

## DWUETAPOWE ALGORYTMY STEROWANIA

*Klasyczne algorytmy sterowania działają zazwyczaj w bliskim otoczeniu punktu pracy, podczas gdy jakość sterowania znacząco pogarsza się w pozostałych przypadkach. W opozycji do klasycznych algorytmów na trzech przykładach zostają zaprezentowane algorytmy dwuetapowe. Proponowane podejście pozwala w prosty sposób uzupełniać niektóre braki klasycznych algorytmów.*

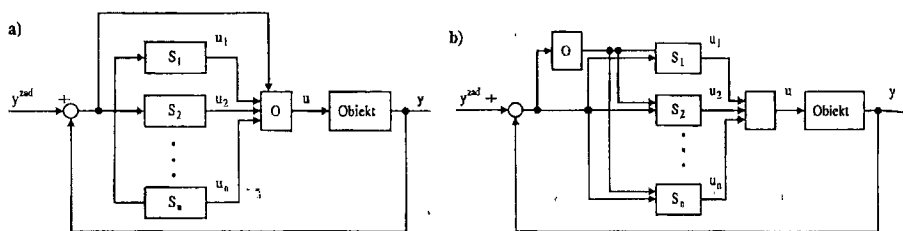
### 1 Wprowadzenie

Przez układ wieloetapowy rozumiemy układ sterujący, składający się z dwóch lub więcej etapów sterowania, z których każdy ma miejsce w innym obszarze przestrzeni stanu obiektu sterowanego. Innymi słowy, dla układu wieloetapowego określa się dwa lub więcej algorytmów sterowania; poszczególne algorytmy są uruchamiane w zależności od stanu, w którym obiekt się znajduje.

Uzasadnijmy celowość wprowadzenia tego typu układu przykładem: założmy, że celem algorytmu sterowania jest przeprowadzenie robota przez skażony teren do punktu docelowego, przy czym istnieje wydzielona trasa (robot startuje z punktu spoza trasy). W przypadku klasycznym, algorytm sterowania będzie miał na celu znalezienie najkrótszej drogi zazwyczaj ignorując istniejącą trasę. To zadanie może się okazać niewykonalne lub bardzo trudne do zrealizowania. W proponowanym przez autorów podejściu, celem algorytmu sterowania będzie sprowadzenie robota do drogi, którą łatwo już dotrzeć do punktu docelowego. W ten sposób, przy zachowaniu względnej prostoty, uzyskuje się działający we wszystkich przypadkach układ. Niestety, może się to wiązać z wydłużeniem czasu działania układu sterującego.

Strukturę algorytmu wieloetapowego można w sposób schematyczny przedstawić jak na rys. 1 a). Układy sterowania  $S_1, S_2 \dots S_n$  działają niezależnie, natomiast układ nadzorca  $O$  odpowiada za wybór odpowiedniego wyjścia  $u_n$ . Oczywiście, nie jest tutaj wymagany wybór binarny, możliwa jest np. pewna suma ważona wyjść poszczególnych układów sterowania.

Przedstawiona struktura ma zastosowanie w sytuacji, gdy dysponuje się już gotowymi układami sterującymi, których nie możemy modyfikować. Wadą tej struktury jest to, że niezależnie działające układy sterowania są stale uruchomione, co przekłada się na zajętość czasu procesora. Jeżeli istnieje więc możliwość wprowadzenia modyfikacji do układów sterujących, lepiej jest zastosować strukturę jak na



Rys. 1: Struktury układu ze sterownikiem wieloetapowym

rys. 1 b). Do układów sterujących doprowadzony jest dodatkowy sygnał  $o_n$  z zakresu  $[0,1]$ . Jeżeli sygnał ten ma wartość 0, dany układ sterowania jest wyłączony (nie przeprowadza się żadnych obliczeń), w przeciwnym przypadku sygnał  $o_n$  jest mnożony przez wyjście układu.

Podstawowym problemem przy tworzeniu wieloetapowych algorytmów sterowania jest prawidłowe zaprojektowanie algorytmu nadzorca. W tym problemie można upatrywać braku powszechności proponowanego rozwiązania w przypadku algorytmów analogowych. Aktualnie, dostępne są zarówno nowoczesne algorytmy, takie jak sieci neuronowe czy układy rozmyte (które, jak zostanie pokazane, dobrze nadają się do implementacji nadzorca), jak i szybkie maszyny cyfrowe, dzięki którym możliwa jest implementacja tych algorytmów.

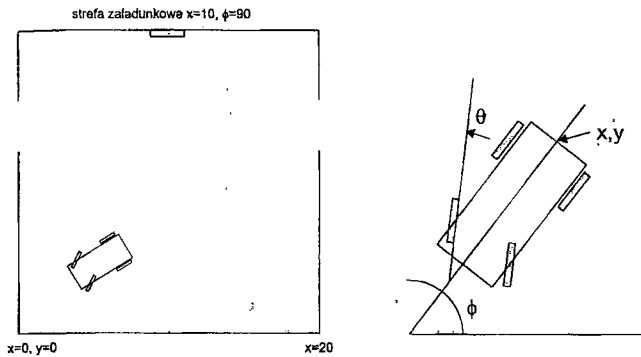
Poniżej zostaną przedstawione trzy przykłady ilustrujące możliwości zastosowań algorytmów dwuetapowych.

## 2 Parkowanie ciężarówky tyłem

Poniższy przykład ilustruje możliwość wykorzystania algorytmów dwuetapowych do kompensowania niepełnej wiedzy przy projektowaniu sterownika. Przez wiedzę rozumie się znajomość właściwego sposobu sterowania obiektem. Z sytuacją niepełnej wiedzy inżynier może mieć do czynienia w przypadku nieznanego modelu i konieczności polegania na doświadczeniu eksperta. Pozyskanie tej wiedzy może być w szczególnych przypadkach kosztowne, szczególnie gdy ma ona postać niewerbalną.

Rozpatrywany jest problem parkowania ciężarówky tyłem, który można sformułować w następujący sposób:

Położenie ciężarówky jest określone przez trzy współrzędne  $\phi$ ,  $x$  i  $y$  (rys. 2). Sterowaną wielkością jest kąt wychylenia kół  $\theta$ . Ciężarówka porusza się w jednym kierunku o stałą jednostkę odległości w każdym kroku. Kinematyka ciężarówky dana jest uproszczonymi wzorami:



Rys. 2: Strefa załadunkowa i symulowana ciężarówka

$$x(t+1) = x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)] \sin[\phi(t)] \quad (1)$$

$$y(t+1) = y(t) + \sin[\phi(t) + \theta(t)] - \sin[\theta(t)] \cos[\phi(t)] \quad (2)$$

$$\phi(t+1) = \phi(t) - \arcsin \left[ \frac{2 \sin(\theta(t))}{b} \right] \quad (3)$$

Rozwiązaniem problemu jest algorytm sterowania, który dla  $x \in [0, 20]$ ,  $\phi \in [-90 \text{ deg}, 270 \text{ deg}]$  wygeneruje takie trajektorie ( $\theta_0 \dots \theta_f$ ) dla  $\theta \in [-40 \text{ deg}, 40 \text{ deg}]$ , że  $(x_f, \phi_f) = (10, 90 \text{ deg})$ .

Jako wskaźnik jakości sterowania zastosowano sumę ważoną wartości położenia  $x$  i kąta  $\phi$  dla  $y = 30$ :

$$\epsilon_c = \arctan(\epsilon_x + 0.0267\epsilon_\phi) \quad (4)$$

gdzie  $\epsilon_x$  i  $\epsilon_\phi$  oznaczają odpowiednio bezwzględny błąd położenia i kąta. Jako zadowalające przyjęto sterowanie z błędem mniejszym niż 0.4.

Przyjęto, że odległość między ciężarówką a punktem załadunkowym (10, 90 deg) jest duża, więc wielkość  $y$  nie jest uznawana za wejście. To założenie w znakomity sposób upraszcza problem, natomiast oczywiście negatywnie wpływa na wyniki systemu w obszarze bliskim punktu załadunkowego.

Jednym z podejść do problemu parkowania ciężarówki tyłem jest sterowanie rozmyte. Przykładowa metoda projektowania sterownika rozmytego została zaproponowana w [1]. Metodologia ta polega na odpowiednim doborze reguł rozmytych nałożonych na sterownik, przy arbitralnym doborze zbiorów rozmytych reprezentujących stan i sterowanie układu. Reguły te dobiera się na podstawie wiedzy eksperckiej, którą stanowią w tym przypadku przykładowe trajektorie sterowania.

Takie trajektorie można uzyskać np. poprzez obserwację sterowania układem przez eksperta lub inny, działający układ sterujący.

Powtórzono eksperyment opisany w [1], w którym zasugerowano, że zastosowana metoda pozwala na prawidłowe wysterowanie ciężarówki dla wszystkich możliwych punktów początkowych z badanego przedziału. Autorom nie udało się powtórzyć tych wyników. Badany układ działał poprawnie jedynie w około 50% przypadków, w pozostałych doprowadzając ciężarówkę zbyt daleko od celu.

Jedną z możliwości poprawy działania sterownika jest rozszerzenie danych eksperckich. Należy mieć na uwadze, że pozyskanie wiedzy dotyczącej poprawnego sterowania może być kosztowne – w przypadku ciężarówki taki koszt jest raczej znikomy, ale może być istotny w przypadku np. rakiety typu Cruise. Drugą możliwością jest oczywiście przeprojektowanie sterownika, tj. zmiana parametrów bądź całkowita zmiana algorytmu.

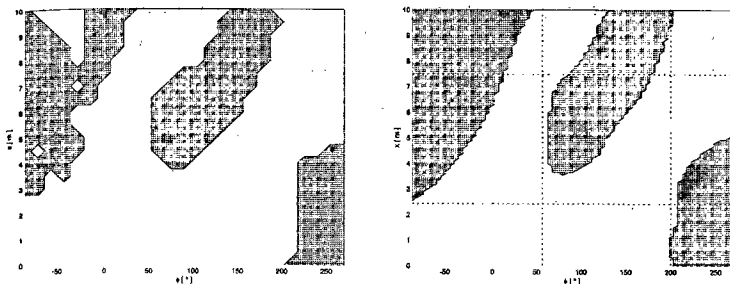
Tego typu postępowania mają następujące wady: nie uwzględniają istnienia działającego w pewnych przypadkach sterownika; mogą być kosztowne; nie dają gwarancji poprawności działania. Zamiast tego, autorzy proponują wprowadzenie dodatkowego etapu sterowania, co oznacza zaprojektowanie sterownika dwuetapowego. Dodatkowy etap sterowania będzie miał na celu sprowadzenie stanu obiektu w obszar poprawnego działania, skąd sterownik drugiego etapu (czyli istniejący algorytm) efektywnie doprowadzi układ do pożądanego celu.

Zaprojektowanie układu dwuetapowego wymaga w tym wypadku następujących kroków: 1- znalezienia obszarów niepoprawnego działania istniejącego sterownika; 2- zaprojektowanie sterownika dla tych obszarów; 3- zaprojektowanie układu nadzorcy.

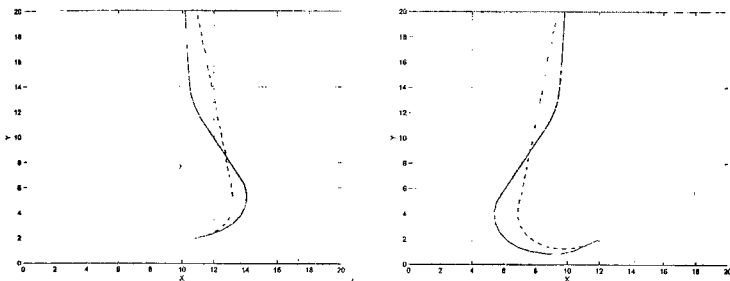
Pierwszy krok został wykonany poprzez wielokrotną symulację układu dla zmieniających się ze stałym krokiem warunków początkowych. Przypisując poszczególnym punktom wartości 0 lub 1 (w zależności od tego, czy układ zadziałał poprawnie), uzyskano „mapę” punktów startowych, dla których układ działa poprawnie. Posługując się tymi danymi zaprojektowano układ nadzorcy – wykorzystano do tego celu sieć neuronową, która miała odtwarzać kształt „mapy”. Na rys. 3 widoczna jest mapa uzyskana z symulacji oraz odtworzona przez sieć neuronową.

Te obszary, w których działanie układu było nieprawidłowe, byłyby „przejmowane” przez sterownik pierwszego etapu. Gładkie przełączanie między dwoma sterownikami było zapewnione przez strukturę sieci – zastosowano sigmoidalne funkcje aktywacji.

Układ sterujący pierwszego etapu zaprojektowano jako układ rozmyty, stosując arbitralnie dobrane reguły, w taki sposób, żeby układ ten sprowadzał obiekt w obszar poprawnego działania sterownika drugiego etapu. W wyniku połączenia obu sterowników i zastosowania omówionej sieci neuronowej jako układu nadzorcy uzyskano satysfakcjonujące wyniki (rys 4): układ działał poprawnie dla każdego punktu początkowego. Niewielkim nakładem pracy poprawiono zatem pierwotny algorytm, uzyskując w pełni sprawny układ sterowania.



Rys. 3: Obszary prawidłowego działania sterownika



Rys. 4: Porównanie działania sterownika jedno- i dwuetapowego

### 3 Poprawianie sterownika czaso- optymalnego

Przy ocenie jakości sterowania danego algorytmu istotnym czynnikiem, oprócz stabilności i dokładności, jest szybkość sterowania w sensie czasu osiągnięcia stanu ustalonego zgodnego z wartością zadaną. Jest to czynnik szczególnie istotny w przypadku działających non- stop instalacji (np. zwiększenie prędkości działania robotów w fabryce samochodów o 1% może oznaczać 1 samochód dziennie więcej) lub układów z natury wymagających dużej szybkości działania, np. systemów sterowniczych samolotów myśliwskich.

Klasyczne algorytmy bazujące na liniowej teorii układów sterowania (np. typu PID) oferują niewielką złożoność (a więc i cenę), prostotę doboru parametrów oraz dobrą dokładność sterowania, natomiast szybkość regulacji zazwyczaj nie jest duża. Regulatory tego typu są jednak bardzo popularne w przemyśle. Do przyspieszenia regulacji można stosować różnorodne algorytmy nieliniowe. Ich wadą jest jednak duża złożoność, trudny dobór struktury i parametrów oraz w ogólnym przypadku niemożność analizy.

Pewnym pomysłem na szybką regulację jest algorytm czaso-optimalny. Jest to algorytm z grubsza przybliżający ludzki sposób myślenia o szybkiej regulacji, tj. ustawianie maksymalnego możliwego sterowania a następnie korektę (wyhamowanie, także stosując maksymalne sterowanie, ale z przeciwnym znakiem) aż do osiągnięcia wartości zadanej. Teoretycznie algorytm taki zapewnia optymalny (najmniejszy) czas regulacji. W praktyce jednak zastosowanie algorytmu czaso-optimalnego wiąże się z dużymi problemami, min. wymagana jest znajomość dokładnego modelu obiektu. Zastosowanie modelu przybliżonego (tak strukturalnie jak i co do parametrów), wraz z błędami pomiaru i skończoną częstością próbkowania prowadzi do niedopuszczalnego zachowania regulatora (zostanie to pokazane niżej). Błędów tych w zasadzie nie da się wyeliminować, co w zasadzie dyskwalifikuje ten algorytm.

Rozpatrzmy serwomechanizm pozycyjny, opisany znormalizowanymi równaniami:

$$\frac{dy}{d\tau} + y = u; \quad \frac{dx}{d\tau} = y; \quad |u| \leq 1 \quad (5)$$

Normalizacja oznacza tutaj, że dla potrzeb regulacji skaluje się odpowiednie sygnały w taki sposób, aby uprościć stałe występujące w równaniach. W rozpatrywanym przypadku stała czasowa silnika została dobrana jako duża, co nie jest widoczne w znormalizowanych równaniach.

Sterowanie czaso-optimalne wymaga oddziaływania sygnałem  $u$  o maksymalnej dopuszczalnej amplitudzie (w przypadku układu liniowego). Trajektorie fazowe układu dla sterowania  $u \in [-1, 1]$  opisują się wzorami

$$x - x_0 = \begin{cases} (y_0 - y) - \ln\left(\frac{1-y}{1-y_0}\right); & u = +1 \\ (y_0 - y) + \ln\left(\frac{1+y}{1+y_0}\right); & u = -1 \end{cases} \quad (6)$$

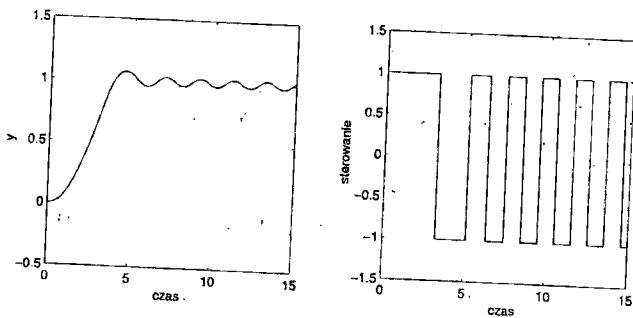
Z rodziny trajektorii fazowych układu wybierane są takie, które przechodzą przez  $(x, y) = (0, 0)$ , przez co uzyskuje się równanie opisujące krzywą przełączeń:

$$\varphi = [|y| - \ln(1 + |y|)] \operatorname{sgn}(y) \quad (7)$$

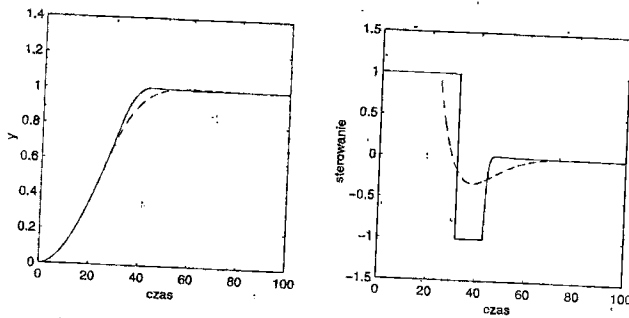
Następnie, przyjmuje się następujące prawo sterowania:

$$\begin{aligned} \varphi - x > 0 &\rightarrow u = +1 \\ \varphi - x < 0 &\rightarrow u = -1 \end{aligned} \quad (8)$$

Działanie układu zostało przedstawione na rysunku 5. Po osiągnięciu przez sterowany obiekt zadanej wartości kąta  $\alpha$  występują niegasnące oscylacje (nie widoczne na wykresie) o małej amplitudzie, spowodowane przełączaniem układu sterowania. Takie zachowanie układu jest w ogólnym przypadku niekorzystne i niepożądane - a jednocześnie niemożliwe do wyeliminowania w strukturze algorytmu czaso-optimalnego. W celu eliminacji niepożądanych zachowań podjęta zostanie próba zastosowania algorytmu dwuetapowego.



Rys. 5: Odpowiedź skokowa układu ze sterownikiem czasooptymalnym



Rys. 6: Działanie układu dwustopniowego (l. ciągła) i liniowego (l. przerywana),

Zastosowano strukturę układu odpowiednio do rys. 1 b). Jako dodatkowy układ sterujący zastosowano regulator w układzie PDI, sygnał wyjściowy regulatora jest więc wypadkową sygnałów wyjściowych układów czaso-optymalnego i PDI. O udziale każdego z tych układów w sterowaniu decyduje rozmyty układ nadzorcy. Sygnał wybierający z nadzorcy jest uzależniony od uchybu oraz prędkości kątowej obiektu- jeżeli uchyb i prędkość są „małe”, wtedy wybierany jest sygnał ze sterownika PDI, w przeciwnym przypadku wybierany jest układ czaso-optymalny.

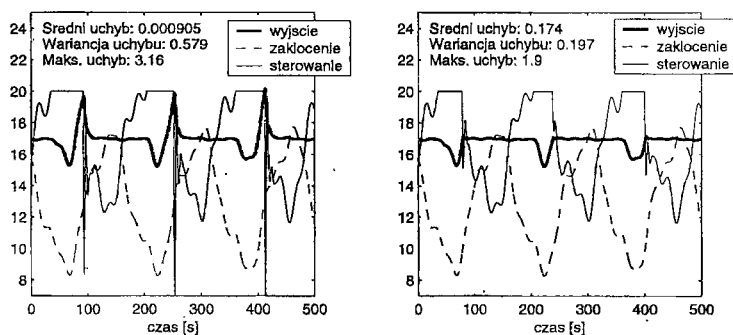
Wartości progowe w układzie nadzorcy zostały dobrane eksperymentalnie, podobnie jak parametry sterownika PDI. Zastosowanie układu rozmytego zamiast prostego przełącznika zerojedynkowego jest podyktowane chęcią wyeliminowania gwałtownych zmian wartości sterowania.

Przykładowe wyniki działania układu zostały przedstawione na rysunku 6. Observed się przyrost szybkości regulacji w stosunku do regulatora liniowego o około 20%, przy niewielkim przeregulowaniu i braku uchybu. Poprzez połączenie regulacji liniowej i czasoptymalnej w układ dwuetapowy udało się więc uzyskać szybki, stabilny regulator.

#### 4 Sterowanie w otoczeniu ograniczeń sterowania

Rozważmy układ sterowania z liniowym obiektem i ograniczeniami sterowania. Jako regulator zastosowany został typowy układ PID, z dobranymi eksperymentalnie nastawami.

Układ pracuje w trybie pracy ciągłej i jest stale zakłócany sygnałem o dużej amplitudzie. Przykładowy przebieg został pokazany na rys. 7. Regulator poprawnie kompensuje zakłócenie, jednak w momencie osiągnięcia przez wartość sygnału sterowania wartości górnego ograniczenia zakłócenie nie może być w pełni kompensowane, przez co powstaje znaczący uchyb. Po zmniejszeniu się wartości zakłócenia sterowanie pozostaje na ograniczeniu jeszcze przez pewien czas, co powoduje dodatkowy niepożądany uchyb wartości regulowanej.



Rys. 7: Układ liniowy z ograniczeniem sterowania, bez i z pętlą anti-wind-up

W przypadku nasycenia wartości sterowania następuje „przerwanie” pętli sprzężenia zwrotnego, ponieważ wartość sterowania może pozostać na ograniczeniu nawet w przypadku zmiany wartości wyjścia- co oznacza że wartość sumy w bloku całkującym regulatora może osiągnąć bardzo duże wartości. Po zmniejszeniu się błędu, wartość sumy może być na tyle duża, że zejście do normalnych wartości może zająć istotnie wiele czasu. Ten efekt nazywa się *nakręcaniem* (ang. *windup*) i usuwa się go wprowadzając dodatkową pętlę (tzw. *anti-wind-up*) do algorytmu regulacji. W przypadku algorytmu dyskretnego taką pętlę można alternatywnie wyrazić w postaci prostego warunku `if..else`.



Na rysunku 7 przedstawiono działanie pętli regulacji z dodatkowym warunkiem *anti-wind-up*. Błąd wynikający z narastania wartości sumy w integratorze został zniwelowany, pozostaje jednak znaczący uchyb w momencie osiągnięcia przez zakłócenie dużych wartości.

Uchyb ten jest oczywiście niemożliwy do usunięcia. Możliwe jest jednak polepszenie pewnych wskaźników jakości sterowania, np. średniego błędu. Zmniejszenie średniego błędu może być istotne np. w przypadku automatycznego pilota- jeżeli średnia odchyłka od planowanego kursu jest bliska zeru, punkt docelowy zostanie osiągnięty niezależnie od bezwzględnych wartości uchybu.

Polepszenie takie jest możliwe przez „wyprzedzanie” wystąpienia uchybu poprzez wprowadzenie uchybu o przeciwnym znaku i podobnej amplitudzie.

Wprowadźmy do układu dodatkowy blok regulacji nadrzędnej (pierwszy „stopień” sterowania). Załóżmy czas próbkowania bloku nadrzędnego jako dużo większy od czasu próbkowania bloku bezpośredniego. Wynika to z tego, że w pojedynczym kroku regulacji nadrzędnej w obiekcie powinny zaniknąć stany przejściowe i obiekt może zostać potraktowany jako statyczny o wzmocnieniu proporcjonalnym  $K_p$ . Przyjęto następujące przybliżenie:

$$u_p(k) = \frac{1}{K_p} y^{zad}(k) - d_p(k) \quad (9)$$

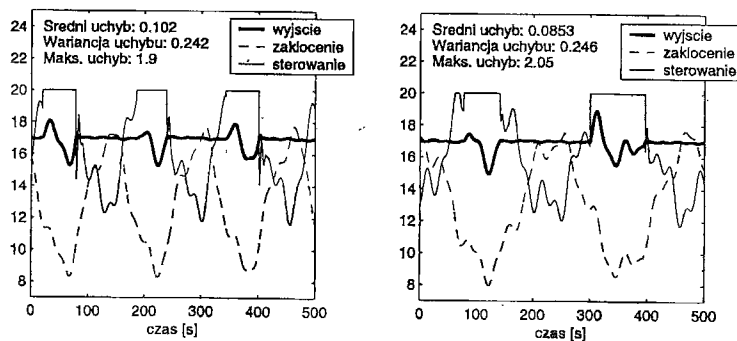
Taka wielkość sterowania jest wymagana do prawidłowej kompensacji zakłócenia w chwili  $k$ -tej. Z tej zależności, mając gotową predykcję wartości zakłócenia  $d_p(k)$  obliczany jest ciąg predykowanych wartości sterowania. Z tych wartości otrzymuje się czasy  $t_r$  i  $t_z$ , odpowiadające najbliższej chwili czasowej, w której układ wejdzie na zakłócenia oraz chwili, w której te zakłócenia opuści. Dodatkowo, zapamiętuje się wartość  $\Delta u$ , określającą maksymalną różnicę między sterowaniem a wartością zadaną, która przystaje do maksymalnej wartości uchybu, jaki wytworzy się w wyniku działania zakłócenia.

Na podstawie tych wielkości na wyjście układu nadrzędnego wystawiana jest wartość poprawki:

$$\Delta y^{zad} = \begin{cases} \Delta u, & \text{gdy } t_r \leq t_z - t_r \\ 0 & \text{w p.p.} \end{cases} \quad (10)$$

Dla tak sformułowanego prostego algorytmu wykonano szereg eksperymentów symulacyjnych. Przyjęto czas próbkowania bloku nadrzędnego 10 razy dłuższy od czasu próbkowania regulatora bezpośredniego, który miał postać algorytmu PID z pętlą *anti-wind-up*. Jako zakłócenie przyjęto quasi- okresowy sygnały o różnych amplitudach i częstotliwościach. Do celów predykcji zastosowano model AR o raczej dużej złożoności.

Przykładowy wynik symulacji znajduje się rys. 8. Dzięki zastosowaniu sterownika dwuetapowego udało się zredukować średnią wartość uchybu, co niestety pociągnęło za sobą zwiększenie błędu średniokwadratowego.



Rys. 8: Działanie algorytmu dwuetapowego dla różnych sygnałów zakłócenia

## 5 Podsumowanie

Dwuetapowe algorytmy sterowania są wygodnym i prostym sposobem omijania problemów związanych z różnorodnymi ograniczeniami klasycznych algorytmów sterowania, takich jak nieliniowości, błędy modelowania i inne.

Przykładowe zastosowania algorytmów dwuetapowych mogą dotyczyć sytuacji, w których: dysponuje się algorytmem klasycznym o odpowiednio dobranych parametrach; algorytm klasyczny działa poprawnie jedynie w pewnym zakresie zmienności zmiennych stanu; algorytm klasyczny posiada cechy zmniejszające jakość sterowania w pewnym obszarze zakresu zmienności zmiennych stanu.

Stosowanie algorytmów dwuetapowych w zasadzie wymaga stosowania sterowników cyfrowych. Ze względu na powszechną dostępność takich urządzeń oraz ich malejące ceny nie stanowi to przeszkody. Pewnym problemem może być jednak stosunkowo duży nakład obliczeń związany z układem nadzorczy, który może wymagać stosowania urządzeń o stosunkowo dużej wydajności.

## Literatura

- [1] L. Wang, J. Mendel: "Generating Fuzzy Rules by Learning from Examples"
- [2] A. Piegat: „Modelowanie i sterowanie rozmyte”, Oficyna Wydawnicza EXIT, Warszawa, 1999
- [3] S. Osowski: „Sieci neuronowe w ujęciu algorytmicznym”, *WNT 1996*
- [4] P. Tatjewski: „Sterowanie zaawansowane obiektów przemysłowych- struktury i algorytmy”, Oficyna Wydawnicza EXIT, Warszawa, 2002