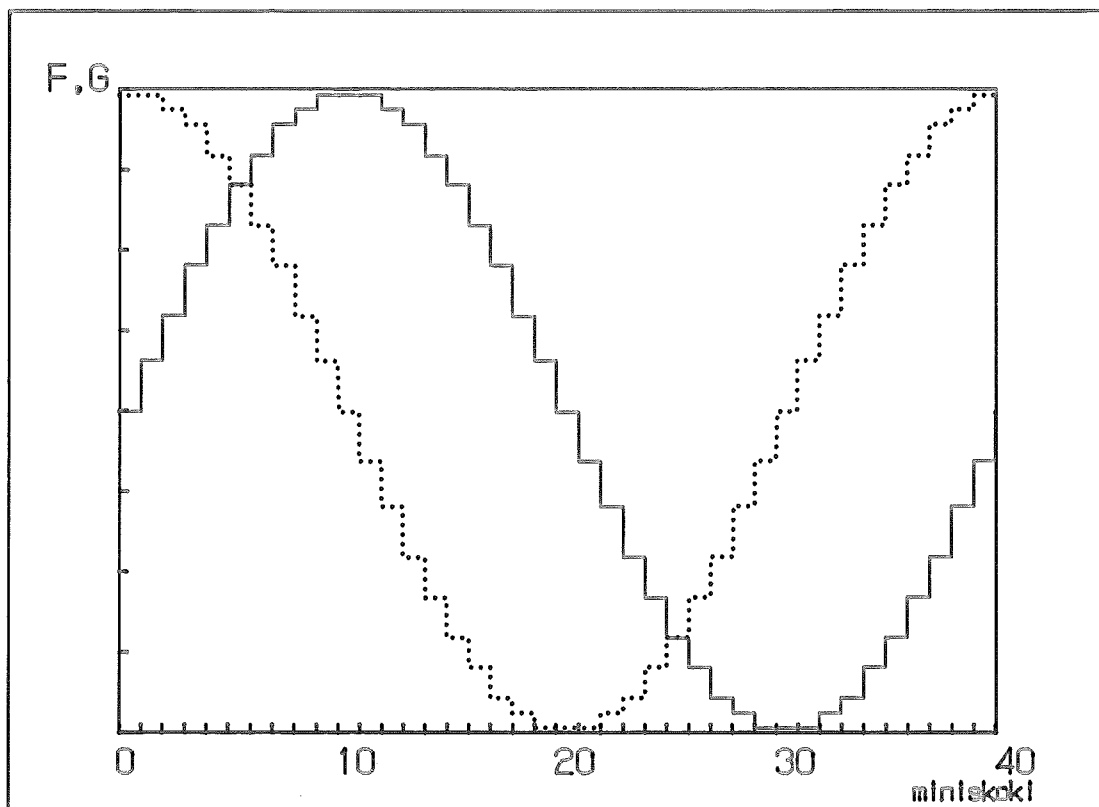
Przebiegi sterujące dla silnika dwupasmowego

$$F_2(t) = F_{\max} - F_1(t)$$

$$G_2(t) = G_{\max} - G_1(t).$$

W powyższych wzorach 'N' oznacza liczbę miniskoków w skoku bazowym, a 'k' kolejny numer miniskoku (k=1..N).

Na rys. 5.1 i 5.2 zostały przedstawione przebiegi tych funkcji dla N=10.



Rys. 5.2.

Przebiegi sterujące dla silnika czteropasmowego

Opisywane funkcje mogłyby znaleźć zastosowanie w praktyce, przy założeniu, że istnieje możliwość przetworzenia wartości cyfrowej na analogową z bardzo dużą dokładnością. Niestety tak nie jest. Produkowane w obecnej chwili przetworniki cyfrowo-analogowe mają ograniczoną rozdzielczość, a ich cena rośnie progresywnie wraz ze wzrostem dokładności przetwarzania. Należy wybrać rozsądny kompromis między wymaganiami technicznymi i ekonomicznymi.

Zastosowanie dyskretnego przetwarzania powoduje powstanie błędów w generacji funkcji sterujących. Można obliczyć dyskretnie funkcje sterowania oraz wartości błędów przetwarzania według poniższych wzorów:

$$F_d(t) = \text{Round}\left(M \cdot \sin\left(\frac{\pi k}{2N}\right)\right)$$
$$G_d(t) = \text{Round}\left(M \cdot \cos\left(\frac{\pi k}{2N}\right)\right).$$

Stosując te wzory zakłada się, że wartości M odpowiada wartość maksymalna ciągłej funkcji sterowania F_{\max} , G_{\max} . Przy takim założeniu można określić wzór na błąd względny generowania funkcji sterowania:

$$\delta_d = \frac{\left| \frac{\pi k}{2N} - \arctan\left(\frac{F_d(t)}{G_d(t)}\right) \right|}{\frac{\pi}{2N}} \cdot 100\%.$$

Określony powyżej błąd określa procentową wartość błędu odniesionego do wartości miniskoku.

Przeprowadzono obliczenia wartości błędu dla liczby

miniskoków równej 10,20,50,100 i długości słowa przetwornika C/A w granicach 8 - 16 (co odpowiada wartości M od 255 do 65535).

Długość słowa przetwornika	Maksymalny błąd względny
----------------------------------	--------------------------------

Liczba miniskoków =10

8	0.7514760 %
9	0.1842750 %
10	0.2905663 %
11	0.1127787 %
12	0.0831204 %
13	0.0334669 %
14	0.0158235 %
15	0.0097472 %
16	0.0034713 %

Liczba miniskoków =20

8	2.2342781 %
9	1.1305663 %
10	0.6767196 %
11	0.3800825 %
12	0.1662408 %
13	0.0928467 %
14	0.0412139 %
15	0.0194944 %
16	0.0088227 %

Liczba miniskoków =50

8	8.2231298 %
9	3.3788424 %
10	1.8637091 %
11	0.8923697 %

12	0.4156020 %
13	0.2278977 %
14	0.1173687 %
15	0.0624247 %
16	0.0235595 %

Liczba miniskoków =100

8	16.4462595 %
9	7.4547182 %
10	3.7607158 %
11	1.9004126 %
12	0.9162819 %
13	0.4642337 %
14	0.2347374 %
15	0.1248495 %
16	0.0552619 %

Z powyższych obliczeń wynika, że błąd $< 2\%$ uznany za standardowy w literaturze jest osiągnięty dla długości słowa przetwornika ≥ 11 bitów.

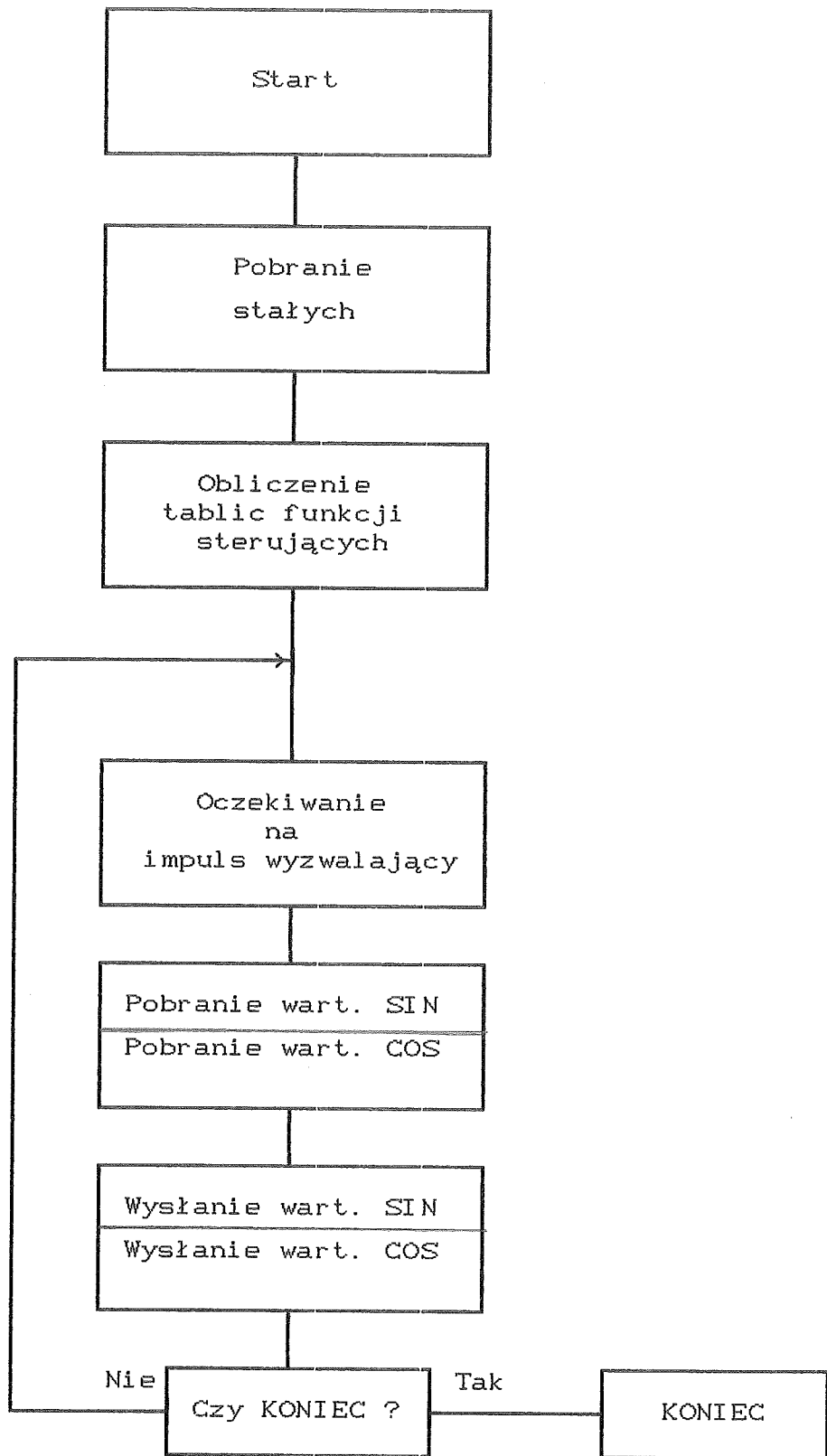
Powyższe wzory pozwalają obliczyć wartości funkcji sterujących dla wszystkich miniskoków w obrębie skoku bazowego. Funkcje są okresowe i w związku z tym w czasie następnym skoków bazowych można stosować te same wartości funkcji.

5.2. ALGORYTMY STEROWANIA MINISKOKOWEGO

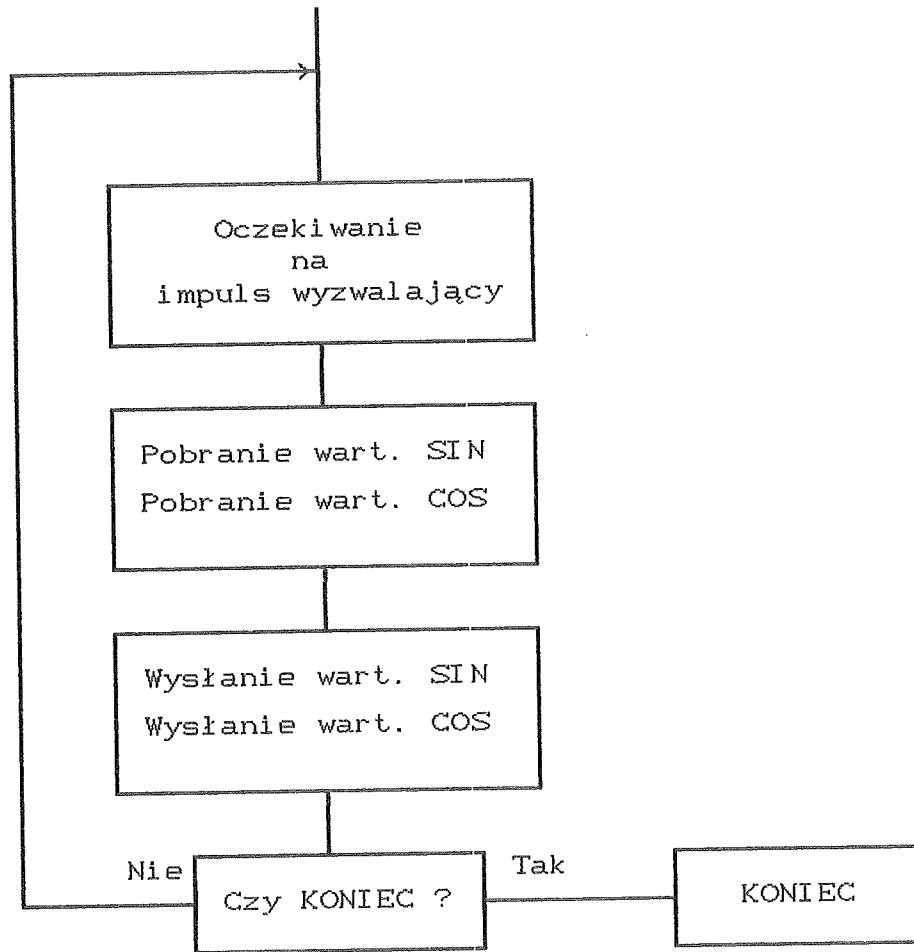
Do celów badawczych opracowano kilka algorytmów sterowania miniskokowego, przystosowanych do różnych rozwiązań sprzętowych. Podstawową różnicą między nimi jest

sposób rozmieszczenia w pamięci mikrokomputera zawartości tablic określających kolejne wartości napięć lub prądów zasilających pasma silnika. Przykładowo zostaną podane dwa algorytmy sterowania miniskokowego, z których pierwszy pobiera kolejne wartości sterowania umieszczone w dwóch oddzielnych obszarach pamięci. Tablice funkcji sterujących są oddzielne dla funkcji \sin i \cos . Takie podejście upraszcza konstrukcję programu sterującego. Graficzny obraz algorytmu jest podany na rys. 5.3.

W dążeniu do zwiększenia szybkości działania programu sterującego został opracowany drugi algorytm, w którym wartości obydwu funkcji sterujących są zapamiętane w jednej tablicy. Wartości funkcji \sin i \cos zapamiętane są kolejno w komórkach pamięci. Taki sposób rozmieszczenia funkcji w pamięci mikrokomputera umożliwia szybsze ich pobieranie i przekazywanie do sterownika. Program działający w oparciu o algorytm drugi jest o ok. 50% szybszy w stosunku do programu wg. algorytmu 1. Schemat algorytmu jest przedstawiony na rys. 5.4.



Rys. 5.3 Algorytm 1



Rys. 5.4 Algorytm 2

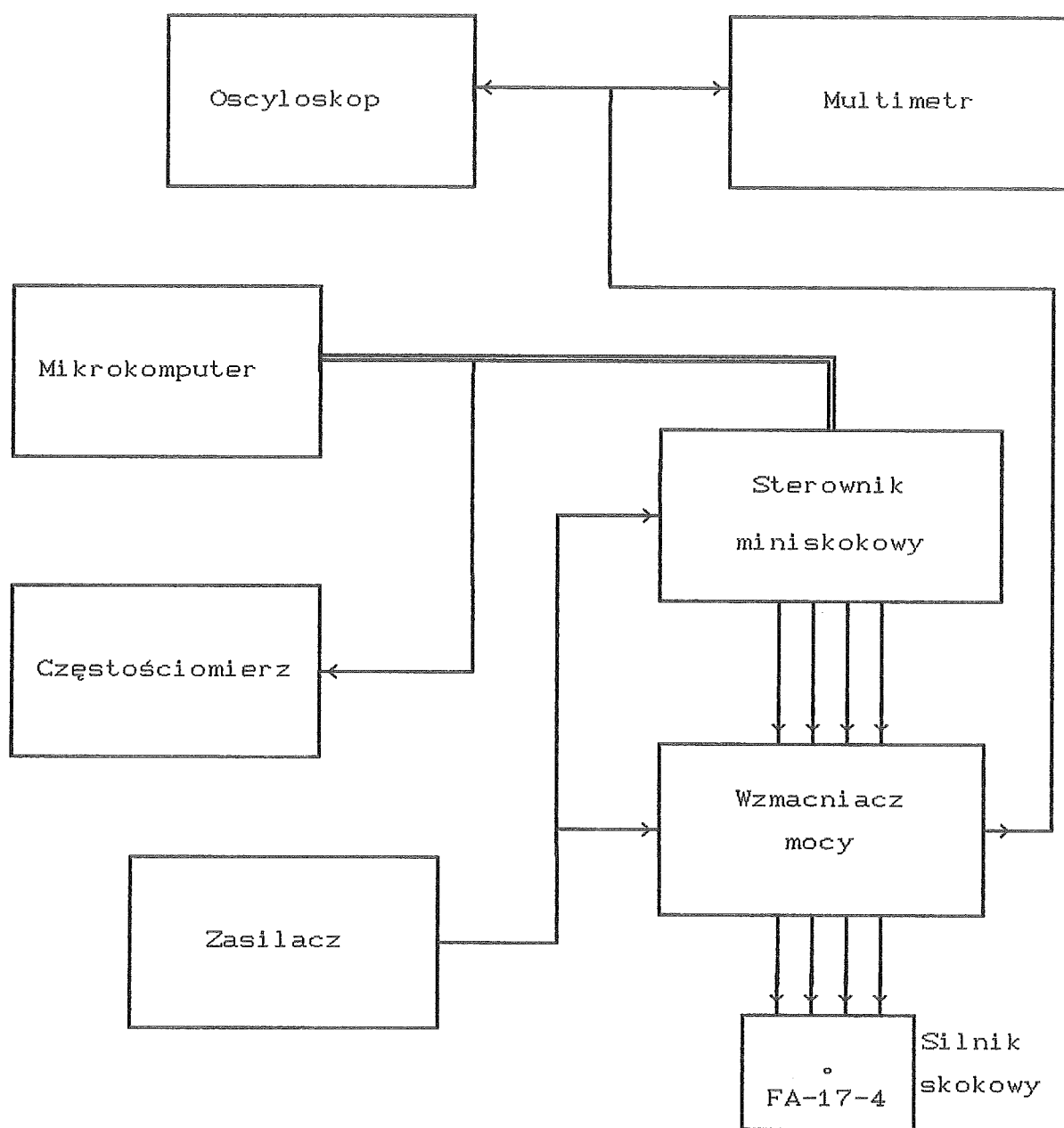
5.3. STANOWISKO DO BADANIA SILNIKÓW SKOKOWYCH PRZY STEROWANU MINISKOKOWYM

W celu wykonania badań silników skokowych podczas pracy miniskokowej zostało zaprojektowane i zbudowane odpowiednie stanowisko badawcze. Zespół urządzeń wchodzący w skład stanowiska zapewnia uzyskanie dostatecznej dokładności pomiarów dla wszystkich przewidywanych mierzonych

charakterystyk i funkcji sterujących. Umożliwia badanie charakterystyk mechanicznych i kątowych przy sterowaniu miniskokowym dla różnej liczby miniskoków. Zestaw badawczy składa się z następujących elementów:

- mikrokomputer Amstrad CPC 6128;
- kaseca urządzeń dodatkowych;
- sterownik miniskokowy;
- cyfrowy wzmacniacz mocy;
- zasilacz 0 - 20 V, typ P-322;
- woltomierz cyfrowy V-544;
- multimetr cyfrowy G-1002.500;
- częstościomierz KZ-2026A-2;
- oscyloskop DT-516A;
- laser Ne-He;
- lusterko pomiarowe.

Schemat blokowy stanowiska jest przedstawiony na rys. 5.5. W przedstawionym stanowisku głównym elementem jest mikrokomputer, steruje on całością procesu generowania sygnałów sterowania miniskokowego. Sygnały z mikrokomputera przekazywane są do sterownika miniskokowego umieszczonego w kasecie urządzeń dodatkowych, gdzie następuje przekształcenie sygnałów cyfrowych na przebiegi sterujące silnik. Przebiegi te są wzmacniane we wzmacniaczu mocy i już bezpośrednio zasilają silnik. Do obciążania silnika służy hamulec tarcowy, a do pomiaru kąta zastosowano układ optyczny z wykorzystaniem lusterka i promienia lasera.



Rys 5.5 Schemat blokowy stanowiska badawczego

5.4. BADANIA PRACY MINISKOKOWEJ SILNIKA.

Badania silnika skokowego przy pracy miniskokowej obejmują zbadanie charakterystyk mechanicznych oraz

dokładności pozycjonowania dla wybranego silnika skokowego typu FA-17-4-1. Parametry tego silnika przedstawione są poniżej:

kąt skoku bazowego: 3.6° ;
napięcie zasilania: 12 V;
maksymalny moment synchronizujący: 50 mNm;
moment bezwładności silnika: $1.6 \cdot 10^{-6} \text{ kg} \cdot \text{m}^2$;
rezystancja pasma: 74 Ω ;
indukcyjność średnia pasma: 35 mH;
amplituda indukcyjności pasma: 3 mH;
współczynnik napięciowy k_U : $0.308 \text{ V} \cdot \text{s} \cdot \text{rad}^{-1}$.

Silnik ten został wybrany z tego względu, że jest to najpowszechniej używany silnik w napędach małej mocy dla celów optoelektroniki, ma dosyć dużą liczbę skoków na obrót, i moment wystarczający dla zastosowań związanych z tematyką prac badawczych autora.

1. Badanie charakterystyk częstotliwościowych (rozruchowych i maksymalnych częstotliwości pracy) silnika, przy zmiennym momencie obciążenia, dla różnej liczby miniskoków: 10, 20, 50, 100 na skok bazowy.

2. Pomiar dokładności pozycjonowania przy różnej liczbie miniskoków: 5, 10, 20, 50, 100 na skok bazowy.

Badanie charakterystyk mechanicznych silnika zostały

przeprowadzone przy pomocy stanowiska opisanego w 5.3. Pomiar momentu został przeprowadzony przy pomocy hamownicy tarciowej. Proces pomiaru został przeprowadzony w następujący sposób:

- wyznaczenie zależności maksymalnej częstotliwości pracy od momentu obciążenia: regulacja częstotliwości skoków od 10 Hz do wartości przy której silnik wypadnie z synchronizmu;
- wyznaczenie zależności maksymalnej częstotliwości rozruchowej od momentu obciążenia: przy zatrzymanym silniku ustawia się moment obciążenia i następnie uruchamia się silnik przy zadanej częstotliwości. Następnie zatrzymuje się silnik i zwiększa obciążenie. Proces powtarza się do momentu gdy silnik nie ruszy po załączeniu impulsów sterujących.

Badania charakterystyk mechanicznych zostały przeprowadzone dla sterowania standardowego typu 1-2 i 1-4 oraz sterowania miniskokowego dla $N=10, 20, 50, 100$:

- badania dokładności pozycjonowania dla pkt.2 przeprowadza się badając dokładność odwzorowania określonej liczby skoków. Bada się odwzorowanie jednokierunkowego ruchu o zadanej liczbie skoków w zakresie pełnego obrotu wału, oraz dla cyklu rewersyjnego przy liczbie skoków jak poprzednio;
- badanie dokładności pozycjonowania w zakresie skoku bazowego przeprowadza się przy pomocy lustra umocowanego na wale silnika i lasera. Po oświetleniu

lustro promieniem laserowym na ekranie oddalonym od silnika o 5.15 m, uzyskuje się odwzorowanie położenia wału. Wartość kąta przesunięcia wału oblicza się przy pomocy odpowiednich wzorów trygonometrycznych;

-badanie przesunięcia kąтового wału silnika w funkcji czasu przeprowadza się przy użyciu dodatkowego silnika synchronicznego z umieszczonym na wale lustrem, którego oś jest prostopadła do osi silnika badanego. Silnik ten realizuje odchylenie pionowe promienia laserowego. Pozwala to na ekranie otrzymać odwzorowanie przesunięcia wału silnika badanego w funkcji czasu.

Badania te przeprowadza się dla sterowania miniskokowego dla $N=10, 20, 50, 100$.

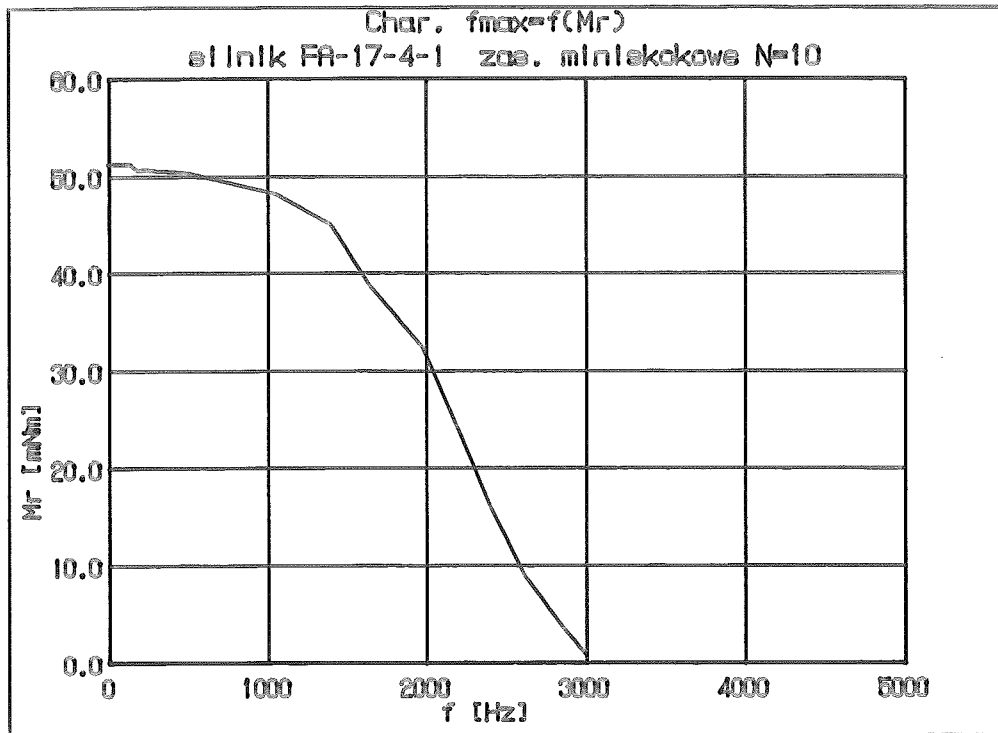
6. WPLYW STEROWANIA MINISKOKOWEGO NA CHARAKTERYSTYKI SILNIKA

6.1. CHARAKTERYSTYKI MECHANICZNE SILNIKA PRZY PRACY MINISKOKOWEJ.

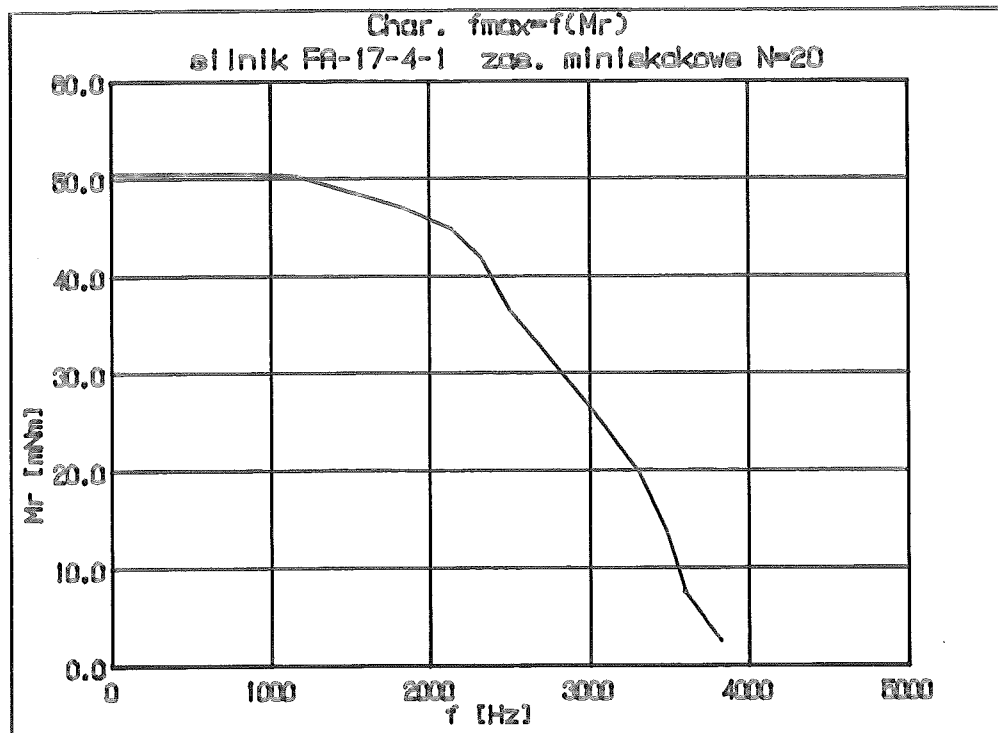
Przeprowadzone badania charakterystyk mechanicznych silnika FA-17-4-1 pozwoliły określić charakterystyki funkcji momentu silnika od częstotliwości impulsów wejściowych układu sterowania. Dostępny sterownik miniskokowy umożliwił przeprowadzenie badań w zakresie 0-5000 Hz przy zasilaniu prądowym o amplitudzie równej maksymalnej wartości prądu zasilania pasma silnika. Na rysunkach 6.1-6.4 zostały przedstawione charakterystyki rozruchowe tego silnika. Dla porównania na rys. 6.5 przedstawiono charakterystykę rozruchową dla sterowania typu 1-4. Na rys. 6.6 - 6.10 przedstawiono analogiczne wykresy dla częstotliwości rozruchowych.

Wyraźnie można zaobserwować wzrost maksymalnej częstotliwości rozruchowej w funkcji liczby miniskoków. Zmienia się również charakter charakterystyki. Staje się ona coraz bardziej zbliżona do prostokątnej.

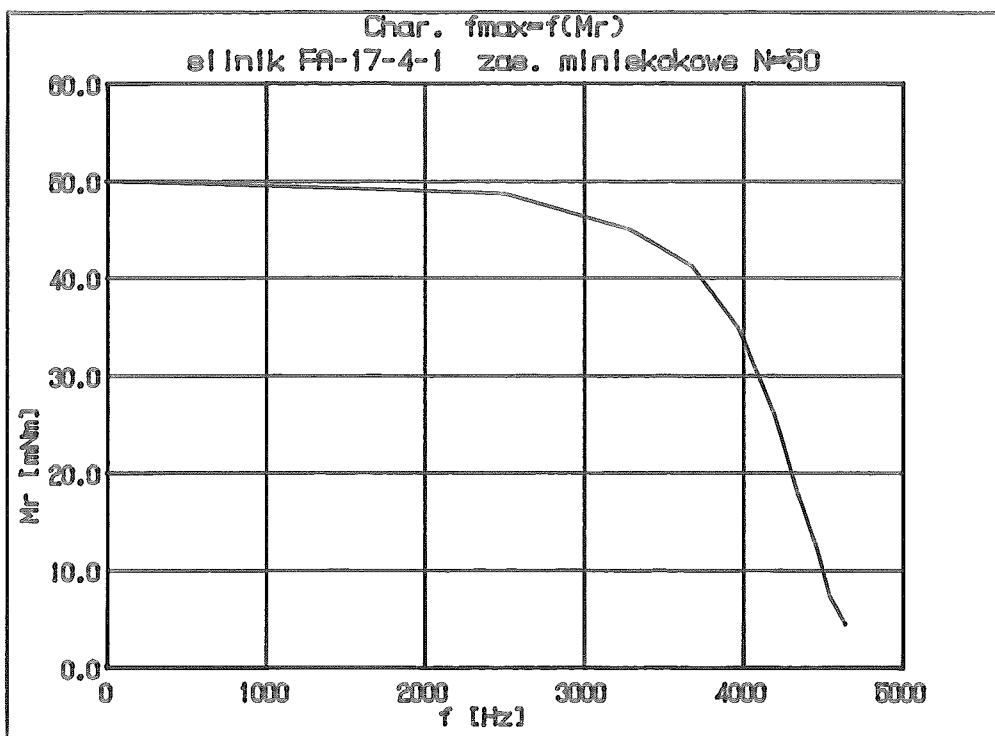
Zostały także przeprowadzone pomiary dokładności pozycjonowania w funkcji kąta (w zakresie jednego skoku bazowego). Wyniki tych badań zostały przedstawione na rys. 6.11-6.14. Przeprowadzono także badanie przesunięcia kąтового wału silnika w funkcji czasu dla różnej liczby miniskoków w



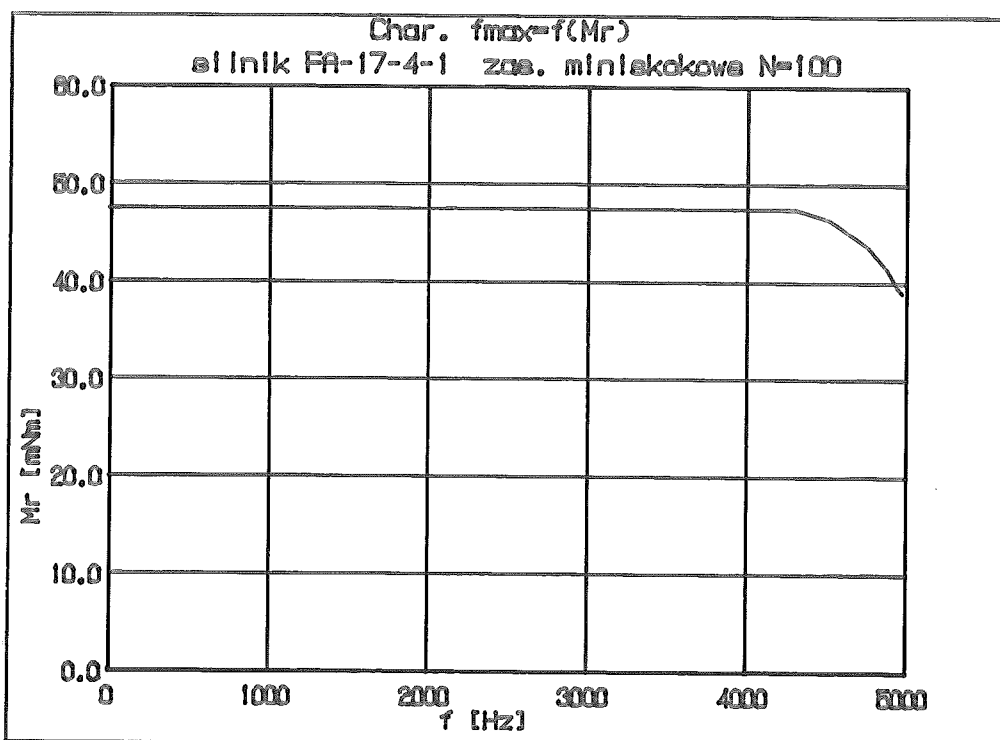
Rys. 6.1



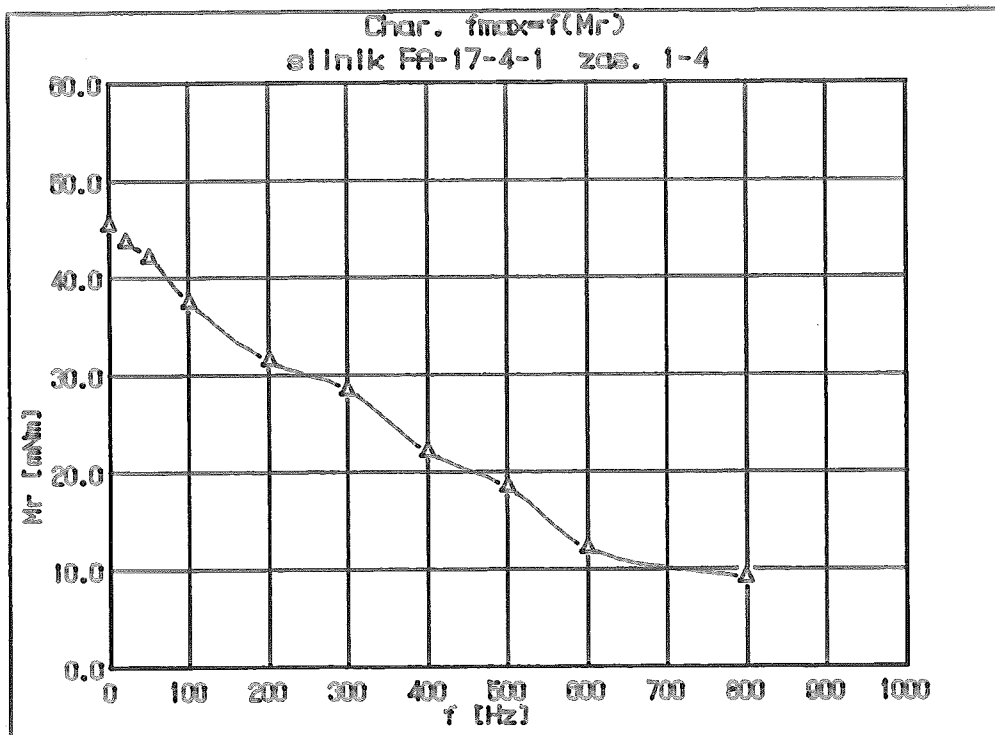
Rys. 6.2



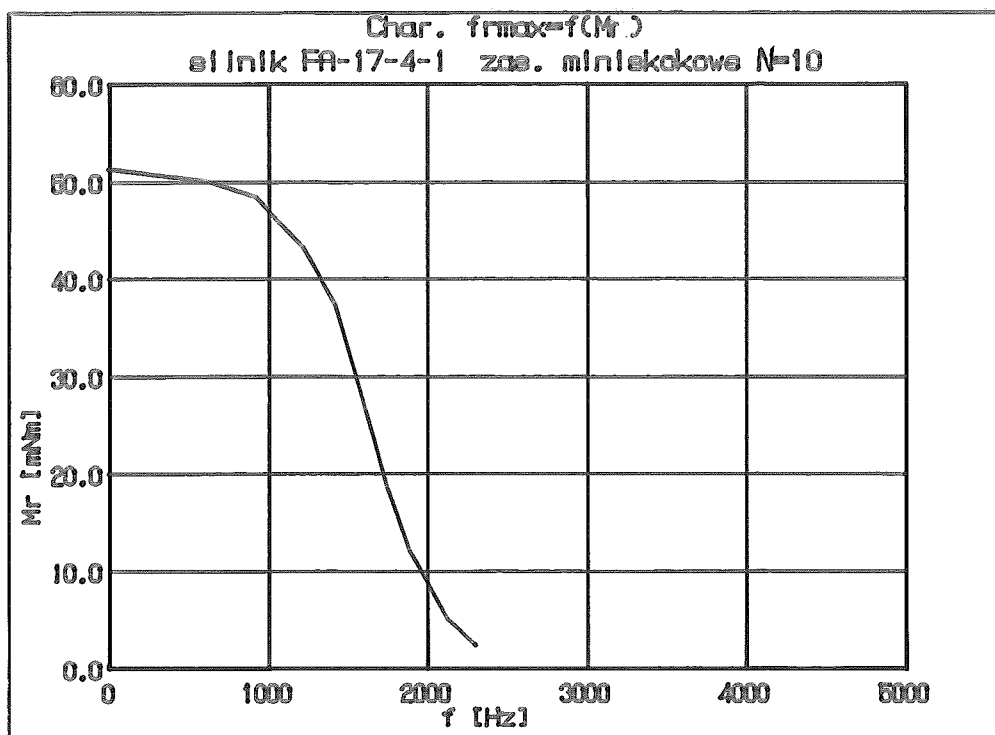
Rys. 6.3



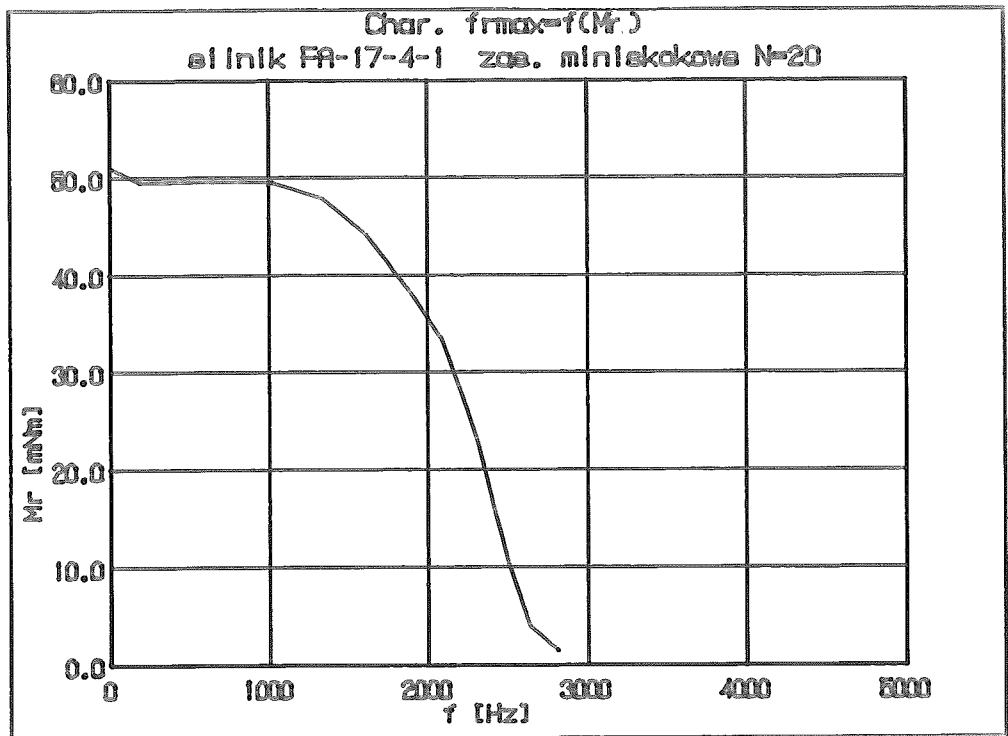
Rys. 6.4



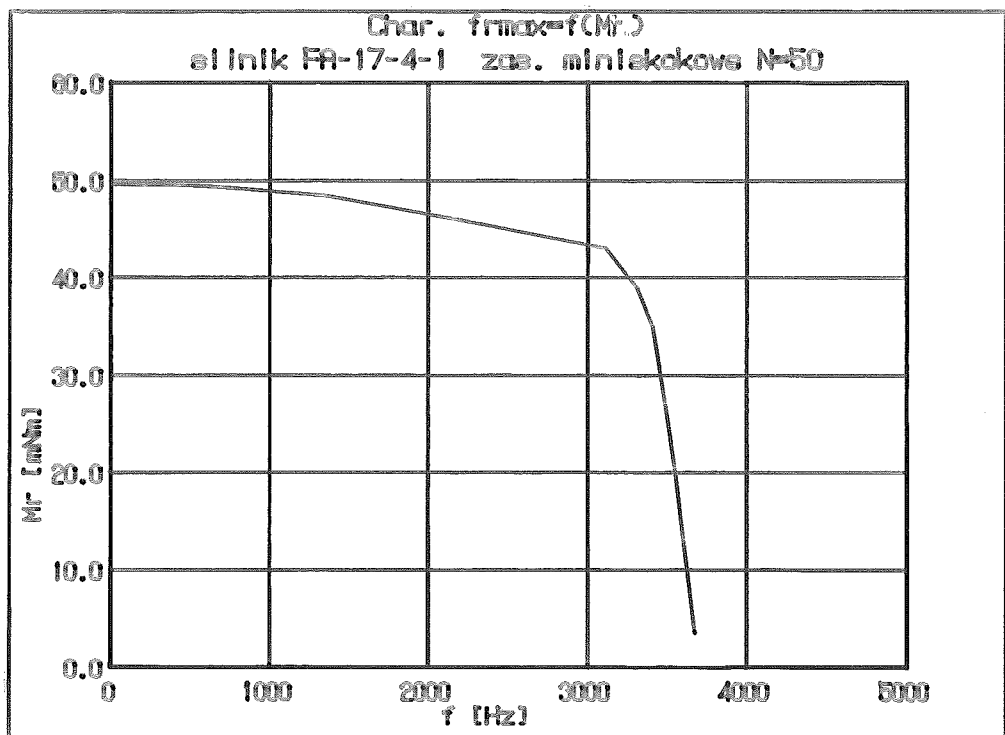
Rys. 6.5



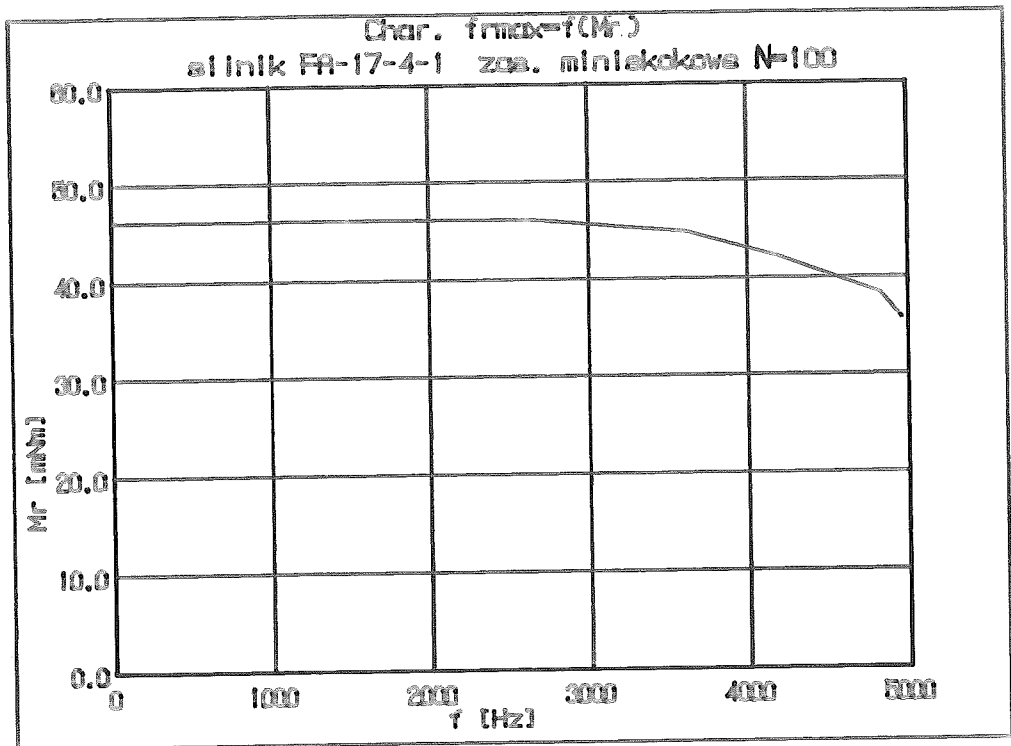
Rys. 6.6



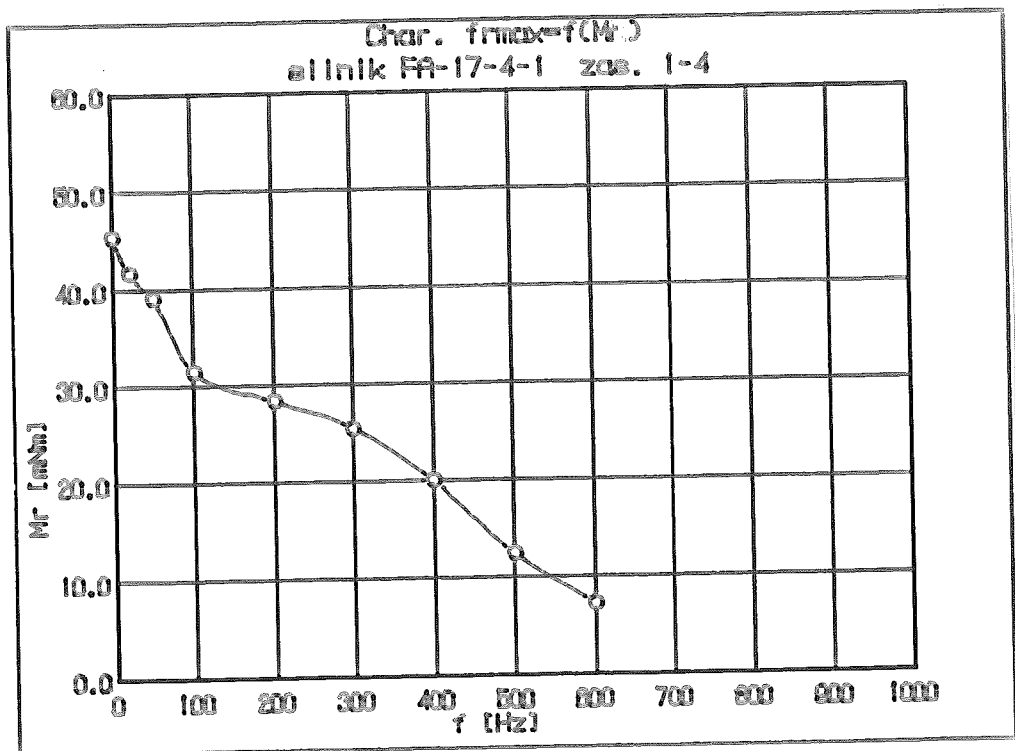
Rys. 6.7



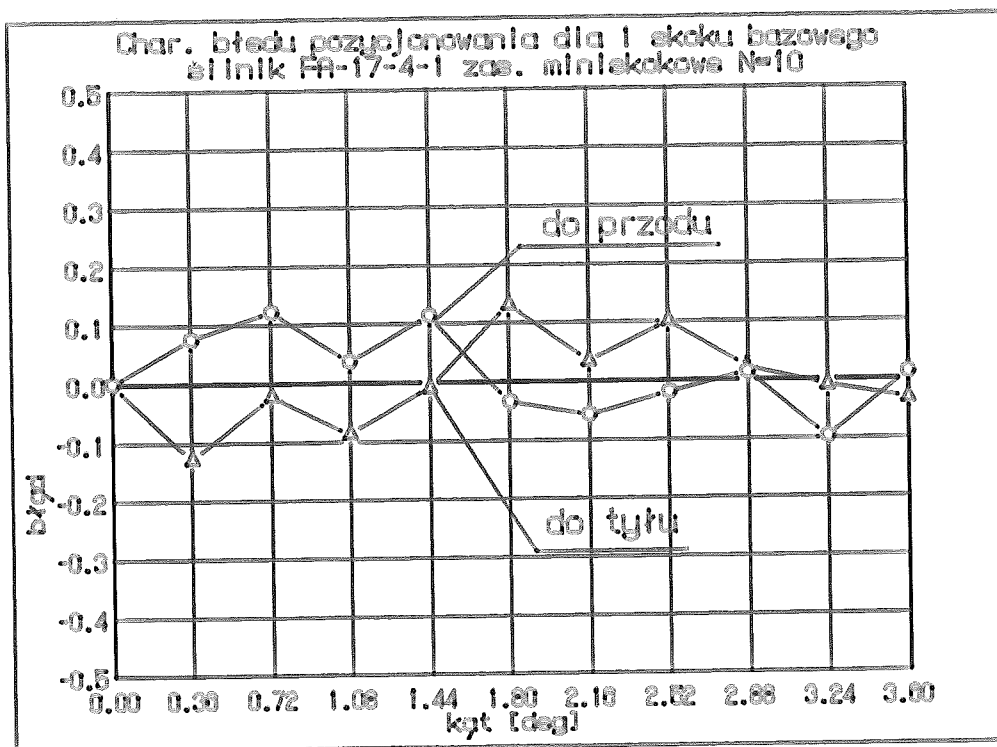
Rys. 6.8



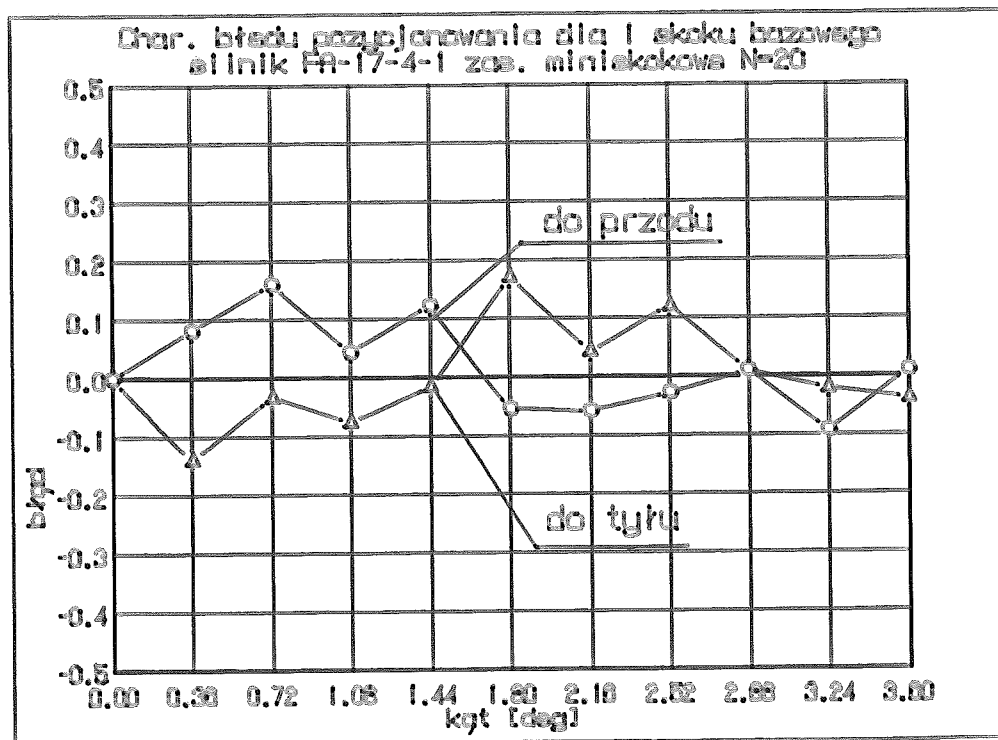
Rys. 6.9



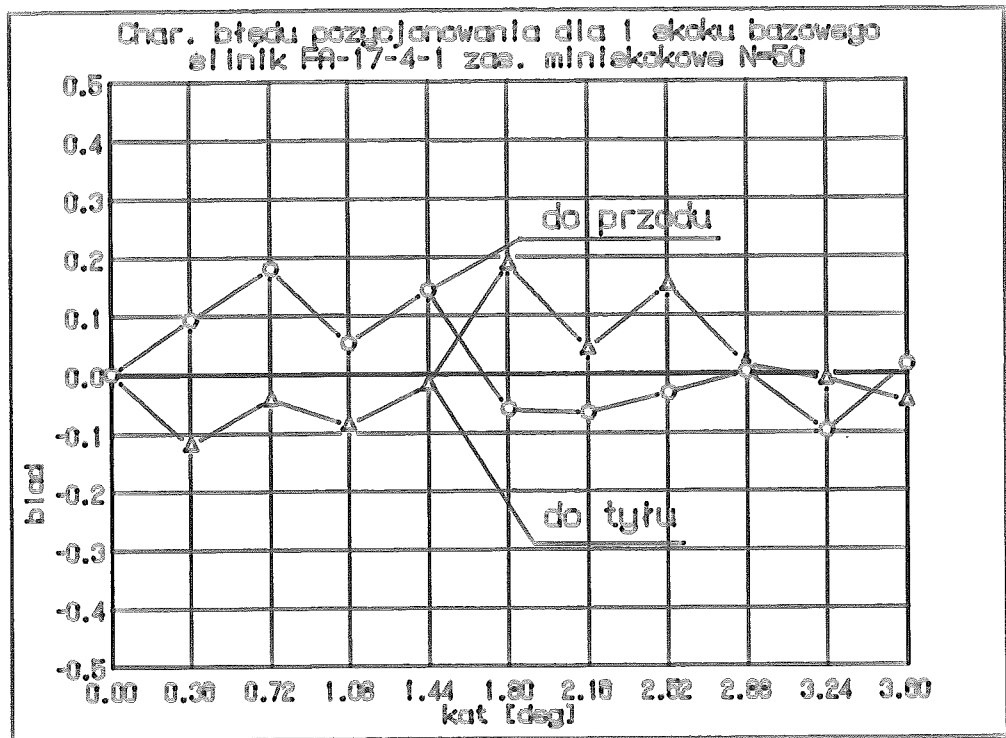
Rys. 6.10



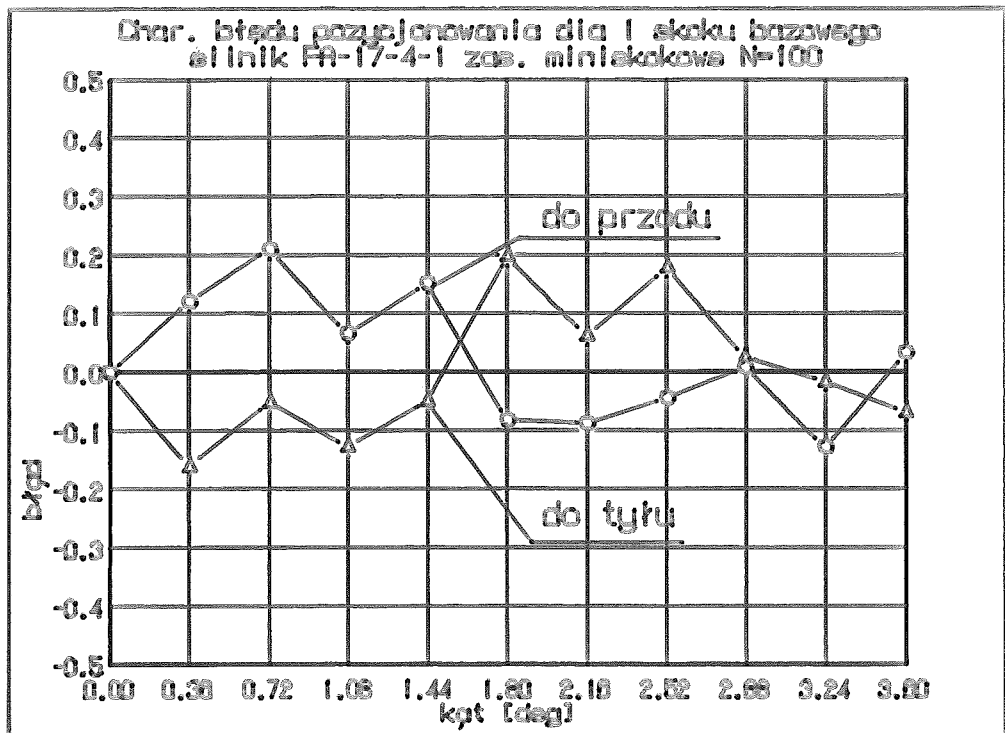
Rys. 6.11



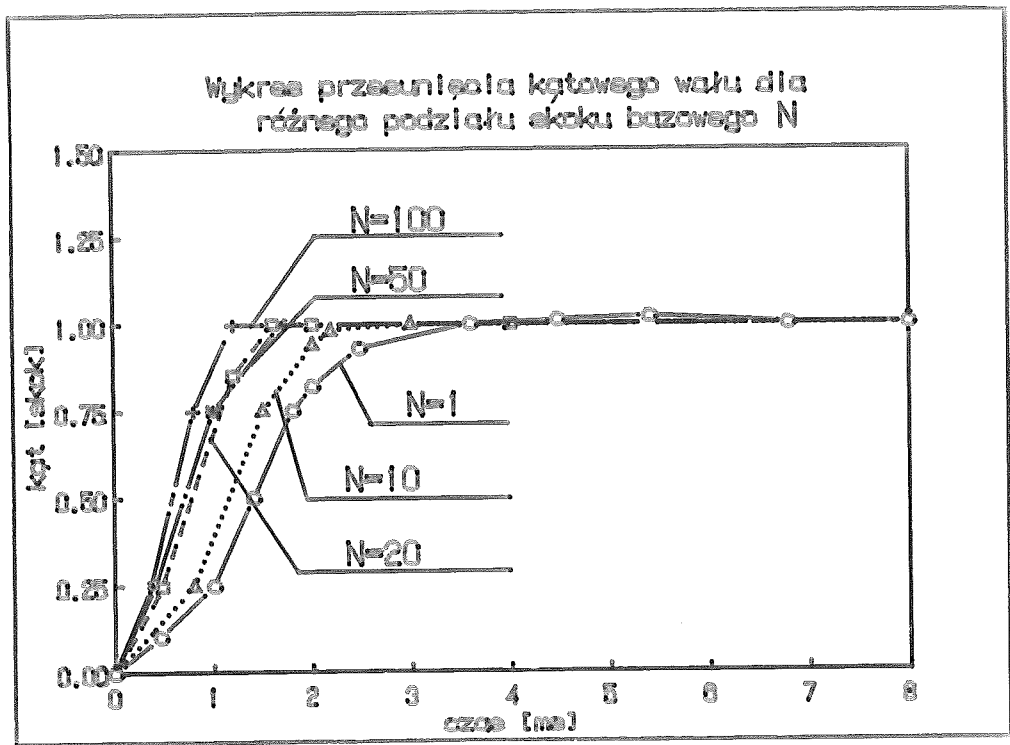
Rys. 6.12



Rys. 6.13



Rys. 6.14



Rys. 6.15

skoku bazowym. Wyniki tych pomiarów zostały przedstawione na rys. 6.15.

6.2. ANALIZA WYNIKÓW BADAŃ

Badania charakterystyk mechanicznych napędu przy pracy miniskokowej pozwalają sformułować następujące wnioski:

1. Przy wzroście liczby miniskoków w skoku bazowym silnika wzrasta maksymalna częstotliwość pracy napędu. Wzrost ten nie jest liniowy, co wynikałoby z teorii, lecz częstotliwość maksymalna wzrasta wolniej. Wynika to przede wszystkim z niedoskonałości sterownika miniskokowego jako idealnego źródła prądowego. Rzeczywiste źródło prądowe nie może wygenerować nieskończonego wielkiego napięcia wymuszającego.
2. Charakterystyki mechaniczne napędu, wraz ze wzrostem liczby miniskoków, stają się coraz bardziej zbliżone do charakterystyki silnika synchronicznego. Jest to zgodne z oczekiwaniami. Przy zwiększaniu liczby miniskoków, maleje zawartość harmonicznych w sygnale zasilającym silnik, a co za tym idzie sygnał ten zbliża się do sinusoidy analogicznie do sygnału zasilającego silnik synchroniczny. Zjawisko to jest bardzo korzystne z punktu widzenia zastosowań, gdyż zwiększa się zakres płaskiego przebiegu charakterystyki.
3. Przy zwiększaniu liczby miniskoków zmniejsza się czas dojścia wirnika do wymaganego położenia. Wynika to ze

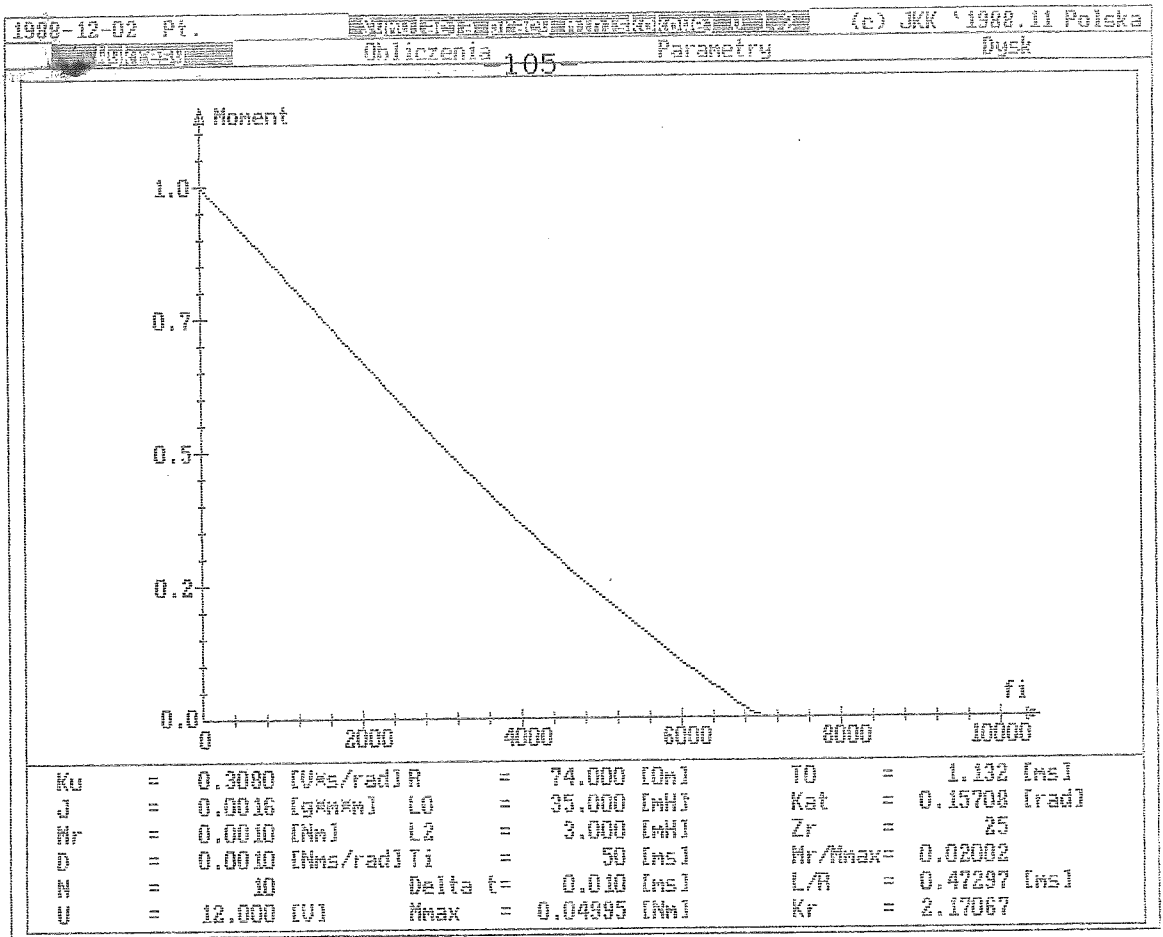
zmniejszenia się energii potrzebnej do przejścia z jednego położenia stabilnego do drugiego. Mniejsze zmiany sygnału zasilającego przy zmianie położenia powodują znaczne zmniejszenie zjawisk rezonansowych i problemów związanych z indukcyjnością pasm.

4. Przy zasilaniu miniskokowym zanikają "siódła" na charakterystyce mechanicznej wynikające z występowania rezonansów mechanicznych napędu.

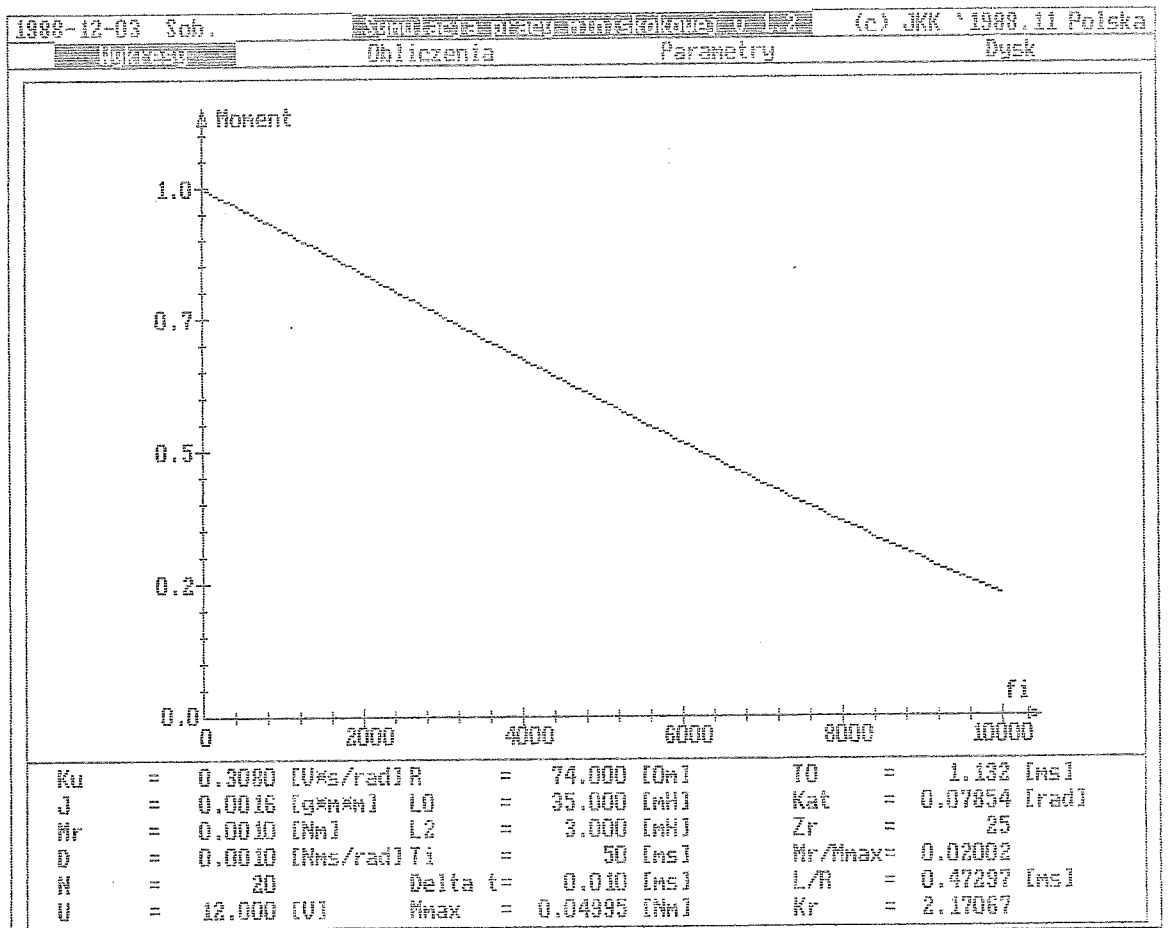
6.3. PORÓWNANIE WYNIKÓW BADAŃ EKSPERYMENTALNYCH Z SYMULACJĄ KOMPUTEROWĄ

Przeprowadzono symulację komputerową napędu dyskretnego przy pomocy programu opisanego w dodatku A. Badania przeprowadzono przy stałych warunkach obciążenia zbliżonych do uzyskiwanych w badaniach eksperymentalnych. Przeprowadzono symulację ruchu wirnika przy zasilaniu miniskokowym oraz przy zasilaniu 1-4, przy takich samych parametrach napędu jak w badaniach eksperymentalnych. Przeprowadzono także obliczenia charakterystyk mechanicznych dla identycznych warunków.

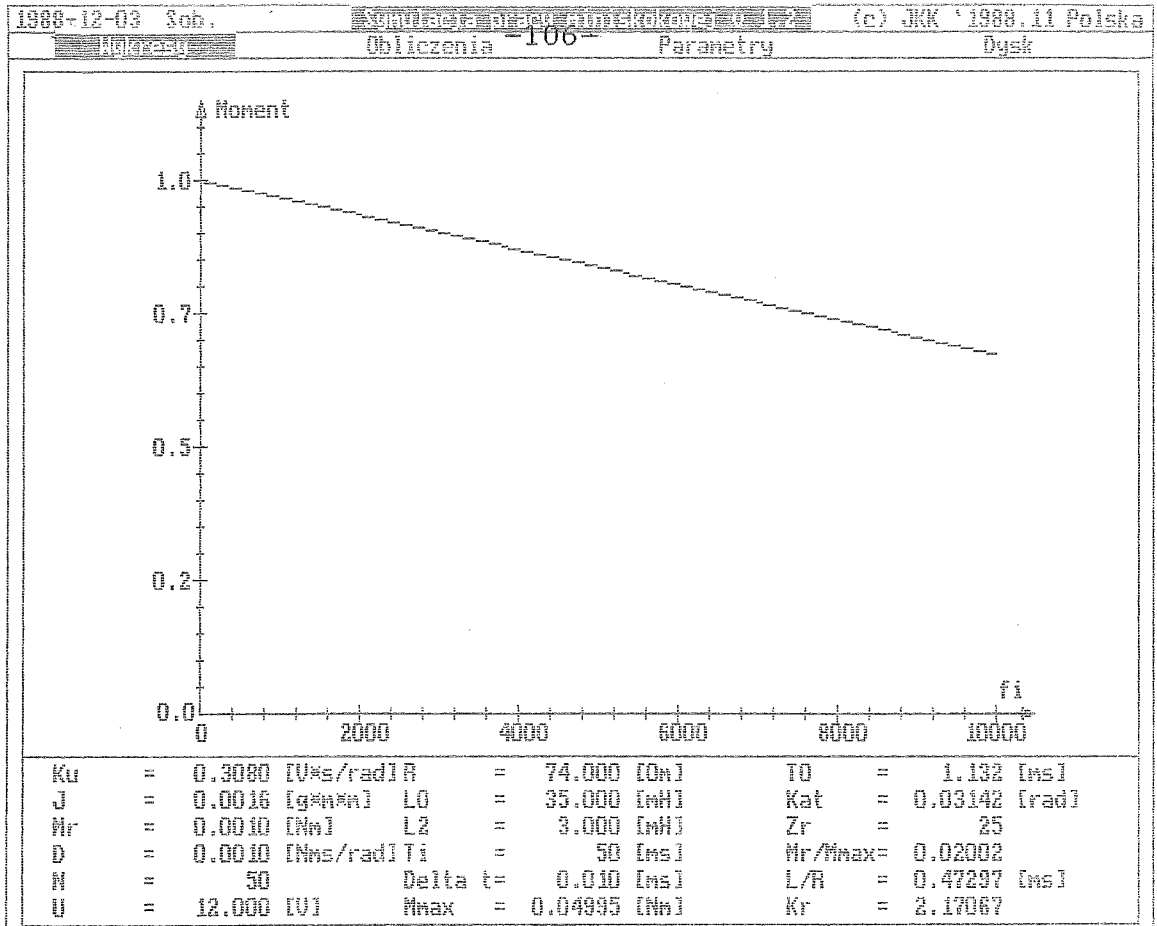
Na kolejnych rysunkach przedstawiono wyniki tej symulacji w postaci charakterystyk mechanicznych dla dwóch różnych przypadków zasilania (napięciowego i prądowego) oraz charakterystyk czasowych. Wyniki symulacji podobnie jak wyniki badań eksperymentalnych przedstawiono w jednostkach względnych.



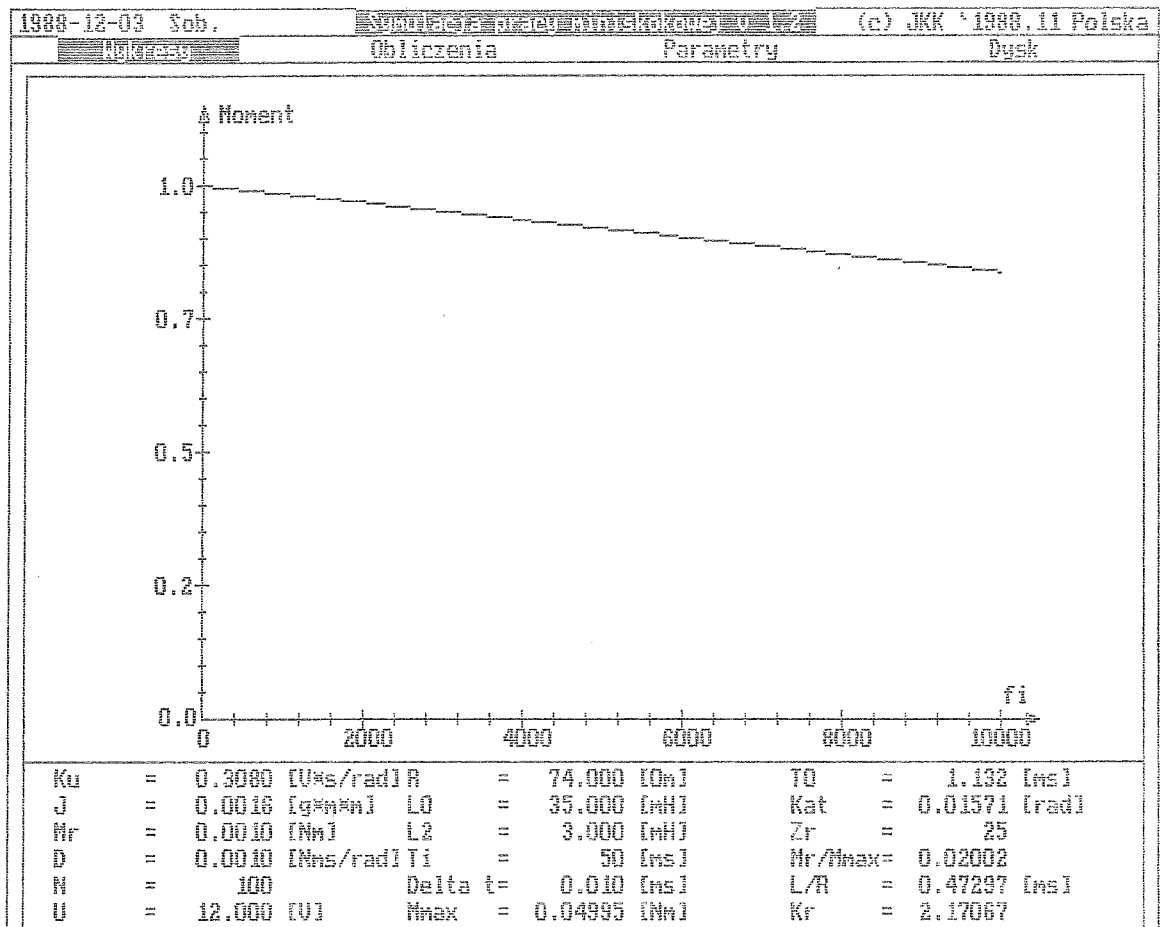
Rys. 6.16.



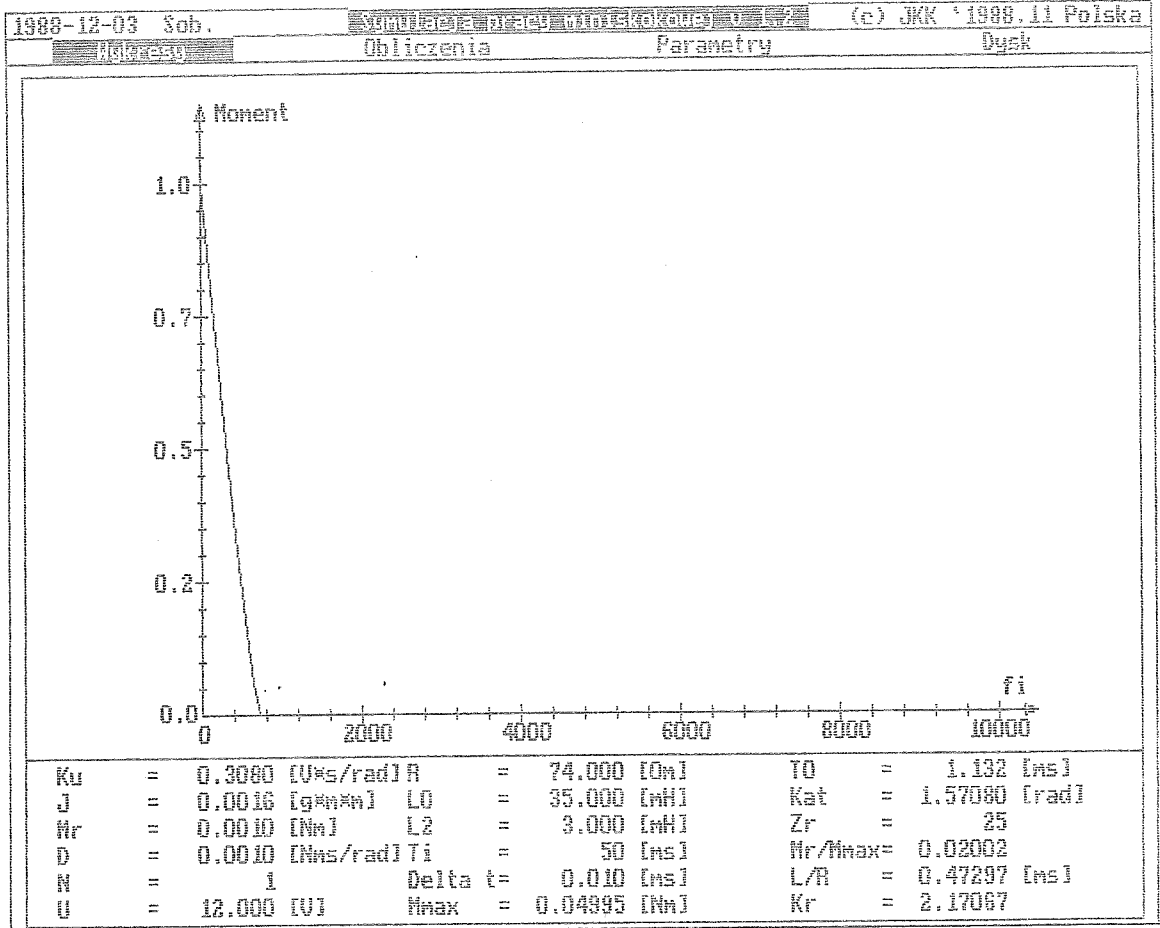
Rys. 6.17.



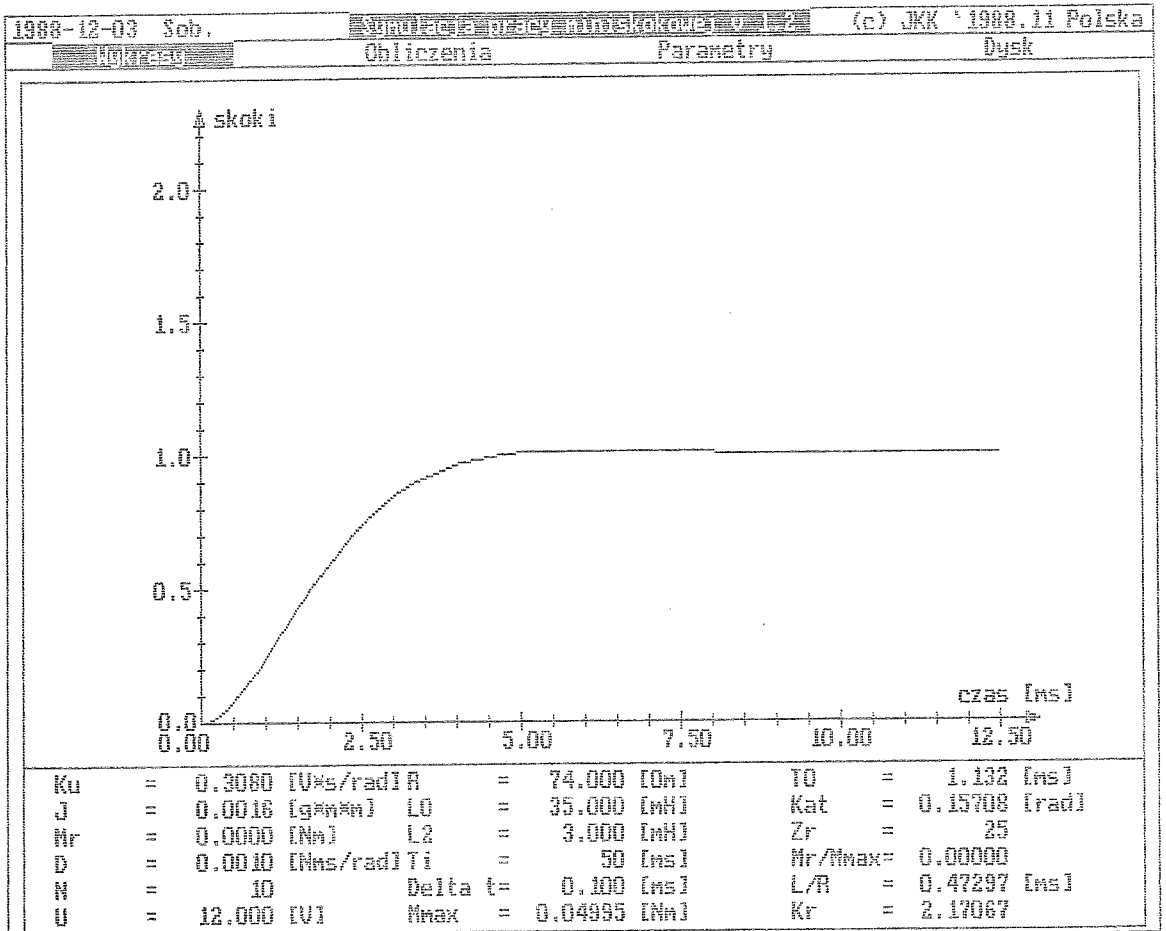
Rys. 6.18.



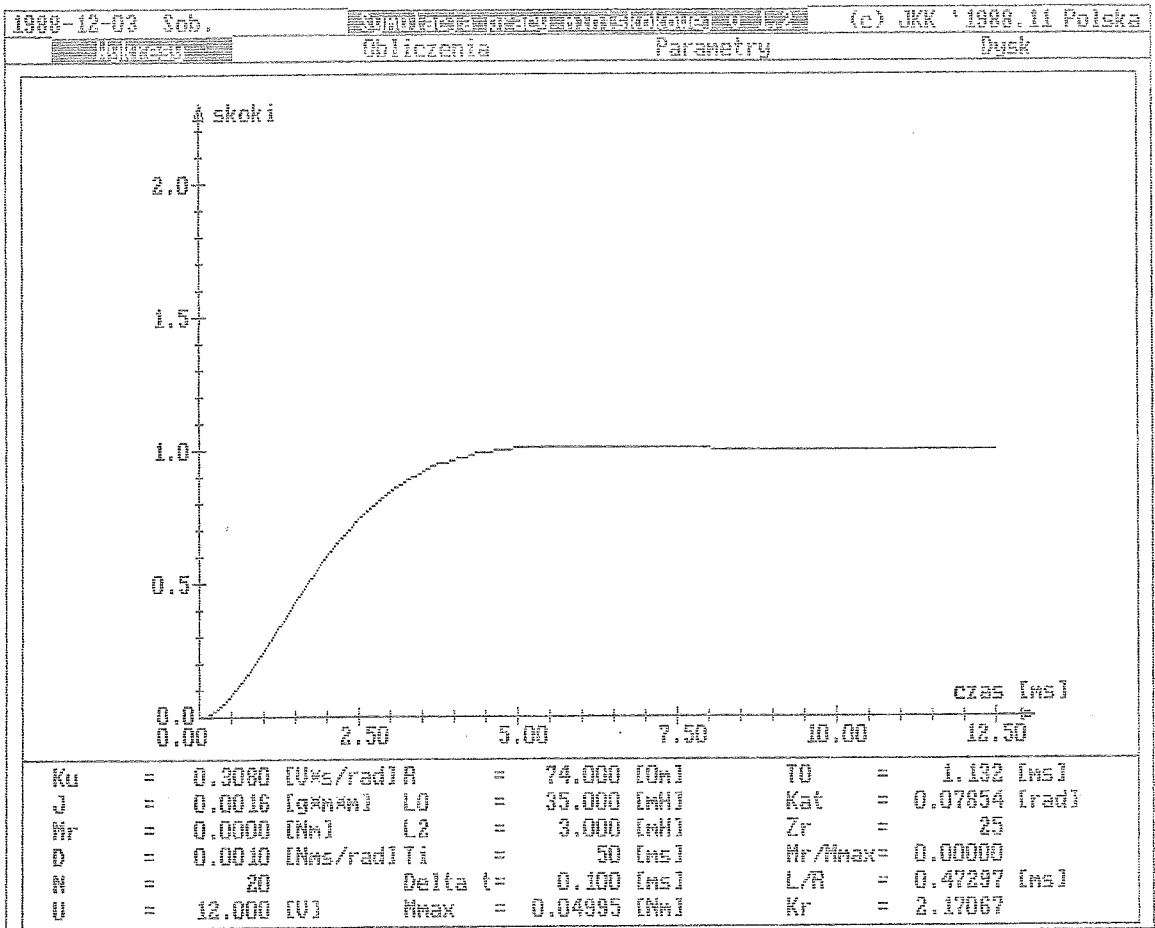
Rys. 6.19.



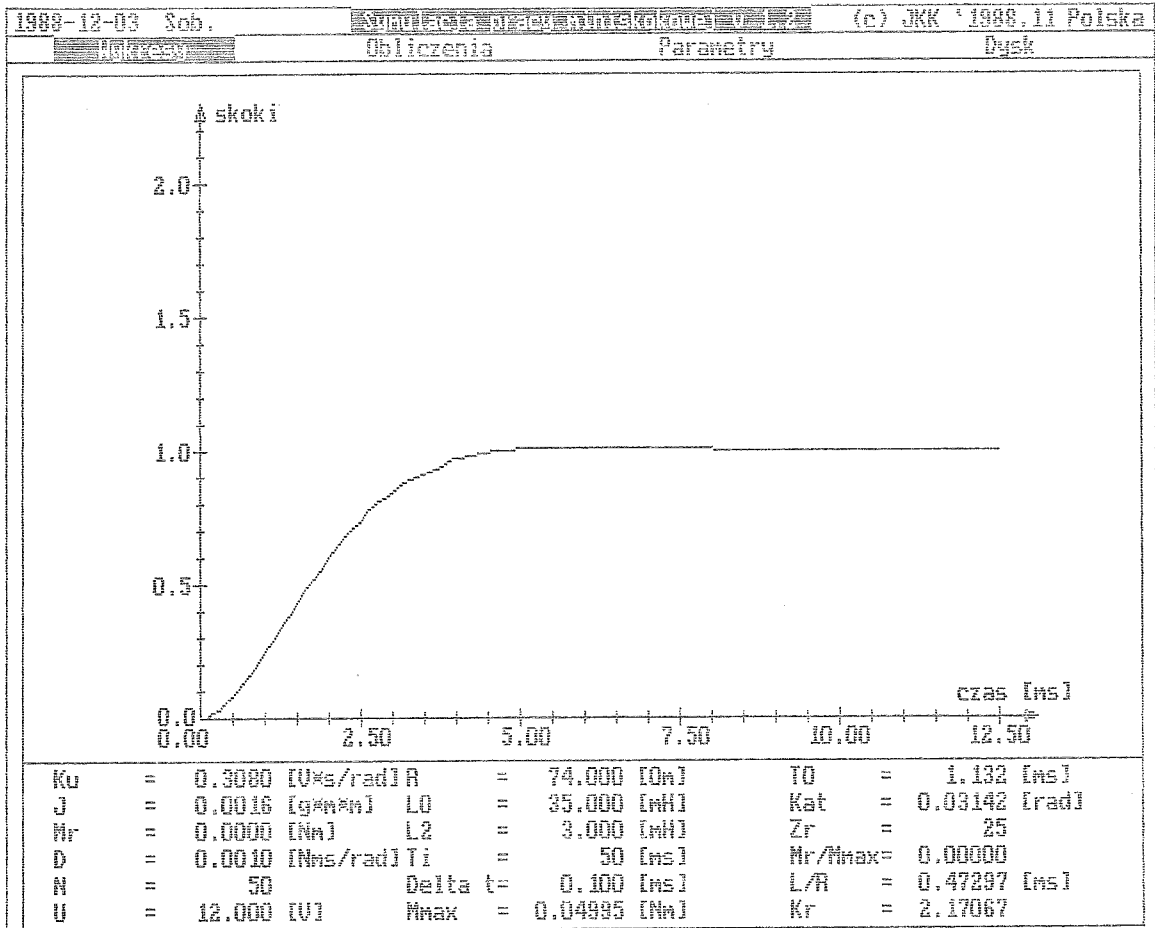
Rys. 6.20.



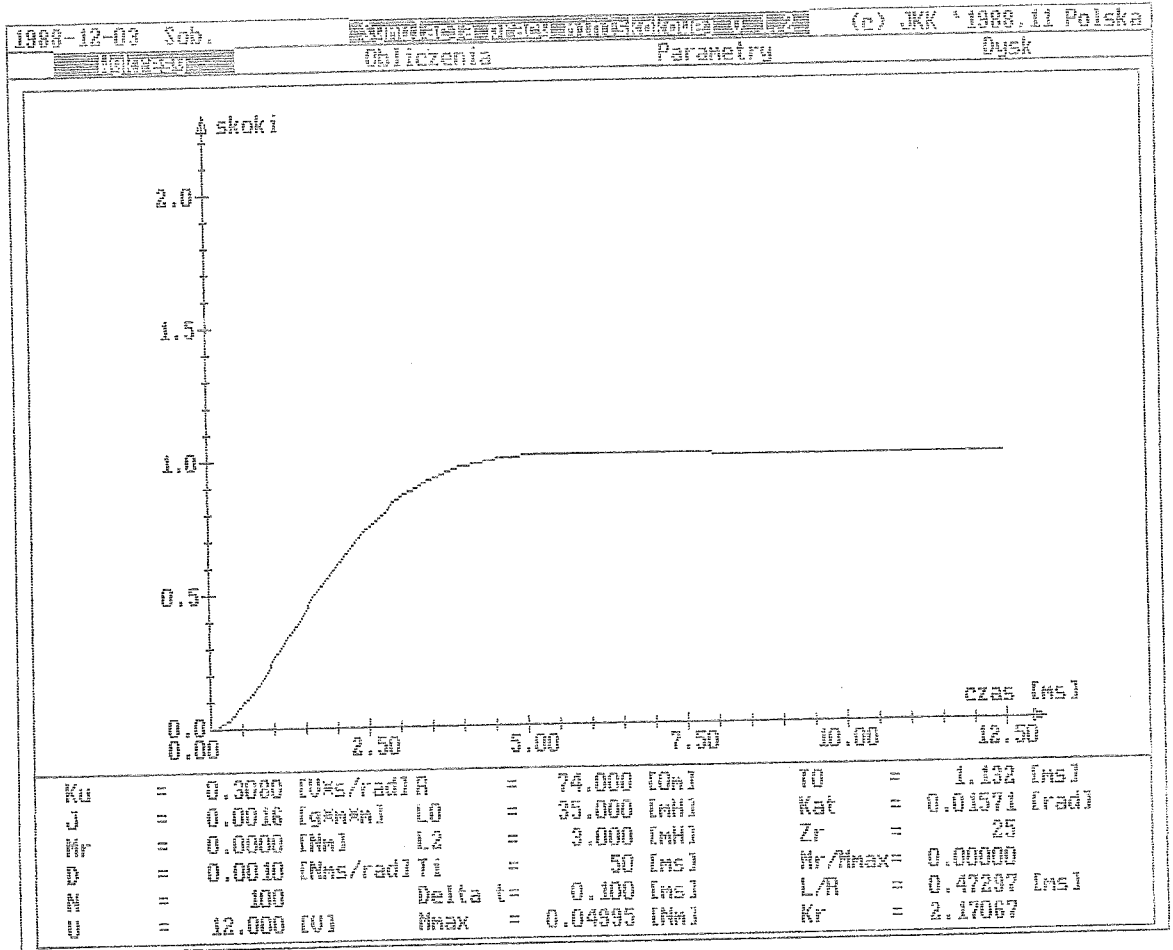
Rys. 6.21.



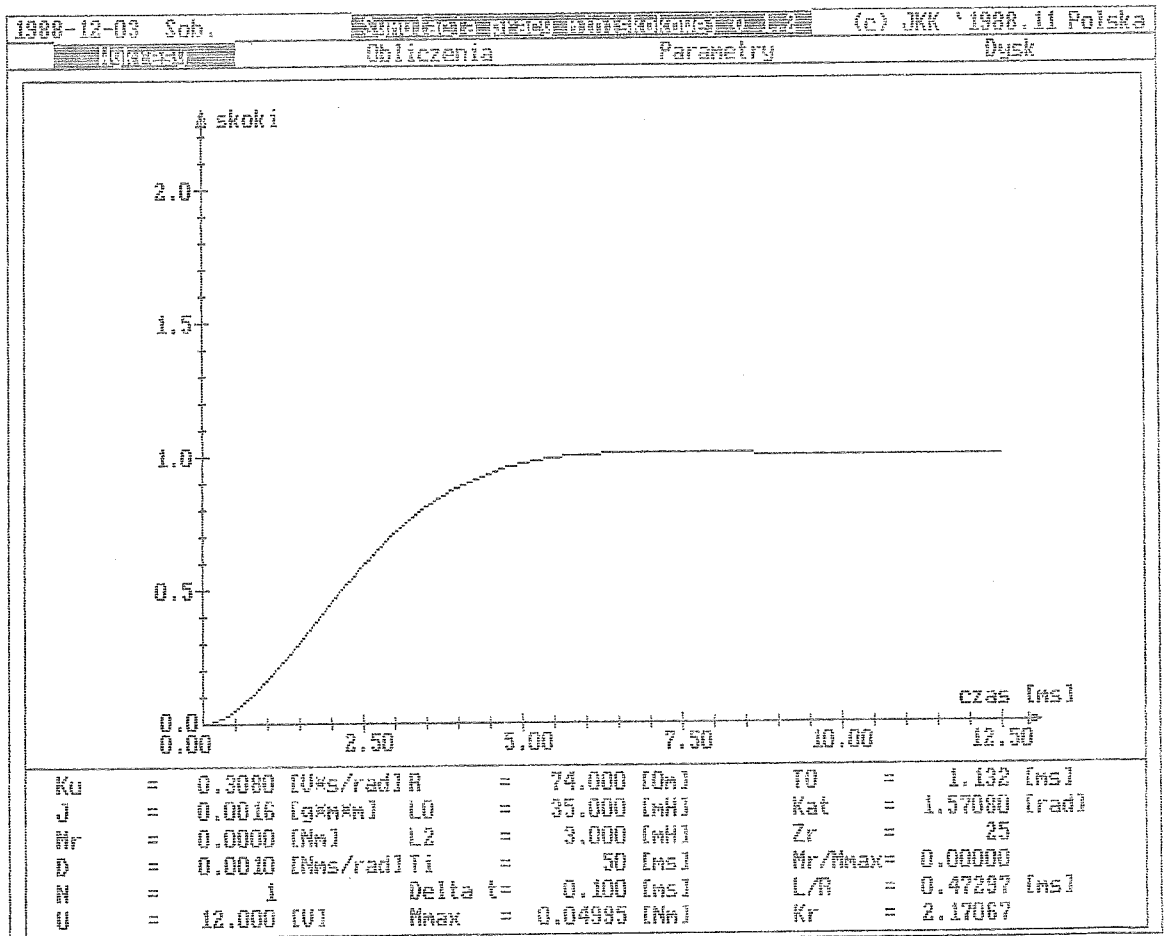
Rys. 6.22.



Rys. 6.23.



Rys. 6.24.



Rys. 6.25.

7. PODSUMOWANIE I WNIOSKI KOŃCOWE

W ramach pracy zrealizowano postawione cele i wymagany zakres pracy. Na podstawie badań eksperymentalnych i symulacji komputerowej określono wpływ sterowania miniskokowego na charakterystyki napędu dyskretnego. Przeprowadzono badania literaturowe w zakresie sterowania i konstrukcji nowoczesnych silników skokowych, ze szczególnym uwzględnieniem pracy miniskokowej (rozdział 2). Zostały przeanalizowane różne metody tworzenia modeli matematycznych napędów dyskretnych z silnikiem skokowym jako elementem wykonawczym (rozdział 3), został przeprowadzony przegląd metod numerycznego modelowania (rozdział 4), a na tej podstawie zbudowano model napędu dyskretnego przy sterowaniu miniskokowym. Model ten stał się podstawą do napisania w języku PASCAL programu symulacji napędu dyskretnego przy pracy miniskokowej, który pozwala obliczać przebiegi czasowe i fazowe oraz charakterystyki mechaniczne modelowanego napędu. (rozdział 4, dodatek A). W rozdziale 5 określono sposób generowania funkcji sterujących sterowania miniskokowego dla napędów z silnikami dwu- i czteropasmowymi przy zasilaniu odpowiednio bipolarnym i unipolarnym. Przeprowadzono badania napędu dyskretnego przy zasilaniu miniskokowym i (dla porównania) przy zasilaniu typu 1-4, w wyniku których uzyskano charakterystyki mechaniczne i czasowe napędu dyskretnego z wybranym silnikiem skokowym typu FA-17-4-1. Przeprowadzono także symulację komputerową takiego napędu przy parametrach silnika FA-17. W czasie symulacji

uzyskano przebiegi czasowe i charakterystyki mechaniczne niezbędne do porównania z badaniami eksperymentalnymi (rozdział 6).

Przeprowadzone badania eksperymentalne i symulacja komputerowa wykazały możliwość znacznego poprawienia parametrów pracy napędu dyskretnego przy pracy miniskokowej. Sterowanie miniskokowe, szczególnie w połączeniu z zasilaniem prądowym, zapewnia elektroniczną redukcję skoku silnika w szerokich granicach, bez znacznego pogorszenia parametrów dynamicznych napędu. Znaczącą cechą napędu miniskokowego (przy zastosowaniu sterownika mikrokomputerowego) jest możliwość zmiany wartości skoku napędu w czasie pracy, wyłącznie w drodze elektronicznego przełączenia. Dodatkową, bardzo korzystną cechą sterowania miniskokowego jest wyeliminowanie zjawisk rezonansu mechanicznego w napędzie, przez co rozszerza się zakres stabilnej pracy napędu i eliminuje występowanie niedozwolonych częstotliwości pracy napędu.

W ramach pracy został skonstruowany cały szereg sterowników miniskokowych przystosowanych do współpracy z silnikami hybrydowymi małej mocy, w których zastosowano zasilanie napięciowe i prądowe. Sterowniki te pozwalają uzyskać do 100 miniskoków w skoku bazowym silnika, co przy zastosowaniu silnika o kącie skoku 3.6° pozwala uzyskać minimalny skok 0.036° (silnik FA-17-4) co daje 10 tys. skoków na obrót silnika. Przy zastosowaniu doskonalszych silników skokowych można uzyskać parametry napędu nie gorsze od reprezentowanego przez większość firm światowych. Napisany w

ramach pracy program symulacyjny pozwala na badanie napędów skokowych (szczególnie w czasie ich projektowania) bez dużych nakładów sił i środków. Wyniki prac zostały już zastosowane w kilku bardzo nowoczesnych napędach skokowych, zrealizowanych m.in. dla potrzeb wojska. Zostały one opisane dokładniej w dodatku D.

Na podstawie pracy można sformułować następujące wnioski końcowe:

- sterowanie miniskokowe pozwala zmniejszyć wartość skoku napędu dyskretnego do ok. 100 razy;
- przy sterowaniu miniskokowym zwiększa się znacznie zakres częstotliwości pracy napędu;
- zastosowanie sterowania miniskokowego eliminuje problem drgań wirnika i rezonanse mechaniczne napędu;
- moment rozwijany przez silnik nie ulega znaczącemu zmniejszeniu przy sterowaniu miniskokowym;
- wymagana dokładność generowania funkcji sterujących (2%) jest osiągnięta przy długości słowa przetwornika C/A 11 bitów.

8. LITERATURA

- [11] Анхимюк В. Л., Панасюк В. И., Комлик Г. П., Политыко Э. Д., Каин Х. А.: Оптимизация по потерям и быстродействию управления шаговым двигателем в процессе позиционирования. Известия Вузов Энерг. nr 3, 1986.
- [12] Athani V. V.: Cadstep computer aided design of variable reluctance stepping motor. Comput. & Electr. vol. 10, nr 1, 1983.
- [13] Bales R.C.: Rotary Pulse Generators Provides all digital Feedback. Control Engineering, August 1967.
- [14] Barriers break down between servo stepper motors. Electr. Rev. nr 15, 1986.
- [15] Baumwolf P.: IC in der Antriebstechnik. Elektrotechnik, nr 20, 1980.
- [16] Bilane G. B.: Step-rate modulation and piecewise linear control to adapt stepping motors to low-bandwidth servos. IEEE Trans. Ind. Electron. nr 1, 1987.
- [17] Bladon J.: Commandes de moteurs pas à pas: Le plus court chemin entre l'électronique et la mécanique. Electron. Ind. nr 115, 1986.
- [18] Boeriu S. G., Lungu M., Graciu Gh.: Adaptive Control System for advancing mechanisms with stepping motor. Proc. 3 Nat. Conf. Elec. Drives, Brasov, May 1982.
- [19] Brown R. H.: New time - optimal control of step motors with a velocity restriction. Powerconvers. and Intell. Motion, nr 3, 1987.

- [10] Чиликин М.Г.: Дискретный электропривод с шаговыми двигателями, Москва, Энергия 1971.
- [11] Carbon C.: La commande électronique des moteurs pas à pas. Bulletin ASE/UC5 73, 5 juin 1982.
- [12] Carlisle B.: Edging into servomotor territory. Mach. Design. nr 26, 1986
- [13] Chua L. O., Pen-Min Lin.: Komputerowa analiza układów elektronicznych. WNT, Warszawa 1981.
- [14] Choosing and using stepper motors. OEM Design, October 1982.
- [15] Commande numérique pour moteurs pas à pas. Electron puissance, nr 14, 1986.
- [16] Costlow T.: Focus on stepper motors. Electronic Design October 31, 1985.
- [17] Crivii M. Trifa V.: Mathematical modelling and simulation of a hybrid - type linear stepping motor. Modell. Simul. Contr. nr 4, 1985.
- [18] Dan T.: Antriebe mit Schrittmotoren - ein Trend zu speichprogrammierten Steuerungen. Messen + Prüfen/Automatik, Dezember 1980.
- [19] Dawson G. E., Eastham A. R., Mira J.: Switched reluctance motor torque characteristics - finite element analysis and test results. IEEE Trans Ind. Appl., nr 3, 1982.
- [20] Delgado M. A.: Mathematical Model of a Stepping Motor Operating as a Fine Positioner Around a Given Step. IEEE Transactions on Automatic Control, August 1969.

- [21] Dietrich L., Spanner K.: Feinpositionierung mit Feinstellschraube und Schrittmotor, Feinwerktechnik & Messtechnik 1982.
- [22] Ellis P.J.: Analysis and control of the permanent magnet stepper motor. The Radio and Electronic Engineer nr 7, 1971.
- [23] Femling D.V.: Use MSI to control stepping motors. Control Engineering, September 1969.
- [24] Fisher R.L., Langley L.: Rare-Earth magnets take motor technology forward. Electrical Review nr 17, May 1979.
- [25] Fulton N.N., Lawrenson P.J., Stephenson J.M., Blake R.J., Davis R.M., Ray W.F.: Recent developments in high-performance switched reluctance drives. 2. Int. Conf. Electr. Mach. - Des. and Appl. London 1985.
- [26] Garner K. C.: Back efm Controls Stepping Rate. Control Engineering, August 1967.
- [27] Geibel G.: Indexer für 5-Phasen-Schrittmotor-Ansteuerung. Antriebstechnik, nr 3, 1987.
- [28] Gleichformig laufende Antriebe kombiniert mit Impulsgebern als Alternative in Schrittmotoren. Maschinenmarkt nr 43, 1983.
- [29] Goddijn B.H.A. : New hybrid stepper motor design. Electronic Components and Applications. Vol. 3 No. 1. November 1980.
- [30] Goddijn B.H.A.: Der Hybrid-Schrittmotor mit Ringspulen. Feinwerktechnik & Messtechnik nr 4, 1981.
- [31] Gollhardt E.: Steuerung von Hybrid - Schrittmotoren. Feingerätetechnik, nr 9, 1986.

- [32] Grabowski J., Kościacz S.: Podstawy i praktyka programowania mikroprocesorów. WNT 1980.
- [33] Gugg H., Sax H.: Schrittmotoren-optimal angesteuert. Steuerungstechnik, nr 26, 1980.
- [34] Halbeck R.: Schachzug wider das Servosystem. Masch. und Werkzeug, nr 5, 1987.
- [35] Hasebe Yoichi .: Manufacturers of Stepping Motors Find Many More Points to Ponder. JEE, July 1985.
- [36] Heine G.: Steuerschaltungen für 5 - Phasen - Hybrid -Schrittmotoren. Feinwerktechnik & Messtechnik nr 4, 1978.
- [37] Hepper E.: "Mikroschritt" besser als "Bürstenlos" ?, Masch. und Werkzeug. nr 11, 1987.
- [38] Iglewski M., Mandey J., Matwin S.: Pascal język wzorcowy, Pascal 360, WNT Warszawa 1986.
- [39] Intel 8080 Microcomputer Systems User's Manual. Intel 1975.
- [40] Iwobotienko i in.: Dyskretne napędy elektryczne z silnikami skokowymi. WNT, Warszawa 1975.
- [41] Ивоботенко В. А., Козим С. С., Лашханри Н. П.: Структура управления шаговым приводом с замыканием по информации с электрических выходов шаговых двигателей. Изв. Вузов Электромеханика, nr 6, 1986.
- [42] Kauffmann J.U., Gudofin E.: Moteur pas à pas a reluctance variable. Revue Generale Electricite, nr 3, 1981.
- [43] Kelemen A., Crivii M., Trifa V.: Numerical Simulation of a Hybrid Type AC 3-phase-stepping Motor

- Incremental Motion Control System. Proc.: 3 Nat. Conf. Elec. Drives, Brasov.
- [44] Keleman A., Trifa V., Crivii M: Phase-locked loop stepping motor positioning control system using microprocessor. Proceedings 3 National Conference Electric Drives. Braşow , May 1982.
- [45] Keleman A., Crivii M., Trifa V.: Mathematical modelling and simulation of stepping motor systems, Math. Modelling nr 8, 1987.
- [46] Keller J.P.: Mise en oeuvre des moteurs pas à pas hybride. Revue Generale Electricite nr 3, 1981.
- [47] Klingman E. E.: Projektowanie systemów mikroprocesorowych. WNT 1982.
- [48] Kenjo T.: Stepping motors and their microprocessor controls. Clarendon Press, Oxford 1984, tłum. na jęz.ros. Энергоатомиздат, Москва 1987.
- [49] Kennedy D.W.: Digital Methods Fixed Phase Lag or Lead. Control Engineering, August 1967.
- [50] Langanke H.: Hochlaufsteuerung eines Schrittmotors mit einem Mikrokomputer. Elektronik nr 9, 1979.
- [51] Lawrenson P.J.: Stepping up to date. Electrics Power, March 1975.
- [52] Ледерер В. В.: Простая схема для управления четырехфазным шаговым двигателем. Приборы и техн. Эксперим. nr 3, 1987.
- [53] Leistungselektronik und Ansteuerreinheiten für 5-Phasen-Schrittmotor. Ind. Ausrüst. Mag. nr 2, 1987.

- [54] Li-Guo-Kun: Magnetostatic energy theory of rare earth cobalt magnetic driving and its applications. Proc 8-th Int. Workshop Rare-Earth Magn. and Appl. and 4-th Int. Symp. Magn. Anisotropy and Coercivity Rare-Earth-Transit. Metal Alloys, Dayton Ohio, May 1985.
- [55] Mailfert A., Charbonnier S., Hambourger G.: Etudy d'un type de moteur hybride utilisable en fonctionnement pas à pas. Revue Generale Electricite nr 3, 1981.
- [56] Marino D.F., Ingle J. D.: Simple and inexpensive microcomputer - controllable stepping motor and drive circuit. Talanta nr 3, 1982.
- [57] Mathams R.F.: Phase plane analysis of a permanent-magnet stepping motor. Electronics Letters nr 6, 1968.
- [58] Microcomputer Data Book. Mostek 1979.
- [59] Möller S.: Antriebe mit Reluktanzschrittmotoren. Feingerätetechnik, nr 5, 1987.
- [60] Niimura Yoshihisa .: The Technologies Used in Making Different Kinds of Stepping Motors. JEE, July 1984.
- [61] O'Donohue J. P.: Transfer Function for a Stepper Motor. Control Engineering, November 1961.
- [62] Ormond T.: Modern stepper-motor systems provide precise position control and resolution. EDN January 8, 1987.
- [63] Owczarek J. i in.: Elektryczne maszynowe elementy automatyki. WNT, Warszawa 1983.

- [64] Page W.D., Singh G., Kuo B.C.: Application of a computer control to a study of open - loop acceleration of step motors. IEEE Transactions on Industrial Electronics and Control Instrumentation, nr 2, May 1975.
- [65] Pal C.: Analogical Modelling of the Stepping Motor Working Equation. Buletinul Institutului Politehnic DIN IASI, Tomul XXVII (XXXI), Fase. 3-4, 1981.
- [66] Paul R.J.A., Wales G.K.: Precise linear incremental displacement using a helical reluctance motor. 2-nd Int. Conf. Small and Spec. Elec. Mach. London, Sept. 1981.
- [67] Pfeffer G.: Anwendung der Theorie der Oberwellendrehfehler auf permanent magnetische Schrittmotoren mit kleinem Schrittwinkel. Archiv für Elektrotechnik nr 58, 1976.
- [68] Pickup I.E.D., Tipping D. .: Method for Predicting the Dynamic Response of a Variable-reluctance Stepping Motor. Proc. IEE, Vol. 32, No. 3, March 1974.
- [69] Pickup J.E.D., Tipping D.: Method for predicting the dynamic response of variable reluctance stepping motor. Proceedings the IEE vol 120, 1973.
- [70] Positionfehler nur $\pm 0.045^\circ$. Elektrotechnik nr 9, 1987.
- [71] Pawletko J. P.: Phase-locked oscillator supplies impuls pulses. Control Engineering, August 1967.
- [72] Praca zbiorowa pod red. S. Boguckiej.: Metody numeryczne w technice. Skrypt Nr 418, Politechnika Śląska, Gliwice 1973.

- [73] Pritchard E.: Curing step motor resonance. Machine Design nr 1, 1980.
- [74] Provaznik F.: Magneticke obvody krokoveho motoru. Elektrotechnicky obzor, nr 10, 1976.
- [75] Pulle D.W.J.: Importance of stepping motor drive matching on the performance. J. Elec. and Electron. Eng. Austral. nr 1, 1986.
- [76] Püschel G., Seibert J.: Schrittmotoren Steuerung an einem CNC-Messprojektor. Feinwerktechnik + Messtechnik nr 5, 1987.
- [77] Puttman R., Seydel E.: Blockorientierte Modellierung eines Schrittmotorantriebes. Feingerätetechnik, Heft 7/1980.
- [78] Ратмиров В.А., Ивоботенко Б.А.: Шаговые двигатели для систем автоматического управления. Москва - Ленинград, ГЭИ, 1962.
- [79] Rauch M.: Optimierung der Ansteuerung von Schrittmotoren. Elektrie nr 8, 1986.
- [80] Renyuan T., Lianfa G., Vichong Ch.: The application of Nd-Fe-B to electrical machines and its comparison with some other permanent magnets. Proc 8-th Int. Workshop Rare-Earth Magn. and Appl. and 4-th Int. Symp. Magn. Anisotropy and Coercivity. Dayton, Ohio, May 1985.
- [81] Renyuan T., Lianfa G., Vichong Ch.: The development and design of the Nd-Fe-B permanent magnet D.C. motor. New Front Rear Earth Sci and Appl. Proc. Int. Conf. Beijing Sept. 1985.
- [82] Richter C.: Positionerantriebe. Feingerätetechnik, nr

- 2, 1982.
- [83] Richter L.: Schrittmotoren - Technologie Ind. Ausrüst. Mag. nr 6, 1986.
- [84] Robinson D.J., Taft C.K. .: A Dynamic Analysis of Magnetic Stepping Motors. Transactions on Industrial Electronics and Control Instrumentation, Vol. IECI-16, No. 2, September 1969.
- [85] Reglič J., Sobek P.: Zapojeni pro přesně kroku krokových motoru. Patent CSRS nr 234177, 1987 r.
- [86] Rösler G., Paschen F.: Schaltregel mit VMOS-Transistoren für Schrittmotoren, Elektronik, nr 7,1979.
- [87] Rubet R.: Les moteurs pas à pas à aimants permanents. Revue Generale Electricite nr 3, 1981.
- [88] Ruschmeyer K.: Elektromotoren für schnelle und präzise Antriebe. Feinwerktechnik & Messtechnik nr 4, 1982.
- [89] Saunier Ch.: Les moteurs pas à pas et leurs techniques de Commandes. EMI 172/1-5-1973.
- [90] Sax H.: Schrittmotoransteuerung mit monolitischen Bausteinen. Elektronik, Heft 23, 1980.
- [91] Schrittmotoren - optimal angesteuert. Elektrotechnik nr 7,1982.
- [92] Schwager B.: Operation and drive of stepper motors. Siemens components. nr 4, 1986.
- [93] Schrittmotoren in Mikroschritt. Antriebstechnik, nr 4, 1986.
- [94] Schwager B.: Ein IC als Verbindungsglied zwischen Mikropcessor und Schrittmotor. Bull. Schweiz.

Elektrotechn. Ver. nr 19, 1987.

- [95] Сиротин А. А., Киреев В. В.: Унифицированные транзисторные разпределители импульсов для управления шаговыми электрическими двигателями. Электричество 2/1965.
- [96] Siefert W., Gfrörer R.: Einsatz und Grenzen des Mikroschritt Betriebes. Ind. Rept. Int. nr 5, 1986.
- [97] Sochocki R.: Sposoby sterowania silników skokowych. Prace Instytutu Cybernetyki Stosowanej, nr.7, 1972.
- [98] Spital I.: Amstrad CPC 6128 User Instruction. Amsoft 1985.
- [99] Stevens H. G.: Selecting a stepping motor. OEM Design, July 1982.
- [100] Strnat K.J.: The hard magnetic properties of rare earth-transition metal alloys. Intermag Conference 1972.
- [101] Szmelter J. .: Metody komputerowe w mechanice. PWN, Warszawa 1980.
- [102] Takahashi H. , Tomoo T.: How Microprocessors Are Used In Motor Control Circuits. JEE, July 1984.
- [103] Teodorescu D.: Schrittmotoren oder asynchrone Stellmotoren. Elektr. Masch. H.2, 1977.
- [104] Tietze E.: Steuerung und Stellglieder zur optimalen Ausnutzung von Schrittmotoren. Elektronik, nr 8, 1975.
- [105] Tisco Spitzenreiter Katalog. Texas Instruments 1979.
- [106] Toacșe G., Romanca M., Pana G.: Step transducer for PM hybrid stepping motor. Electr. Mach and power syst. nr 4, 1986.

- [107] Toacșe G.: Micropășirea - regimul de comandă optim al motorului pas cu pas in realizarea innergistratvarelor XY. Electrotehn., Electron și Automat. Elektrotehn. nr 6, 1981.
- [108] Toacșe G.: Translator design for PM hybrid stepping motor. Proc. 3 Nat. Conf. Electr. Drives, Brasov, May 1982.
- [109] Toacșe G.: Translator design of the stepping motor. IEEE Transactions on industrial electronics and control instrumentation, May 1975.
- [110] Troussel G.: Commande integré Bimos de deux moteurs pas à pas unipolaires. Electron. puissance nr 20, 1987.
- [111] Trummer H.: Störspannungen abgemeldet Maschinenmarkt., nr 9, 1987.
- [112] Trummer H.: Einsatz von Schrittmotoren und Positionierungsteuerungen. Konstr. und Electron. März 1987.
- [113] Vikram M.: Novel four phase stepper motor control. Electronic Engineering June 1979.
- [114] Waldö S.: Intelligenter Treiber für Schrittmotoren. Elektronik, nr 21, 1980.
- [115] Wockel C., Hamman E., Balas M., Lang J.: Comparison of linearized versus nonlinear control of D.C. stepping motors. Contr. and Comput. nr 1, 1986.
- [116] Whitaker R., Paul R.J.A, Jones D.J.: Microprocessor control of a step motor with various inertia loads. 2-nd Int. Conf. Small and Spec. Elec. Mach. London, Sept.1981.

- [117] Wockel C., Hamman E., Balas M.: Comparison of linearized versus nonlinear control of D.C. stepping motors. Mini and Microcomput. Contr. Filter and Signal Process Proc. ISMM Int. Symp. Las Vegas December 1984.
- [118] Worsley R.A.: Distributed processing applied to stepper motor control. Electronics & Power, March 1982.
- [119] Wróbel T.: Uwagi o współczesnych układach sterowania silnikami skokowymi. Wiad. Elektrotech. nr 13-14, 1985.
- [120] Wróbel T.: Sterowanie silnikami skokowymi. WPT, nr 8/9, 1985.
- [121] Wróbel T.: Mniej znane zastosowania silników skokowych. WPT, nr 11, 1982.
- [122] Wróbel T.: Magnesy z domieszką metali ziem rzadkich w maszynach elektrycznych. WPT, nr 5, 1982.
- [123] Wróbel T.: Próba klasyfikacji silników skokowych. Wiad. Elektrot. nr 17-18, 1985.
- [124] Wróbel T.: Silnik skokowy w układzie otwartym i zamkniętym w porównaniu z innymi rodzajami napędów. Wiad. Elektr., nr 19-20, 1985.
- [125] Yell R., White B., Holland K.: Improving stepping motor angular resolution. Electronic Engineering, Part I, March 1978, Part II, April 1978.
- [126] Zylka W.: Dosierpumpenantrieb mit Schrittmotor. Feinwerktechnik & Messtechnik nr 3, 1983.

9. LITERATURA PATENTOWA

- [P-1] Bechtle B. i in., Method of controlling a stepper motor. US Patent 4072888.
- [P-2] Pritchard E. K., Digital stepping motor device circuit. US Patent 4087732.
- [P-3] Maeda Y. Matsumoto K., Control Apparatus for driving a pulse motor. US Patent 4145643.
- [P-4] Langley L., Kidd K., Incremental motion motor controller. US Patent 4234838.
- [P-5] Meier W., Stepping motor control circuit. US Patent 4253052.
- [P-6] Chiang D., Method of and apparatus for operating stepping motors. US Patent 4280084.
- [P-7] Marinko J. A., Motor control. US Patent 4336484.
- [P-8] Araki S. i in., Steping motor drive apparatus. US Patent 4339702.
- [P-9] Cannon L. E., Stepping motor control circuit. US Patent 4362979.
- [P-10] Buxton K., Stepper motor control circuit. US Patent 4349771.
- [P-11] Gessner G. A., Stepping motor drive circuit. US Patent 4414498.
- [P-12] Larson N. G. et al., Stepping motor control system. US Patent 4426608.
- [P-13] Kaseta R. G., Stepping motor excitation circuitry. US

Patent 4438380.

- [P-14] Yardley et al., Control circuit for numerically controlled motor. US Patent 4570112.
- [P-15] Janes R. S., Control for a stepper motor or other synchronous motor. European Patent 0 202 043.
- [P-16] Janes R. S., Control for a stepper motor or other synchronous motor. US Patent 4700120.
- [P-17] Kanemura S., Motor controlling system. US Patent 4707649.
- [P-18] Turvey S., Stepper motor control circuit. European Patent 0099918.
- [P-19] Matsushita T., Method of driving a stepping motor and a drive circuit thereof. European Patent 0116677.
- [P-20] Kumpfmüller H-G., Verfahren und Einrichtung zur Steuerung eines Schrittmotors. Europäische Patent 0143241.
- [P-21] Henle H., Langfeld F., Schaltungsanordnung zur Steuerung von Schrittmotoren. Europäische Patent 0144843.
- [P-22] Sato M. et al., Stepping motor control system. European Patent 0200959.
- [P-23] Dölves J., Verfahren und Schaltungsanordnung zur Ansteuerung eines Schrittmotors. Europäische Patent 0228475.
- [P-24] Бондаренко Ю. Г., Устройство для программного управления шаговым двигателем. SU 1200248 A.

- [P-25] Шильников С. В., Дынец В. С., Устройство для программного управления шаговым двигателем. SU 1267363 A1.
- [P-26] Ледерер В.В., Устройство для управления четырехфазным шаговым двигателем. SU 1348775 A1.
- [P-27] Majewski M., Stepper motor logic circuit. US Patent 4613806.
- [P-28] Szwoch W. i in. Układ sterowania silnikiem skokowym. Patent PRL 104233.
- [P-30] Gorzelak J. i in., Sterownik do silników krokowych czterofazowych. Patent PRL 139518.
- [P-31] Rosochowicz L., Układ cyfrowego sterowania i kontroli ruchem silnika krokowego, zwłaszcza w goniometrach. Patent PRL 132010.
- [P-31] Newell H. R., Apparatus for dividing the step angle of a stepping motor. US Patent 4297625.
- [P-32] Pritchard E.K., Microstepping drives with waveform tuning and microprocessor control. US Patent 4584512.
- [P-33] Maeno S. et al., System for driving drum. European patent 0230134.
- [P-34] Giguere I.J., Digital motor control method and means. European Patent 0145198.

ZALĄCZNIK A.

PROGRAM ANALIZY PRACY SILNIKA SKOKOWEGO PRZY STEROWANIU MINISKOKOWYM

Program symulacji komputerowej silnika skokowego przy pracy miniskokowej został napisany w języku Pascal i skompilowany przy użyciu kompilatora Turbo Pascal v 4.0.

Program Miniskok;

```
uses Graph,Crt,Printer,Dos;
```

```
const wym=2;
```

```
    D1Opcji=14;
```

```
    Odst=30;
```

```
    tytul=' Symulacja pracy miniskokowej v 1.2 ';
```

```
    fmax=2000;
```

```
{ $N+ }
```

```
type nazwa=string[70];
```

```
    MenuText=array [1..25] of nazwa;
```

```
    Wynik=array [0..500] of record
```

```
        czas, kat_fi, predk, blad:Real;
```

```
    end;
```

```
    Par = record
```

```
{ ===== parametry zadawane ===== }
```

```
    Ku,          ( wspolczynnik napieciowy [V*s/rad] )
```

```
    Mmax:Real;   ( moment obciazenia [Nm] )
```

```
    J,          ( calk. moment bezwladnosci [kg*m*m] )
```

```
    Mr,         ( calkowity moment obciazenia [Nm] )
```

```
    D,         ( wsp. tarcia lepkiego [N*m*s/rad] )
```

```
    R,         ( rezystancja pasma [Om] )
```

```
    LO,        ( indukcyjnosc srednia pasma [H] )
```

```
L2:Real;      { amplituda indukcyjnosci pasma [H] }
Zr:Integer;   { liczba zebow wirnika }
K:Byte;       { liczba miniskokow w skoku bazowym }
U,            { napiecie zasilania [V] }
Delta_t:Real; { odstep czasu miedzy krokami calkowania [s] }
Okres:Real;   { okres impulsow sterujacych [1/s] }
IIPktObl:Integer; {Liczba punktow calkowania}
IlWykr:Byte;  {Liczba ekranow wyswietlania IIPktObl/125 }
Fmax:Real;    { maksymalna czestotliwosc badan }

( =====
===== parametry obliczane =====)

Omega_0,      { czestosc drgan wlasnych [Hz] }
Ksi_0,        { wzgl. stala czasowa od L0 }
Ksi_2,        { wzgl. stala czasowa od L2 }
Beta_e,       { wzgl. wsp. tarcia lepkiego elektromag. }
Beta_m,       { wzgl. wsp. tarcia lepkiego mech. }
Beta,         { calk. wsp tarcia lepkiego }
Mi_r:Real;    { wzgl. moment obciazenia Mr/Mmax}
Kat:Real;     { kat miniskoku = pi/2/K }
LR:Real;      { stala czasowa obwodu pasma }
Kr:Real;      { wspolczynnik momentu }

                end; {of Parametr}

var GrafType,GrafMode:Integer; {zmienne okreslajace rodzaj drajwera
                                graficznego i jego mod pracy}

MaxX,MaxY:Word;                {zmienne okreslajace maksymalne
                                wymiary ekranu w punktach}

Wykres:wynik;                  {wektor z charakterystykami obliczonymi }
Parametr:Par;                  {parametry obliczen }
ZbDane:text;                    {zbior z wynikami obliczen}
M:Byte;                         {okresla wybrana funkcje}
NazZbDane:nazwa;               {nazwa pliku z danymi z obliczen}

{ zmienne parametrow wejsciowych }
{ zmienne globalne procedur dyskowych }
```

```
NazParLad:nazwa;      {nazwa zbioru parametrow do ladowania}
NazParZap:nazwa;      {nazwa zbioru parametrow do zapisu}
MaskaKat :nazwa;      {maska katalogu zbiorow}
  Katalog:nazwa;      {aktualny katalog dyskowy}
ZbWynOtw:Boolean;
CharMom: array [0..500] of record
  f,M:Real;
end; {zbior wyników obliczeń charakterystyki
      momentu w funkcji częstotliwości}
```

{=====}

Const

```
MenuPar:MenuText=
(*F1 -Wspolczynnik napieciowy : ? ,
 *F2 -Moment synchronizujacy : ? ,
 *F3 -Calkowity moment bezwl. : ? ,
 *F4 -Calkowity moment obc. : ? ,
 *F5 -Wsp. tarcia lepkiego : ? ,
 *F6 -Rezystancja pasma : ? ,
 *F7 -Indukcyjnosc srednia : ? ,
 *F8 -Amplituda indukcyjnosci : ? ,
 *F9 -Liczba zebow wirnika : ? ,
 *F10-Liczba miniskokow : ? ,
 *F11-Napiecie zasilania : ? ,
 *F12-Okres probkowania : ? ,
 *F13-Okres impulsow ster. : ? ,
 *F14-Liczba punktow obliczen : ? ,
 *F15-Max. czest. badan : ? ,
 * Kat miniskoku : ? ,
 * Okres drgan wlasnych T0 : ? ,
 * Wzgl. stala czasowa 0 : ? ,
 * Wzgl. stala czasowa 2 : ? ,
 * Wzgl. wsp. tarcia lep. em: ? ,
 * Wzgl. wsp. tarcia lep. me: ? ,
 * Mr/Mmax : ? ,
```

```
* Tr/TO : '?',
* L/R : '?',
* Kr : '?';
```

PytPar:array [1..15] of nazwa=

```
('Wspolczynnik napieciowy silnika w [V*s/rad] : '?',
'Maksymalny moment synchronizujacy [mNm] : '?',
'Calkowity moment bezwladnosci silnika i obciazenia w [kg*m*m] : '?',
'Calkowity moment obciazenia w [Nm] : '?',
'Wspolczynnik tarcia lepkiego : '?',
'Rezystancja pasma w [Om] : '?',
'Indukcyjnosc L0 w [mH] : '?',
'Indukcyjnosc L2 w [mH] : '?',
'Liczba zebow wirnika : '?',
'Liczba miniskokow w skoku bazowym : '?',
'Napiecie zasilania w [V] : '?',
'Okres probkowania [ms] : '?',
'Okres impulsow sterujacych [ms] : '?',
'Liczba punktow obliczen (1=125, 2=250, 3=375, 4=500) : '?',
'Maksymalna czestotliwosc pracy w [Hz] : '?' );
```

JednPar:MenuText=

```
(' [V*s/rad]',
' [mNm]',
' [kg*m^2*E-3]',
' [mNm]',
'',
' [Om]',
' [mH]',
' [mH]',
'',
' [V]',
' [ms]',
' [ms]',
'');
```

```
' [Hz]',  
' [rad]',  
' [ms]',  
'',  
'',  
'', ''', ''', ''', ''', '' [ms]', ''');
```

{ \$I untility1.jkk }

```
Procedure ObliczParametry;  
begin  
With parametr do  
begin  
Omega_0:=Sqrt((Ku*Zr*U)/(R*J));  
Ksi_0:=Omega_0*L0/R;  
Ksi_2:=Omega_0*L2/R;  
Beta_e:=Omega_0*Ku/(2*Zr*U);  
Beta_m:=D/(Omega_0*2*J);  
Beta:=Beta_e+Beta_m;  
Mmax:=Ku*U/R;  
Mi_r:=mr/Mmax;  
kat:=pi/2/K;  
LR:=L0/R;  
Kr:=Ku*R/Zr/L0/U;  
end;  
end; { of ObliczParametry }
```

```
Procedure TestZap;  
var i,j:Word;  
N:Integer;  
c:Real;  
begin
```

```
for i:=0 to Parametr.ilPktObl do
begin
  Wykres[i].kat_fi:=Parametr.kat+Parametr.kat*sin(i/10)*Exp(-i/60);
  Wykres[i].czas:=i/25;
  CharMom[i].f:=50*i;
  CharMom[i].M:=i/50;
end;
end; {of zapis kontrolny}
```

```
{SI ramka.jkk}
{SI parametr.jkk}
{SI dysk.jkk}
{SI oblicz.jkk}
{SI wysw.jkk}
```

```
{Program Glowny}
```

```
var
```

```
  f:file;
  p,q:Pointer;
  s,s1:Word;
  r:Word;
  a:Real;
```

```
Begin
```

```
  GrafType:=7;
```

```
  GrafMode:=0;
```

```
  {wartosci poczatkowe dla FA-17-4-1}
```

```
  NazParLad:='PARAM.PAR';
```

```
  NazParZap:='PARAM.PAR';
```

```
  MaskaKat:='*.*';
```

```
GetDir (0, Katalog);
NazZbDane:='WYNIKI.DTA';
Assign (ZbDane, NazZbDane);
ZbWynOtw:=False;
With Parametr do
begin
  Ku:=0.308;
  Mmax:=0.05;
  J:=1.6E-6;
  Mr:=1E-3;
  D:=0.001;
  Okres:=50E-3;
  K:=1;
  U:=12.0;
  R:=74.0;
  L0:=0.035;
  L2:=0.003;
  Zr:=25;
  Delta_t:=1E-5;
  IIPktObl:=125;
  I1Wykr:=1;
  Fmax:=2000;
end;
ObliczParametry;
TestZap;
{ koniec zadawania wartosci poczatkowych }

InitGraph (GrafType, GrafMode, 'c:\turbo4\drivers');
MaxX:=GetMaxX;
MaxY:=GetMaxY;

Ekran;
M:=4;
repeat
```



```
MenuG1;  
  Case M of  
    1:Wyswietlanie;  
    2:Oblicz;  
    3:Param;  
    4:Dysk;  
  end;  
if m=0 then  
  Begin  
    S1:=Okno(100,250,600,300,q);  
    Napis(100,271,500,'Jestes pewien (T/N) ?',False);  
    Napis(100,269,500,'      '+''''+      ',False);  
    if UpCase(ReadKey)<>'T' then m:=4;  
    KasOkno(100,250,q,s1);  
  end;  
until m=0;  
CloseGraph;  
if ZbWynOtw then Close(ZbDane);  
end.
```

```
Procedure ReTypeInv(IloscZnakow:Byte);
```

```
var
```

```
  Size      :Word;
```

```
  xp,yp,xk,yk:Integer;
```

```
  p          :Pointer;
```

```
begin
```

```
  xp:=GetX;
```

```
  yp:=GetY-2;
```

```
  xk:=xp+8*IloscZnakow;
```

```
  yk:=yp+9;
```

```
  Size:=ImageSize(xp,yp,xk,yk);
```

```
  GetMem(p,Size);
```

```
  GetImage(xp,yp,xk,yk,p^);
```

```
    PutImage(xp, yp, p^, 4);  
    FreeMem(p, Size);  
    MoveTo(xp, yp+1);  
end; {of ReTypeInv}
```

```
Procedure UsunText(IloscZnakow:Byte);
```

```
var
```

```
    Size      :Word;  
    xp, yp, xk, yk: Integer;  
    p        :Pointer;
```

```
begin
```

```
    xp:=GetX;  
    yp:=GetY;  
    xk:=xp+8*IloscZnakow;  
    yk:=yp+8;  
    Size:=ImageSize(xp, yp, xk, yk);  
    GetMem(p, Size);  
    GetImage(xp, yp, xk, yk, p^);  
    PutImage(xp, yp, p^, XORPut);  
    FreeMem(p, Size);  
    MoveTo(xp, yp);
```

```
End; {of UsunText}
```

```
Procedure ClearImage(xp, yp, xk, yk: Integer);
```

```
var
```

```
    Size      :Word;  
    x, y      : Integer;  
    p        :Pointer;
```

```
begin
```

```
    x:=GetX;  
    y:=GetY;  
    Size:=ImageSize(xp, yp, xk, yk);  
    GetMem(p, Size);  
    GetImage(xp, yp, xk, yk, p^);
```

```
    PutImage(xp, yp, p^, XORPut);
    FreeMem(p, Size);
    MoveTo(xp, yp);
End; {of ClearImage}
```

```
Procedure Napis(x, y, dlug: Integer; lwe: nazwa; inv: Boolean);
var
    z: Integer;
begin
    z := Round((Dlug - TextWidth(lwe)) / 2);
    OutTextXY(x + z, y, lwe);
    if inv then begin MoveTo(x + z, y); ReTypeInv(Length(lwe)) end;
end; {Napis}
```

```
Procedure Ustaw(xp, yp, x, y: Integer);
Begin
MoveTo(xp, yp);
MoveRel(x, y);
end;
```

```
Function Okno(x1, y1, x2, y2: Integer; var p: Pointer): Word;
var Size: Word;
Begin
    Size := ImageSize(x1, y1, x2, y2);
    GetMem(p, Size);
    GetImage(x1, y1, x2, y2, p^);
    PutImage(x1, y1, p^, XORPut);
    Rectangle(x1, y1, x2, y2);
    Okno := Size;
end; {of Okno}
```

```
Procedure KasOkno(x, y: Integer; var q: Pointer; Size: Word);
Begin
    PutImage(x, y, q^, NormalPut);
    FreeMem(q, Size);
```

```
end; {of KasOkno}
```

```
Function StrVal(a:Real; dl,u:Byte):nazwa;  
var x:Nazwa;  
begin  
if u>dl then Str(a:dl,x) else Str(a:dl:u,x);  
StrVal:=x;  
end; {of StrVal}
```

```
Procedure PrintMenu(x,y:Integer; wys,szer:Byte; Menu:MenuText);  
var i,j:Integer;  
Begin  
ClearImage(x+1,y+1,x+szer*B-1,y+wys*10-1);  
for i:=1 to wys do  
Begin  
OutTextXY(x+2,y+3+(i-1)*10,Menu[i]);  
end;  
end; {of PrintMenu}
```

```
Procedure MenuGl;  
var i:Byte;  
z:Char;  
ip:Byte;  
Begin  
i:=M;  
ip:=Round(MaxX/4);  
MoveTo(Odst+(i-1)*ip,12);  
ReTypeInv(d1Opcji);  
repeat  
z:=UpCase(ReadKey);  
if z=#0 then z:=UpCase(ReadKey);  
MoveTo(Odst+(i-1)*ip,12);  
ReTypeInv(d1Opcji);
```

```
Case z of
#75:i:=i-1;
#77:i:=i+1;
#27:begin
    M:=0;
    Exit
End;
'W':begin i:=1; z:=#13; end;
'O':begin i:=2; z:=#13; end;
'P':begin i:=3; z:=#13; end;
'D':begin i:=4; z:=#13; end;
end;
if i<1 then i:=1;
if i>4 then i:=4;
MoveTo(Odst+(i-1)*ip,12);ReTypeInv(D1Opcji);
until z=#13;
M:=i;
MoveTo(Odst+(i-1)*ip,12);ReTypeInv(D1Opcji);
end; {of MenuG1}
```

```
Function Menu(x,y:Integer; dl:Byte; max,nr:Byte):Byte;
var i:Byte;
    z:Char;
Begin
i:=nr;
repeat
    MoveTo(x+2,Y+2+(i-1)*10);
    ReTypeInv(dl);
    z:=ReadKey;
    MoveTo(x,y);
    MoveRel(2,2+(i-1)*10);
    ReTypeInv(dl);
case z of
#72:if i>1 then i:=i-1;
```

```
#60:if i<MAX then i:=i+1;
#71:i:=1;
#59..#68: begin i:=Ord(z)-58; if i>MAX then i:=max; end;
#84..#93: begin i:=Ord(z)-73; if i>MAX then i:=max; end;
#27:begin Menu:=0; Exit end;
end;
until z=#13;
MoveTo(x+2,Y+2+(i-1)*10);
Menu:=i;
End; {of Menu}
```

```
Function PodajLiczbe(x,y:Integer; var dl:Byte):Real;
```

```
var znak:Char;
```

```
l :Real;
```

```
rep :Integer;
```

```
liczba:Nazwa;
```

```
Begin
```

```
liczba:='';
```

```
OutTextXY(x,y,'_');
```

```
repeat
```

```
repeat
```

```
znak:=UpCase(ReadKey);
```

```
if znak=#8 then Delete(liczba,Length(liczba),1);
```

```
if znak=#27 then begin dl:=0; Exit end;
```

```
if znak in ['0'..'9','.',',','-','E'] then liczba:=liczba+znak;
```

```
if Length(liczba)>dl then Delete(liczba,1,1);
```

```
MoveTo(x,y);
```

```
UsunText(dl);
```

```
OutText(liczba+'_');
```

```
until znak=#13;
```

```
Val(liczba,l,rep);
```

```
if rep>0 then liczba:=Copy(liczba,1,rep-1);
```

```
until rep=0;
```

```
    PodajLiczbe:=1;
End; {of PodajLiczbe}

Function PodajNazwe(x,y:Integer; var Dl:Byte):Nazwa;
var znak:Char;
    N :nazwa;
Begin
n:='';
OutTextXY(x,y,'_');
repeat
    znak:=UpCase(ReadKey);
    if znak=#8 then Delete(n,Length(n),1);
    if znak=#27 then Begin dl:=0; PodajNazwe:=''; Exit end;
    if znak in ['0'..'9','*','.',':','?','\','A'..'Z'] then n:=n+znak;
    if Length(n)>dl then Delete(n,1,1);
    MoveTo(x,y);
    UsunText(dl);
    OutText(n+'_');
until znak=#13;
    PodajNazwe:=n;
    Dl:=Length(n)+1;
End; {of PodajNazwe}

Function Sign(x:Real):ShortInt;
begin
    if x<0 then Sign:=-1 else Sign:=1;
end; {of Sign}

Procedure StrPion(zwrot:Integer); {1=góra, -1=dół}
const szer=3;
    dl=6;
    od=4;
begin
    LineRel(0,-zwrot*(dl+od));
```

```
LineRel (szer, zwrot*d1);  
LineRel (-2*szer, 0);  
LineRel (szer, -zwrot*d1);  
end; {of StrPion}
```

```
Procedure StrPoz (zwrot:ShortInt); {1=prawo, -1=lewo}  
const szer=2;  
      dl=8;  
      od=6;  
begin  
  LineRel (zwrot*(dl+od), 0);  
  LineRel (-zwrot*dl, szer);  
  LineRel (0, -2*szer);  
  LineRel (zwrot*dl, szer);  
end; {of StrPoz}
```

```
Procedure Grot (kier:ShortInt);  
begin  
  case kier of  
    1:StrPoz(1);  
   -1:StrPoz(-1);  
    2:StrPion(1);  
   -2:StrPion(-1);  
  end;  
end; {of Grot}
```

```
Procedure AxisY(x,y,dl,skok:Integer; komunikat:nazwa;  
               pocz,sk:Real; im,pp,podz:Byte);  
var i,j      :Integer;  
    szer     :Byte;
```

```
Procedure OpisOsiY;  
var k:Byte;
```



```
    j: Integer;  
    a: string[6];  
begin  
    k:=Trunc(Abs(d1)/skok/podz);  
    Str(pocz:im:pp,a);  
    MoveTo(x-TextWidth(a)-2,y-4);  
    OutText(a);  
    for j:=1 to k do  
        begin  
            Str(pocz+j*sk*podz:im:pp,a);  
            MoveTo(x-TextWidth(a)-5,y-3-j*podz*skok*Sign(d1));  
            OutText(a);  
        end;  
    end; {of Opis0siY}
```

```
begin  
MoveTo(x,y);  
LineRel(0,-d1);  
Grot(Sign(d1)*2);  
i:=Trunc(Abs(d1)/skok);  
for j:=1 to i do  
    begin  
        if (j mod podz)=0 then szer:=4 else szer:=2;  
        MoveTo(x-szer,y-j*skok*Sign(d1));  
        LineRel(2*szer,0);  
    end;  
MoveTo(x+10,y-d1-10);  
OutText(komunikat);  
MoveTo(x,y);  
Opis0siY;  
end; {AxisY}
```

```
Procedure AxisX(x,y,d1,skok: Integer;
```

```
        komunikat:nazwa; pocz,sk:Real;
im,pp,podz:Byte);
var i,j:Integer;
const szer1=2;
      szer2=3;

Procedure OpisOsiX;
var k:Byte;
    j:Integer;
    a:string[6];
begin
    k:=Trunc(Abs(dl)/skok/podz);
    Str(pocz:im:pp,a);
    if a[1]=' ' then delete(a,1,1);
    MoveTo(x-Trunc(TextWidth(a)/2)-10,y+4);
    OutText(a);
    for j:=1 to k do
        begin
            Str(pocz+j*sk:im:pp,a);
            MoveTo(x-Round(TextWidth(a)/2)+j*skok*podz*Sign(dl),y+4);
            OutText(a);
        end;
end; {of OpisOsi}

begin
    MoveTo(x,y);
    LineRel(dl,0);
    Grot(Sign(dl));
    i:=Trunc(Abs(dl)/skok);
    for j:=1 to i do
        begin
            if (j mod podz)=0 then
                begin
                    MoveTo(x+j*skok*Sign(dl),y-szer2);
```

```
        LineRel (0,2*szer2);
    end
    else
    begin
        MoveTo(x+j*skok*Sign(dl),y-szer1);
        LineRel (0,2*szer1);
    end;

    end;
MoveTo(x+dl-Round(TextWidth(komunikat)/2),y-12);
OutText(komunikat);
OpisOsix;
MoveTo(x,y);
end; {of AxisX}
```

Procedure Dysk;

const

```
    x1=300;
    y1=22;
    wys=60;
    szer=400;
```

```
var Size1    :Word;
    Size2    :Word;
    p,q      :Pointer;
    x,y      :Integer;
    i,j,k    :Integer;
    x2,y2    :Integer;
    liczba   :nazwa;
    znak     :Char;
    NazwaZb  :nazwa;
    PozKat   :SearchRec;
    Error    :Boolean;
    ParamF   :file of Par;
    dlug     :Byte;
```

```
rep      : Integer;

Procedure NazwaPar( Var Naz:Nazwa);
var NazwaZb:Nazwa;
Begin
  NazwaZb:=PodajNazwe(GetX,GetY,dlug);
  if NazwaZb<>' '
  then
    begin
      j:=Pos(' ',NazwaZb);
      if j=0 then NazwaZb:=NazwaZb+'.PAR';
      if j=1 then NazwaZb:='PARAM.PAR';
      Naz:=NazwaZb;
    end;
End; {of NazwaPar}

Begin
x2:=x1+szer;
y2:=y1+wys;
i:=1;
MoveTo(Odst+3*Round(MaxX/4),12);RetypeInv(DlOpcji);
  Size1:=Okno(x1,y1,x2,y2,p);
  Ustaw(x1,y1,8,2);
  OutText('F1-Katalog dysku      : '+MaskaKat);
  Ustaw(x1,y1,8,12);
  OutText('F2-Nazwa podkatalogu   : '+Katalog);
  Ustaw(x1,y1,8,22);
  OutText('F3-Laduj parametry z     : '+NazParLad);
  Ustaw(x1,y1,8,32);
  OutText('F4-Zapisz parametry do    : '+NazParZap);
  Ustaw(x1,y1,8,42);
  OutText('F5-Nazwa zbioru z wynikami: '+NazZbDane);
```

```
repeat
  i:=Menu(x1,y1,27,5,i);
  y:=GetY+12;
  Case i of
    1:Begin
      dlug:=16;
      Size2:=Okno(50,40,500,340,q);
      Ustaw(50,40,10,2);
      OutText("Maska katalogu ('+MaskaKat+') : ");
      NazwaZb:=PodajNazwe(GetX,GetY,dlug);
      if dlug>0 then
        Begin
          if NazwaZb>' ' then MaskaKat:=NazwaZb;
          if MaskaKat[Length(MaskaKat)]=':' then
            MaskaKat:=MaskaKat+'\'';
          if MaskaKat[Length(MaskaKat)]='\' ' then
            MaskaKat:=MaskaKat+'%.*';
          MoveTo(x1+248,y1+2);
          UsunText(16);
          OutText(MaskaKat);
          Str(DiskFree(0),liczba);
          Ustaw(50,40,10,288);
          OutText('          '+liczba+' bajtow wolnych');
          FindFirst(MaskaKat,AnyFile,PozKat);
          j:=0;k:=0;
          While DosError=0 do
            begin
              Ustaw(50,40,10+k*150,12+j*10);
              OutText(PozKat.Name);
              FindNext(PozKat);
              j:=j+1;
              if j=27 then begin j:=0; k:=k+1 end;
              if k=3 then
                begin
```

```
        j:=0;k:=0;
        repeat until KeyPressed;
        ClearImage(60,52,499,339);
        end;

        end;
        end;
        repeat until ReadKey=#27;
        KasOkno(50,40,q,Size2);
        end;
2:Begin
        dlug:=20;
        Size2:=Okno(16,Y,x2-4,Y+20,q);
        MoveTo(20,Y+2);
        OutText('
Nazwa nowego podkatalogu ('+katalog+' ) : ');
        Liczba:=PodajNazwe(GetX,GetY,dlug);
        if dlug>0 then
        begin
        if liczba>' ' then
        begin
        {$I-}
        ChDir(liczba);
        if IOResult<>0 then
                Begin
                OutText(' Bledna nazwa !!! ');
                repeat until KeyPressed;
                end
                else Katalog:=Liczba;

        {$I+}
        end;
        end;
        end;
        KasOkno(16,Y,q,Size2);
        MoveTo(x1+232,y1+12);
        UsunText(16);
```

```
    OutText (Katalog);
end;
3:Begin
    dlug:=12;
    Size2:=Okno(16, Y, x2-4, Y+20, q);
    MoveTo(20, Y+2);
    OutText('
Laduj parametry ze zbioru ('+NazParLad+') : ');
    NazwaPar (NazParLad);
    MoveTo(x1+232, y1+22);
    UsunText(16);
    OutText (NazParLad);
    If dlug>0 then
        Begin
            {#I-}
            Assign (ParamF, NazParLad);
            Reset (ParamF);
            {#I+}
            if IOResult<>0 then
                Begin {3}
                    OutTextXY(200, y+10, ' Zbioru nie znaleziono !!!');
                    repeat until KeyPressed;
                end {3}
            else
                Begin {4}
                    Read (ParamF, Parametr);
                    Close (ParamF);
                end; {4}
            end;
        end;
        KasOkno(16, Y, q, Size2);
    end;
4:Begin
    dlug:=12;
    Size2:=Okno(16, Y, x2-4, Y+20, q);
```

```
MoveTo(20, Y+2);
OutText('Zapisz parametry do zbioru ('+NazParZap+') : ');
NazwaPar(NazParZap);
MoveTo(x1+232, y1+32);
UsunText(16);
OutText(NazParZap);
If Dlug>0 then
  Begin
    Assign(ParamF, NazParZap);
    Rewrite(ParamF);
    Write(ParamF, Parametr);
    Close(ParamF);
  end;
  KasOkno(16, Y, q, Size2);
end;
5:Begin
  dlug:=12;
  Size2:=Okno(16, Y, x2-4, Y+15, q);
  MoveTo(20, Y+2);
  OutText('Nazwa zbioru z wynikami ('+NazZbDane+') : ');
  NazwaZb:=PodajNazwe(GetX, GetY, dlug);
  if NazwaZb>' '
  then
  Begin
    NazZbDane:=NazwaZb;
    j:=Pos(' ', NazZbDane);
    if j=0 then NazZbDane:=NazZbDane+'.DTA';
    if j=1 then NazZbDane:='WYNIKI.DTA';
    MoveTo(x1+232, y1+42);
    UsunText(16);
    OutText(NazZbDane);
  end;
  if Dlug>0 then
  begin
```



```
        if ZbWynOtw then Close(ZbDane);
        Assign(ZbDane, NazZbDane);
        Rewrite(ZbDane);
        ZbWynOtw:=True;
    end;
    KasOkno(16, Y, q, Size2);
end;
end;
until i=0;
KasOkno(x1, y1, p, Size1);
MoveTo(Odst+3*Round(MaxX/4), 12); ReTypeInv(DlOpcji);
End; {of Dysk}
```

Procedure MenuG1;

```
var i:Byte;
    z:Char;
    ip:Byte;
Begin
i:=M;
ip:=Round(MaxX/4);
MoveTo(Odst+(i-1)*ip, 12);
ReTypeInv(DlOpcji);
repeat
z:=UpCase(ReadKey);
if z=#0 then z:=UpCase(ReadKey);
MoveTo(Odst+(i-1)*ip, 12);
ReTypeInv(dlOpcji);
Case z of
#75:i:=i-1;
#77:i:=i+1;
#27:begin
M:=0;
Exit
End;
```

```
'W':begin i:=1; z:=#13; end;
'O':begin i:=2; z:=#13; end;
'P':begin i:=3; z:=#13; end;
'D':begin i:=4; z:=#13; end;
end;
if i<1 then i:=1;
if i>4 then i:=4;
MoveTo(Odst+(i-1)*ip,12);ReTypeInv(Dl0pcji);
until z=#13;
M:=i;
MoveTo(Odst+(i-1)*ip,12);ReTypeInv(Dl0pcji);
end; {of MenuG1}
```

```
Function Menu(x,y:Integer; dl:Byte; max,nr:Byte):Byte;
var i:Byte;
    z:Char;
Begin
i:=nr;
repeat
    MoveTo(x+2,Y+2+(i-1)*10);
    ReTypeInv(dl);
    z:=UpCase(ReadKey);
    MoveTo(x,y);
    MoveRel(2,2+(i-1)*10);
    ReTypeInv(dl);
    case z of
        #72:i:=i-1;
        #80:i:=i+1;
        #71:i:=1;
        #27:Begin Menu:=0; Exit end;
    end;
    if i<1 then i:=1;
    if i>max then i:=max;
until z=#13;
```

```
MoveTo(x+2,Y+2+(i-1)*10);
Menu:=i;
End; {of Menu}
```

```
Function PodajLiczbe(x,y:Integer; var dl:Byte):Real;
var znak:Char;
    l :Real;
    rep :Integer;
    liczba:Nazwa;
```

```
Begin
liczba:='';
repeat
repeat
znak:=UpCase(ReadKey);
if znak=#8 then Delete(liczba,Length(liczba),1);
if znak=#27 then begin dl:=0; Exit end;
if znak in ['0'..'9', '.', '-', 'E'] then liczba:=liczba+znak;
if Length(liczba)>dl then Delete(liczba,1,1);
MoveTo(x,y);
UsunText(dl);
OutText(liczba);
until znak=#13;
Val(liczba,l,rep);
if rep>0 then liczba:=Copy(liczba,1,rep-1);
until rep=0;
PodajLiczbe:=l;
End; {of PodajLiczbe}
```

```
Function PodajNazwe(x,y:Integer; var Dl:Byte):Nazwa;
var znak:Char;
    N :nazwa;
Begin
n:='';
```

```
repeat
  znak:=UpCase(ReadKey);
  if znak=#8 then Delete(n,Length(n),1);
  if znak=#27 then Begin dl:=0; PodajNazwe:=''; Exit end;
  if znak in ['0'..'9','*','.',':','?','\','A'..'Z'] then n:=n+znak;
  if Length(n)>dl then Delete(n,1,1);
  MoveTo(x,y);
  UsunText(dl);
  OutText(n);
until znak=#13;
PodajNazwe:=n;
dl:=Length(n)+1;
End; {of PodajNazwe}
```

Procedure Param;

const

```
  x1=200;
  y1=22;
  wys=260;
  szer=450;
  dlmenu=25;
```

```
var Size1 :Word;
    Size2 :Word;
    p,q :Pointer;
    x,y :Integer;
    i1,i2 :Integer;
    x2,y2 :Integer;
    liczba :nazwa;
    znak :Char;
    Error :Boolean;
    dlug :Byte;
    par :Real;
    dl :Byte;
```

```
a      :array[1..dimenu] of Real;

Procedure zap;
begin
With Parametr do
begin
a[1]:=Ku;
a[2]:=Mmax*1000;
a[3]:=J*1000;
a[4]:=Mr*1000;
a[5]:=D;
a[6]:=R;
a[7]:=L0*1000;
a[8]:=L2*1000;
a[9]:=Zr;
a[10]:=K;
a[11]:=U;
a[12]:=Delta_t*1000;
a[13]:=Okres*1000;
a[14]:=IIPkt0bl;
a[16]:=Kat;
a[17]:=1000/Omega_0;
a[18]:=Mi_r;
a[19]:=Okres*Omega_0;
a[15]:=Fmax;
a[20]:=ksi_0;
a[21]:=ksi_2;
a[22]:=beta_e;
a[23]:=beta_m;
a[24]:=LR*1000;
a[25]:=Kr;
end;
end; {of zap}
```

```
Procedure Wpisz;  
begin  
    Ustaw(x1,y1,248,(i2-1)*10+4);  
    UsunText(20);  
    Str(a[i2]:12:4,liczba);  
    OutText(liczba+JednPar[i2]);  
end; {of wpisz}  
  
Begin  
With Parametr do  
begin  
zap;  
MoveTo(Odst+2*Round(MaxX/4),12);RetypeInv(D10pcji);  
i1:=1;  
x2:=x1+szer;  
y2:=y1+wys;  
Size1:=Okno(x1,y1,x2,y2,p);  
PrintMenu(x1+4,y1,dlmenu,27,MenuPar);  
for i2:=1 to dlmenu do  
begin  
    Ustaw(x1,y1,248,(i2-1)*10+4);  
    UsunText(20);  
    if i2 in [9,10,14,15] then Str(a[i2]:7:0,liczba) else  
Str(a[i2]:12:4,liczba);  
    OutText(liczba+JednPar[i2]);  
end;  
repeat  
dlug:=12;  
i1:=Menu(x1,y1+2,29,15,i1);  
if i1>0 then  
begin  
y:=GetY+5;  
Size2:=Okno(60,y,680,y+14,q);
```

```
Ustaw(60,y,4,3);
OutText(PytPar[i1]);
par:=PodajLiczbe(GetX,GetY,dlug);
if dlug>0 then
begin
  Case 11 of
    1: begin
      if par>1E-6 then Ku:=par;
      MMax:=Ku*(U/R);
      end;
    3: if par>1E-7 then J:=par;
    4: Mr:=par/1000;
    5: if par>1E-6 then D:=par;
    13: if par>9000*Delta_t then Okres:=par/1000
      else Okres:=Delta_t*10;
    10: if par>=1 then K:=Trunc(par);
    11: if par>1E-3 then U:=par;
    6 : if par>1E-6 then R:=par;
    7 : if par>1E-6 then L0:=par/1000;
    8 :if par>1E-6 then L2:=par/1000;
    9 :if par>=2 then Zr:=Trunc(par);
    12:if par>0 then Delta_t:=par/1000;
    14: begin
      case Trunc(par) of
        1: IIPktObl:=125;
        2: IIPktObl:=250;
        3: IIPktObl:=375;
        4: IIPktObl:=500;
      end;
      I1Wykr:=Round(IIPktObl/125);
      end;
    2 :begin
      if par>1E-6 then Mmax:=par/1000;
      Ku:=Mmax/(U/R);
```

```
        end;
        15:Fmax:=par;
        end;
end;
KasOkno(60,y,q,Size2);
ObliczParametry;
Zap;
Ustaw(x1,y1,248,(i1-1)*10+4);
UsunText(20);
if i1 in [9,10,14,15] then Str(a[i1]:7:0,liczba)
        else Str(a[i1]:12:4,liczba);
        OutText(liczba+JednPar[i1]);
        for i2:=16 to dmenu do Wpisz;
        i2:=1; Wpisz;
        i2:=2; Wpisz;
        end;
until i1=0;
KasOkno(x1,y1,p,Size1);
MoveTo(0dst+2*Round(MaxX/4),12);RetypeInv(D1Opcji);
end; {of with}
End; {of Parametr}

Procedure Screen;
const
    menu:array [1..4] of string[20]=('Wykresy ', 'Obliczenia ',
        'Parametry ', 'Dysk ');
var
    xpoz:Word;
    i,ip:Byte;
    Year,Month,Day,DayofWeek:Word;
    Data:string[20];
    Poz:string[5];
begin
    GetDate(Year,Month,Day,DayofWeek);
```



```
Str (Year, Poz);
Data:=Poz;
Str (Month, Poz);
if Length(poz)<2 then Poz:='0'+Poz;
Data:=Data+'-'+poz;
Str (Day, poz);
if Length(poz)<2 then Poz:='0'+Poz;
Data:=Data+'-'+Poz+'  ';
case DayOfWeek of
  0: Data:=Data+' Niedz. ';
  1: Data:=Data+' Pon.  ';
  2: Data:=Data+' Wt.   ';
  3: Data:=Data+' Sr.   ';
  4: Data:=Data+' Czw.  ';
  5: Data:=Data+' Pt.   ';
  6: Data:=Data+' Sob.  ';
end;
ip:=Round (MaxX/4);
MoveTo(0, 10);
LineRel (MaxX, 0);
Napis(0, 2, MaxX, Tytul, True);
  for i:=1 to 4 do
    Napis((i-1)*ip, 12, ip, menu[i], False);
    OutTextXY(4, 2, Data);
    OutTextXY(530, 2, '(c) JKK '1988.11 Polska');
end; {of Screen}

Procedure Ramka;
begin
  Rectangle(0, 0, MaxX, MaxY);
  MoveTo(0, 20);
  LineRel (MaxX, 0);
  MoveTo(0, 0);
end; {of Ramka}
```

```
Procedure Ekran;  
Begin  
MaxX:=GetMaxX;  
MaxY:=GetMaxY;  
SetViewPort (0,0,MaxX,MaxY,ClipOn);  
ClearViewPort;  
Ramka;  
Screen;  
End; {of Ekran}
```

```
Procedure PrintText(x,y,d1:Integer; s:nazwa);  
begin  
MoveTo(x,y);  
UsunText(d1);  
OutText(s);  
end; {of PrintText}
```

```
type mat=array [1..10] of Real;
```

```
var  
    n :Integer;  
    h :real;
```

{ silnik skokowy hybrydowy o kacie skoku 90 deg przy
zasilaniu miniskokowym ze zrodla pradowego }

```
Procedure gprim(var x,xprim:mat; i:Integer);  
var Me:Real;  
begin  
with parametr do
```

```
begin
Me:=sin((i+1)*kat-x[2]);
xprim[1]:=1.0;
xprim[2]:=x[3];
xprim[3]:=Me-2*beta*x[3]-Mi_r*Sign(x[3]);
end;
end; {of gprim}
```

```
procedure RKG4(var x,q:mat; n:integer; h:real; nr:Integer);
```

```
{*****
* Procedura rozwiazuje ukklad rownan roznicekowych zwyczajnych *
* metoda Rungego - Kutty - Gilla 4-go rzędu *
*****}
```

```
max liczba rownan = 10;
przed pierwszym wywołaniem x[n] zawiera warunki początkowe;
q[n] = 0;
h-okresla krok całkowania
n-okresla ilość równań;
fprim jest procedura obliczająca pochodne
xprim[n] w punkcie x[n];
Procedura jest deklarowana w postaci
"Procedure fprim(var x,xprim:mat);}
```

```
type MatPom=array [1..4] of real;
const a:MatPom=(0.5,0.292893219,1.707106781,0.1666666667);
b:MatPom=(0.5,0.292893219,1.707106781,0.3333333333);
c:MatPom=(0.5,0.292893219,1.707106781,0.5);
var xprim:mat;
i,j:integer;
ak,r:real;
```

```
begin
  for j:=1 to 4 do begin
    gprim(x,xprim,nr);
    for i:=1 to n do begin
      ak:=h*xprim[i];
      r:=a[i]*ak-b[i]*q[i];
      x[i]:=x[i]+r;
      q[i]:=3.0*r+q[i]-c[i]*ak;
    end;
  end;
end; (of RK64)
```

```
Procedure WartPocz(var x,q:mat);
var i:Integer;
begin
  x[1]:=0.0;
  x[2]:=0.0;
  x[3]:=0.0;
  for i:=1 to 10 do q[i]:=0.0;
  for i:=0 to Parametr.IIPktObl do
    begin
      with Wykres[i] do begin
        czas:=0;
        kat_fi:=0;
        predk:=0;
      end;
    end;
end; (WartPocz)
```

```
Procedure Oblicz;
const
  hmin=1E-5;
  hmax=9E-2;
  emin=2E-4;
```

```
emax=6E-4;

var i,j   :Integer;
    x,q   :mat;
    x1,x2 :mat;
    p     :Pointer;
    S     :Word;
    z     :Char;
    h,eps:Real;
    ll    :string[20];
    skok  :Integer;
    rw    :Byte;
    fi    :Real;
    Omega :Real;
    Omega_krok:Real;

Procedure CharCzas;
label koniec;
var i,j:Integer;
begin
    S:=Okno(240,130,480,206,p);
    Napis(240,134,240,'OBLICZENIA W TOKU',False);
    OutTextXY(245,144,'Eps=');
    OutTextXY(245,154,' h=');
    OutTextXY(245,164,' t=');
    OutTextXY(245,174,' fi=');
    OutTextXY(245,184,' v=');
    OutTextXY(245,194,' nr=');
    ObliczParametry;
    WartPocz(x,q);
    Wykres[0].czas:=x[1];Wykres[0].kat_fi:=x[2];
    Wykres[0].predk:=x[3];
    Wykres[0].blad:=Parametr.Kat;
```

```
RK64(x, q, 3, hmin, 1);
  h:=hmin;
  eps:=0.0;
  for i:=1 to Parametr.IIPkt0b1 do
  begin
    skok:=Trunc(i*Parametr.Delta_t/Parametr.Okres);
    repeat
  (***** petla calkowania ze zmiennym krokiem *****)
    repeat
      x1:=x;
      x2:=x;
      RK64(x1, q, 3, h, skok);
      RK64(x2, q, 3, 2*h, skok);
      eps:=0.0;
      for j:=2 to 3 do if Abs(x2[j]-x1[j])>eps then
        eps:=Abs(x2[j]-x1[j]);
      if eps>emax then h:=h/2;
      if eps<emin then h:=2*h;
      if h<hmin then h:=hmin;
      if h>hmax then h:=hmax;
      if KeyPressed then begin z:=ReadKey; if z=#27 then
        GoTo Koniec; end;
    until (eps<=emax);
    x:=x1;
  (*****)
  until x[1]/Parametr.Omega_0>=i*Parametr.Delta_t;
  Str(x[1]/Parametr.Omega_0:16,11); PrintText(285,164,17,11);
  Str(x[2]:16,11); PrintText(285,174,17,11);
  Str(x[3]:16,11); PrintText(285,184,17,11);
  Str(h:16,11); PrintText(285,154,17,11);
  Str(eps:16,11); PrintText(285,144,17,11);
  Str(i:4,11);PrintText(285,194,5,11);
  Wykres[i].czas:=x[1]/Parametr.Omega_0;
  Wykres[i].kat_fi:=x[2];
```

```
Wykres[i].predk:=x[3];
Wykres[i].blad:=(skok+1)*Parametr.kat-x[2];
if KeyPressed then begin Delay(250); z:=ReadKey;
  z:=ReadKey end;
end;
  MoveTo(242,134);
  UsunText(28);
  Napis(240,134,240,'Zapisac wyniki (T/N)?: ',False);
z:=UpCase(ReadKey);
if z='T' then begin
if Not ZbWynOtw then begin Rewrite(ZbDane);
  ZbWynOtw:=True end;
  for i:=0 to Parametr.IIPktObl do
    With Wykres[i] do
      begin
WriteLn(ZbDane,'t=',czas:11,' Fi_e =',kat_fi:16,'
  Omega=',predk:16);
      end;
    end;
  koniec;
  KasOkno(240,130,p,S);
end; {of CharCzas}
```

```
Procedure CharMoment;
label koniec;
var i:Integer;
begin
  S:=Okno(240,130,480,206,p);
  Napis(242,132,240,'Charakterystyka momentu',false);
  with Parametr do
    begin
      Omega:=pi*fmax/2/K;
      Omega_krok:=Omega/IIPktObl;
      Omega:=0.0;
```

```
PrintText(245,162,20,'L/R='+StrVal(LR,12,13));
PrintText(245,172,20,'Kr =' +StrVal(Kr,12,13));
PrintText(245,142,20,'fi =');
PrintText(245,152,20,'Me =');
for i:=0 to Parametr.IIPktObl do
begin
  CharMom[i].f:=2*Parametr.K*Omega/pi;
  CharMom[i].M:=1/(Sqrt(1+Sqr(Omega)*Sqr(LR))
    -Kr*Omega*LR/(1+Sqr(Omega)*Sqr(LR)));
  PrintText(280,142,8,StrVal(CharMom[i].f,8,0));
  PrintText(280,152,8,StrVal(CharMom[i].M,8,5));
  Omega:=Omega+Omega_krok;
  if KeyPressed then
  begin
    z:=ReadKey;
    if z=#27 then GoTo Koniec
      else z:=ReadKey;
  end;
end;
PrintText(264,132,24,'Zapisac wyniki (T/N) ?: ');
z:=UpCase(ReadKey);
if z='T' then begin
  if Not ZbWynOtw then begin
Rewrite(ZbDane); ZbWynOtw:=True end;
    for i:=0 to Parametr.IIPktObl do
      With CharMom[i] do
        begin
          Writeln(ZbDane,'f=',f:11,' M=',M:16);
        end;
      end;
  end;
end; {of with Parametr}
koniec:
  KasOkno(240,130,p,s);
end; {of CharMoment}
```



```
begin
MoveTo(Odst+Round(MaxX/4), 12); RetypeInv(D10pcji);
S:=Okno(240, 130, 484, 154, p);
  Napis(240, 133, 240, 'Wykres czasowy', false);
  Napis(240, 143, 240, 'Charakterystyka momentu', false);
  rw:=Menu(240, 132, 30, 2, 1);
  KasOkno(240, 130, p, S);
  case rw of
    1: CharCzas;
    2: CharMoment;
  end;
MoveTo(Odst+Round(MaxX/4), 12); RetypeInv(D10pcji);
end; (of Oblicz)

(wyswietlanie wykresu)
Procedure Wyswietlanie;
const
  x1=10;
  y1=25;
  szer=700;
  wys=320;

var p:Pointer;
  s:Word;
  i, j:Integer;
  x2, y2:Integer;
  x0, y0:Integer;
  z, w:char;
  skala:Byte;
  dzielnik:Integer;

const IIPunktow=125;
```

```
d10siY=200;
d10siX=500;

const a=20;
      b=240;
      c=480;

begin
  with parametr do
    begin
      OutTextXY(x1+a,y2-60,'Ku      = '+StrVal(Ku,B,4)+' [V*s/rad]');
      OutTextXY(x1+a,y2-50,'J      = '+StrVal(J*1000,B,4)+' [g*m*m]');
      OutTextXY(x1+a,y2-40,'Mr     = '+StrVal(Mr,B,4)+' [Nm]');
      OutTextXY(x1+a,y2-30,'D      = '+StrVal(D,B,4)+' [Nms/rad]');
      OutTextXY(x1+a,y2-20,'K      = '+StrVal(K,B,0));
      OutTextXY(x1+a,y2-10,'U      = '+StrVal(U,B,3)+' [V]');
      OutTextXY(x1+b,y2-60,'R      = '+StrVal(R,B,3)+' [Om]');
      OutTextXY(x1+b,y2-50,'LO     = '+StrVal(LO*1000,B,3)+' [mH]');
      OutTextXY(x1+b,y2-40,'L2     = '+StrVal(L2*1000,B,3)+' [mH]');
      OutTextXY(x1+b,y2-30,'Ti     = '+StrVal(Okres*1000,B,0)+' [ms]');
      OutTextXY(x1+b,y2-20,'Delta t= '+StrVal(Delta_t*1000,B,3)+' [ms]');
      OutTextXY(x1+b,y2-10,'Mmax   = '+StrVal(Mmax,B,5)+' [Nm]');
      OutTextXY(x1+c,y2-60,'T0     = '+StrVal(1000/Omega_0,B,3)+' [ms]');
      OutTextXY(x1+c,y2-50,'Kat    = '+StrVal(kat,B,5)+' [rad]');
      OutTextXY(x1+c,y2-40,'Zr     = '+StrVal(Zr,B,0));
      OutTextXY(x1+c,y2-30,'Mr/Mmax= '+StrVal(Mi_r,B,5));
      OutTextXY(x1+c,y2-20,'L/R    = '+StrVal(LR*1000,B,5)+' [ms]');
      OutTextXY(x1+c,y2-10,'Kr     = '+StrVal(Kr,B,5));
    end;
  end;

  Procedure RysWykrCzas(x0,y0:Integer; skalaY:Real);
  var
    CzasMin,CzasMax    :Real;
```

```
KatMin          : Real;
CzasSkok, KatSkok : Real;
Skok            : Integer;
i, j, k        : Integer;
liczba         : nazwa;

Function Fwykr(nr: Integer): Integer;
begin

FWykr:=y0-Round((Wykres[nr].kat_fi-skala*Parametr.Kat)*DlOsiY/Parametr.kat/2);
end; {of FWykr}

begin
j:=0;
dzielnik:=2;
skala:=0;
repeat
  CzasMin:=j*IIPunktow*Parametr.Delta_t;
  CzasMax:=((j+1)*IIPunktow)*Parametr.Delta_t;
  CzasSkok:=(CzasMax-CzasMin)/5*1000;
  ClearImage(0,0,x2-x1-2,y2-65-y1);
AxisY(x0,y0,dlOsiY+20,10,'skoki ',skala,0.1,3,1,5);
AxisX(x0,y0,dlOsiX+10,20,'czas [ms]',CzasMin*1000,CzasSkok,5,2,5);

MoveTo(x0,FWykr(j*IIPunktow));
for i:=1 to IIPunktow do LineTo(x0+i*4,FWykr(i+J*IIPunktow));

w:=UpCase(ReadKey);
if w=#0 then z:=ReadKey;
case z of
  #77:if j<Parametr.IIWykr-1 then j:=j+1;
  #75:if j>0 then j:=j-1;
```

```
#71: j:=0;
#79: j:=Parametr.IIWykr-1;
#72: skala:=skala+1;
#80: if skala>0 then skala:=skala-1;
#81: skala:=0;
end;
Case w of
  'P':begin
    SetViewport(0,0,MaxX,MaxY,ClipOn);
    Param;
    Oblicz;
    ClearImage(x1+1,y2-62,x2-1,y2-1);
    WyswPar;
    SetViewport(x1+1,y1+1,x2-1,y2-64,ClipOn);
  end;
end;
until w=#27;
end; {of RysWykr}

Procedure RysWykrFaz;
begin
end; {of RysWykrFaz}

Procedure RysWykrMoment;
var  y0: Integer;
     z: Char;
     F_Skok, F_min, F_max: Real;
     i, j: Integer;

Function Fwykr(nr: Integer): Integer;
begin
  if CharMom[nr].M>=0 then Fwykr:=y0-Round(D10siY*CharMom[nr].M)
    else Fwykr:=y0;
```

```
end; {of Fwykr}

begin
  y0:=y2-80-y1;
  j:=0;
  repeat
    F_Min:=j*fmax/Parametr.IlWykr;
    F_Max:=(j+1)*Parametr.fmax/Parametr.IlWykr;
    F_Skok:=(F_Max-F_Min)/5;
    ClearImage(0,0,x2-x1-2,y2-65-y1);
    AxisY(x0,y0,dl0siY+20,10,'Moment ',0,0.05,3,1,5);
    AxisX(x0,y0,dl0siX+10,20,'fi',F_Min,F_Skok,5,0,5);
    MoveTo(x0,FWykr(j*IlPunktow));
  for i:=1 to IlPunktow do LineTo(x0+i*4,FWykr(i+J*IlPunktow));

  w:=UpCase(ReadKey);
  if w#0 then z:=ReadKey;
  case z of
    #77:if j<Parametr.IlWykr-1 then j:=j+1;
    #75:if j>0 then j:=j-1;
    #71:j:=0;
    #79:j:=Parametr.IlWykr-1;
  end;
  Case w of
    'P':begin
      SetViewPort(0,0,MaxX,MaxY,ClipOn);
      Param;
      Oblicz;
      ClearImage(x1+1,y2-62,x2-1,y2-1);
      WyswPar;
      SetViewPort(x1+1,y1+1,x2-1,y2-64,ClipOn);
    end;
  end;
end;
```

```
until w=#27;
end; {of RysWykrMoment}

var rw:Byte;

begin
MoveTo(Odst,12);RetypeInv(D10pcji);
x2:=x1+szer;
y2:=y1+wys;
S:=Okno(x1,y1,x1+244,y1+32,p);
  Napis(x1,y1+3,240,'Wykres czasowy',false);
  Napis(x1,y1+13,240,'Wykres fazowy',false);
  Napis(x1,y1+23,240,'Charakterystyka momentu',false);
  rw:=Menu(x1,y1+1,30,3,1);
KasOkno(x1,y1,p,s);
S:=Okno(x1,y1,x2,y2,p);
x0:=110;
Line(x1,y2-63,x2,y2-63);
WyswPar;
SetViewPort(x1+1,y1+1,x2-1,y2-64,ClipOn);
ClearViewPort;
  case rw of
    1:RysWykrCzas(x0,y2-80-y1,0.5);
    2:RysWykrFaz;
    3:RysWykrMoment;
  end;
SetViewPort(0,0,GetMaxX,GetMaxY,ClipOn);

KasOkno(x1,y1,p,s);
MoveTo(Odst,12);RetypeInv(D10pcji);
end; {of Wyszwietlanie}
```

ZAŁĄCZNIK B.

PROGRAM STERUJĄCY SILNIKIEM PRZY PRACY MINISKOKOWEJ

Do sterowania silnikiem skokowym przy pracy miniskokowej został napisany program w języku Pascal. Zapewnia on uzyskanie maksymalnej liczby miniskoków w skoku bazowym do 100 i maksymalnej częstotliwości pracy do 5000 Hz. Poniżej przedstawiono treść programu.

Program Miniskok; (12-4)
{C+}

LABEL S,P;

```
const RejSter=$F887;  
      RLicz0=$F884;  
      RLicz1=$F885;  
      RLicz2=$F886;  
      ModLicz0=$3E;  
      ModLicz1=$52;  
      ModLicz2=$92;  
      PortDod=$F888;  
      IloscSkokowSilnika=100;
```

```
var TabSinCos: array [0..2048] of Byte;  
    IloscMiniskok,Odstep,ik:byte;  
    Okr,I:Integer;  
    znak:char;  
    a,b,kat,op:Real;  
    podgl:Boolean;
```

```
type HexStr=string[4];
```

```
Function Hex(Number:Integer; Bytes:Integer):HexStr;  
const
```

```
  T:array [0..15] of char='@123456789ABCDEF';  
var  
  D:Integer;  
  H:HexStr;
```

```
Begin
```

```
  H[0]:=chr(Bytes);  
  for d:=Bytes downto 1  
  do begin  
    H[d]:=T[(Number and 15)];  
    Number:=Number shr 4;  
  end;  
  Hex:=H;  
end; { of Hex function }
```

```
Function ReadKey:Char;
```

```
var r:Char;  
Begin  
  repeat until KeyPressed;  
  Read(Kbd,R);  
  ReadKey:=r;  
end; {of ReadKey}
```



```
Procedure Init;
begin
  Port[RejSter]:=ModLicz0;
  Port[RejSter]:=ModLicz1;
  Port[RejSter]:=ModLicz2;
  Port[RLicz0]:=0;
  Port[RLicz0]:=1;
  Port[RLicz1]:=TabSinCos[0];
  Port[RLicz2]:=TabSinCos[1];
  Port[PortDod]:=TabSinCos[2];
end; {Koniec inicjalizacji}
```

```
Procedure ZapisTabeli;
var f:text;
    i:integer;
    nazwa:String[8];
begin
  Write('Nazwa zbioru do zapisu? ');
  Readln(Nazwa);
  Assign(f,nazwa+'.MSK');
  Rewrite(f);
  Writeln(f,IloscMiniskok);
  i:=0;
  repeat
    Writeln(f,Trunc(i/3):3
      ,TabSinCos[i]:4
      ,TabSinCos[i+1]:4
      ,TabSinCos[i+2]:4
      , ' === '
      ,hex(TabSinCos[i],2),' ',Hex(TabSinCos[i+1],2),' ',
      Hex(TabSinCos[i+2],2));
    i:=i+3;
  until i>=12*IloscMiniskok;
  Close(f);
end; {ZapisTabeli}
```

```
Procedure OdczytTabeli;
var f:text;
    i:integer;
    nazwa:String[8];
    k:Integer;
    z:Char;
begin
  repeat
    { $I- }
    Write('Nazwa zbioru do odczytu? ');
```

```
Assign(f,nazwa+'.MSK');
Reset(f);
i:=IOResult;
if i>0 then Writeln('Brak takiego zbioru !!!');
until i=0;
{#I+}
Readln(f,IloscMiniskok);
i:=0;
repeat
  Readln(f,k,
  TabSinCos[i],
  TabSinCos[i+1],
  TabSinCos[i+2]);
  i:=i+3;
until i>=12*IloscMiniskok;
Close(f);
end; {OdczytTabeli}
```

```
Procedure Wydruk;
var i:Integer;
    a:Real;
begin
  i:=0;
  repeat
    Writeln('k=',Trunc(i/3):3,
    ' Sin=',TabSinCos[i]:4,
    ' dod=',(TabSinCos[i+2] mod 16):4,
    ' Cos=',TabSinCos[i+1]:4,
    ' dod=',Trunc(TabSinCos[i+2]/16):4);
    i:=i+3;
  until i>=12*IloscMiniskok;
end;
```

```
Procedure GenTab;
const tst1:array[0..7] of Byte=(1,255,255,255,255,1,1,1);
      amp=2039;
```

```
var k:Integer;
    a,s,c:Real;
    znak:char;
    x,y,xd,yd:Integer;
```

```
begin
  Write('Podaj podzial skoku bazowego (1..127)
  [', IloscMiniskok,']: ');Readln(Iloscminisk
  If IloscMiniskok>127 then Iloscminiskok:=127;
  If IloscMiniskok<1 then IloscMiniskok:=1;
  if ik=IloscMiniskok then exit;
  ik:=IloscMiniskok;
  case IloscMiniskok of
```

```
1:
begin
  for k:=0 to 7 do
    TabSinCos[k]:=tst1[k];
  end;
2..127:
begin
  a:=pi/IloscMiniskok/2;
  k:=0;
  repeat
    s:=1+sin(k/3*a);
    c:=1+cos(k/3*a);
    x:=Round(amp*s)+16;
    y:=Round(amp*c)+16;
    TabSinCos[k]:=Trunc(x/16);
    TabSinCos[k+1]:=Trunc(y/16);
    xd:= x mod 16;
    yd:= y mod 16;
    TabSinCos[k+2]:=xd+yd*16;
    k:=k+3;
  until k>=12*IloscMiniskok;
end;
end;
Writeln('Podac tabele (t/n)? ');
if UpCase(ReadKey)='T' then Wydruk;
end; {Generowania tablicy sinusow i cosinusow}
```

```
VAR POCZ,KON,ADRTAB: INTEGER;
```

```
PROCEDURE MskUp(odstep,IloscSkokow: Integer);
```

```
var CZAS,KROK,LICZ1,LICZ2,TABPOCZ,TABKON,DOD: Integer;
```

```
begin
  Licz1:=RLicz1;
  Licz2:=Rlicz2;
  DOD:=PortDod;
  Czas:=Odstep;
  Krok:=IloscSkokow;
  TabKon:=Kon+3;
  TabPocz:=Pocz;
  WRITELN('Skok do przodu');
  Inline(
    $F3/           { DI }
    $E5/           { PUSH HL }
    $D5/           { PUSH DE }
    $C5/           { PUSH BC }
```

```

$2A/ ADRTAB/           {           LD HL, (ADRTAB) }
$ED/ $4B/ LICZ1/      { START: LD BC, (LICZ1) ;20 }
$23/                  {           INC HL           ;6 }
$23/                  {           INC HL           ;6 }
$23/                  {           INC HL           }
$ED/ $5B/ TABKON/    {           LD DE, (TABKON) ;20 }
$7C/                  {           LD A,H             ;4 }
$BA/                  {           CP D              ;4 }
$20/ $07/            {           JR NZ,KONT      ;7/12 }
$7D/                  {           LD A,L             ;4 }
$BB/                  {           CP E              ;4 }
$20/ $03/            {           JR NZ,KONT      ;7/12 }
$2A/ TABPOCZ/        {           LD HL, (TABPOCZ) ;20 }
$7E/                  { KONT:  LD A, (HL)         ;7 }
$ED/ $79/            {           OUT (C),A        ;11 }
$23/                  {           INC HL           ;4 }
$ED/ $4B/ LICZ2/    {           LD BC, (LICZ2)   ;20 }
$7E/                  {           LD A, (HL)         ;7 }
$ED/ $79/            {           OUT (C),A        ;11 }
$23/                  {           INC HL           }
$ED/ $4B/ DOD/       {           LD BC, (DOD)     }
$7E/                  {           LD A, (HL)         }
$ED/ $79/            {           OUT (C),A        }
$2B/                  {           DEC HL           }
$2B/                  {           DEC HL           ;4 }
$ED/ $5B/ KROK/     {           LD DE, (KROK)   ;20 }
$1B/                  {           DEC DE            ;4 }
$ED/ $53/ KROK/     {           LD (KROK), DE    ;20 }
$7B/                  {           LD A,E             ;4 }
$B2/                  {           OR D              ;4 }
$2B/ $0B/           {           JR Z,KONIEC     ;7/12 }
$ED/ $5B/ CZAS/    {           LD DE, (CZAS)   ;20 }
$1B/                  { DELAY: DEC DE           ;4 }
$7B/                  {           LD A,E             ;4 }
$B2/                  {           OR D              ;4 }
$20/ $FB/           {           JR NZ,DELAY     ;7/12 }
$1B/ $BD/           {           JR START        ;12 }
$22/ ADRTAB/        { KONIEC: LD (ADRTAB),HL }
$C1/                  {           POP BC          }
$D1/                  {           POP DE          }
$E1/                  {           POP HL          }
$FB);                {           EI             }
  { End of Inline }
end;

```

```
PROCEDURE MskDown(odstep, IloscSkokow: Integer);
```

```

var CZAS, KROK, LICZ1, LICZ2, TABPOCZ, TABKON, DOD: Integer;
begin
  Licz1:=Rlicz1;

```

```

Licz2:=RLicz2;
Dod:=PortDod;
Czas:=Odstep;
Krok:=IloscSkokow;
TabPocz:=Pocz-3;
TabKon:=kon;
Writeln ('Skok to tylu');
Inline(
$F3/          {          DI }
$E5/          {          PUSH HL }
$D5/          {          PUSH DE }
$C5/          {          PUSH BC }
$2A/ ADRTAB/  {          LD HL, (ADRTAB) }
$ED/ $4B/ LICZ1/  { START: LD BC, (LICZ1) ;20 }
$2B/          {          DEC HL ;4 }
$2B/          {          DEC HL ;4 }
$2B/          {          DEC HL }
$ED/ $5B/ TABPOCZ/ { LD DE, (TABPOCZ) ;20 }
$7C/          {          LD A,H ;4 }
$BA/          {          CP D ;4 }
$20/ $07/      {          JR NZ,KONT ;7/12 }
$7D/          {          LD A,L ;4 }
$BB/          {          CP E ;4 }
$20/ $03/      {          JR NZ,KONT ;7/12 }
$2A/ TABKON/    {          LD HL, (TABKON) ;20 }
$7E/          { KONT: LD A, (HL) ;7 }
$ED/ $79/      {          OUT (C),A ;11 }
$23/          {          INC HL ;4 }
$ED/ $4B/ LICZ2/ { LD BC, (LICZ2) ;20 }
$7E/          {          LD A, (HL) ;7 }
$ED/ $79/      {          OUT (C),A ;11 }
$23/          {          INC HL }
$ED/ $4B/ DOD/  {          LD BC, (DOD) }
$7E/          {          LD A, (HL) }
$ED/ $79/      {          OUT (C),A }
$2B/          {          DEC HL }
$2B/          {          DEC HL ;4 }
$ED/ $5B/ KROK/ { LD DE, (KROK) ;20 }
$1B/          {          DEC DE ;4 }
$ED/ $53/ KROK/ { LD (KROK), DE ;20 }
$7B/          {          LD A,E ;4 }
$B2/          {          OR D ;4 }
$2B/ $0B/      {          JR Z,KONIEC ;7/12 }
$ED/ $5B/ CZAS/ { LD DE, (CZAS) ;20 }
$1B/          { DELAY: DEC DE ;4 }
$7B/          {          LD A,E ;4 }
$B2/          {          OR D ;4 }
$20/ $FB/      {          JR NZ,DELAY ;12/7 }
$1B/ $BD/      {          JR START ;12 }
$22/ ADRTAB/    { KONIEC: LD (ADRTAB), HL }

```

```
$C1/      {      POP BC }
$D1/      {      POP DE }
$E1/      {      POP HL }
$FB);     {      EI   }
  { End of Inline }
end;
```

```
Function Word(x: Integer): Integer;
begin
  Word:=Trunc(x-Pocz);
end;
```

```
var kat1, kat2: Real;
    k: Integer;
```

```
Procedure Info;
begin
  Writeln('Liczba miniskokow = ', IloscMiniskok);
  Writeln('Odstep = ', op:6:0, ' us');
  Writeln('Liczba impulsow = ', I);
end;
```

```
begin
  ClrScr;
  op:=200000.0;
  IloscMiniskok:=2;
P: ik:=0;
  podgl:=false;
  I:=1;
  Gentab;
  Pocz:=Addr(TabSinCos[0]);
  AdrTab:=Pocz;
  Kon:=Pocz+12*IloscMiniskok-3;
  Init;
  AdrTab:=Pocz;
S:
  Write('Opoznienie po kazdym miniskoku w [us]
        [', op:6:0, '](200..200000)= '); Readln(op)
  if op<200 then op:=200;
  if op>200000.0 then op:=200000.0;
  okr:=Round((op-67)/6.5);
  Write('Ilosc miniskokow w grupie (1..32000) [', i, ']= '); Readln(I);
  IF I=0 THEN GOTO P;

  repeat
    Read(kbd, znak);
    case znak of
```

```
'/' : Mskup(okr,i);
'\ ' : Mskdown(okr,i);
'r' : goto s;
'k' : goto p;
'^' : halt;
'i' : Init;
'p' : begin podgl:=not podgl; end;
'z' : ZapisTabeli;
'o' : OdczytTabeli;
'i' : Info;
'w' : Wydruk;
end;

if podgl then
begin
Writeln(hex(adrtab,4),mem[adrtab]:6,mem[adrtab+1]:6,
        ' dod=',hex(mem[adrtab+2],2));
end
until False;

end.
```

ZAŁĄCZNIK C.

SCHEMATY UKŁADÓW STEROWANIA MINISKOKOWEGO

Do badań silnika skokowego przy pracy miniskokowej zostały skonstruowane specjalne układy sterowników przeznaczone do pracy napięciowej i prądowej. Rysunek C-1 przedstawia sterownik miniskokowy, a rysunki C-2 i C-3 wzmacniacze mocy napięciowy i prądowy.

ZALĄCZNIK D.

NAPĘDY MINISKOKOWE OPRACOWANE PRZEZ AUTORA.

W ramach powyższej pracy stworzono kolejne wersje rozwojowe sterowników miniskokowych o programowanych parametrach, dostosowanych do współpracy z mikrokomputerami. Prace te zaowocowały zgłoszeniem dwóch wynalazków do opatentowania w Urzędzie Patentowym PRL oraz kilkoma rozwiązaniami praktycznymi napędów miniskokowych, które zostały krótko przedstawione poniżej.

1. Napęd miniskokowy kamery termalnej dla potrzeb lotnictwa.

W kamerach termalnych stosuje się dwa napędy przeszukiwania: wolnego i szybkiego. Układ przeszukiwania wolnego w w.w. kamerze został wykonany w oparciu o napęd miniskokowy. Przy zastosowaniu silnika skokowego typu FA-17-4-1 prod. polskiej uzyskano parametry umożliwiające wyeliminowanie przekładni mechanicznych przy napędzie lustra przeszukiwania wolnego. Układ charakteryzuje się rozdzielczością przetwornika C/A równą 8 bitów, maksymalną częstotliwością pracy do 3000 Hz. Napęd jest sterowany z mikrokomputera kompatybilnego z IBM PC XT/AT.

2. Napędy odchyłania laserowego miernika odległości satelitarnych.

Do precyzyjnego odchyłania promienia laserowego w odległościomierzu kosmicznym, zastosowano dwa napędy miniskokowe sterowane z mikrokomputera Amstrad CPC 6128. Napęd zapewnia pozycjonowanie lustra z krokiem 0.2 mrad przy dokładności ok. 5%. Przy uzyskiwanej maksymalnej częstotliwości pracy rzędu 4000 Hz zapewnia szybkie i precyzyjne odchyłanie promienia lasera. W napędzie zastosowano silnik FA-17-4-1.

3. Synchroniczny miniskokowy napęd stołu pomiarowego do badań giroskopów światłowodowych.

Niekonwencjonalnym zastosowaniem napędu miniskokowego jest wykorzystanie go w napędzie stołu pomiarowego do badań giroskopów światłowodowych. Napęd zrealizowano poprzez takie wykonanie części obrotowej stołu, że stanowi ona wirnik silnika skokowego o magnesach trwałych. Odpowiednie zasilanie stojana tego dwupasmowego silnika o 180 biegunach zapewnia równomierną, stabilizowaną kwarcowo prędkość kątową stołu pomiarowego w zakresie ultramałych prędkości kątowych od $30^\circ/\text{h}$ do $1800^\circ/\text{h}$. Sterownik silnika oparty jest o mikrokomputer CPC 6128, wartości funkcji sterujących zapamiętane są w pamięci RAM mikrokomputera. Wartości te przetwarzane są na wartości analogowe przez dwa 16-bitowe przetworniki C/A. Ze względu na toczące się postępowanie patentowe bliższe dane nie mogą być ujawnione.

mgr inż. Jerzy K. Kowalski

Rozprawa doktorska pt.:

"Wpływ sterowania miniskokowego na charakterystyki napędu dyskretnego"

ZAŁĄCZNIK

W wyniku przeprowadzonych rozmów z Recenzentami i uwzględniając sugestię Promotora podaję ściślej sformułowaną tezę rozprawy:

"Jest możliwe opracowanie algorytmu oraz programów obliczeniowych umożliwiających symulacyjne badanie napędu ze sterowaniem miniskokowym".

SPIS TREŚCI

1.	Geneza oraz cel i zakres pracy	1
1.1	Rozwój silników skokowych.	1
1.2	Napędy skokowe.	5
1.3	Cel i zakres pracy.	9
2.	Sterowanie i konstrukcje nowoczesnych silników skokowych.	11
2.1	Praca miniskokowa silnika skokowego.	11
2.2	Konstrukcje silników skokowych.	17
2.3	Silniki skokowe a inne rodzaje silników wykonawczych.	30
3.	Modele matematyczne silników skokowych.	32
4.	Numeryczne modelowanie silników skokowych.	63
4.1	Model silnika skokowego przy pracy miniskokowej.	64
4.2	Numeryczne metody rozwiązywania układów równań różniczkowych zwyczajnych.	65
4.2.1	Metoda Eulera pierwszego rzędu.	66
4.2.2	Metoda Taylora drugiego rzędu.	67
4.2.3	Metoda Rungego-Kutty-Gilla.	69
4.2.4	Metoda Milne'a.	72
4.3	Program analizy numerycznej modelu silnika skokowego.	75
5.	Sterowanie silnika skokowego przy pracy miniskokowej.	80
5.1	Synteza funkcji sterujących do pracy miniskokowej.	80

5.2	Algorytmy sterowania miniskokowego.	85
5.3	Stánowisko do badania silników skokowych przy sterowaniu miniskokowym.	88
5.4	Badania pracy miniskokowej silnika.	90
6.	Wpływ sterowania miniskokowego na charakterystyki silnika	94
6.1	Charakterystyki mechaniczne silnika przy pracy miniskokowej.	94
6.2	Analiza wyników badań.	103
6.3	Porównanie wyników badań eksperymentalnych z symulacją komputerową.	104
7.	Podsumowanie i wnioski końcowe.	110
8.	Literatura.	113
9.	Literatura patentowa.	125
	Załącznik A: Program analizy pracy silnika skokowego przy sterowaniu miniskokowym.	128
	Załącznik B: Program sterujący silnikiem przy pracy miniskokowej.	173
	Załącznik C: Schematy układów sterowania miniskokowego.	182
	Załącznik D: Napędy miniskokowe opracowane przez autora.	186
	Spis treści.	188