

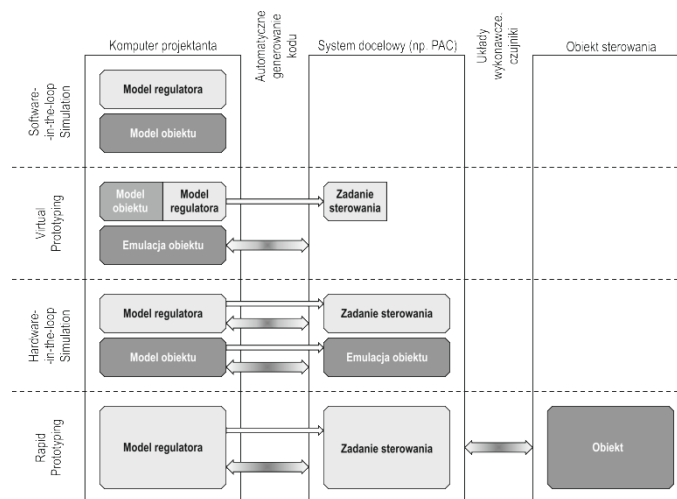
## Prototypowanie algorytmów sterowania cyfrowego serwonapędu silnika PMSM

**Streszczenie.** W artykule przedstawiono wyniki prac nad jednym ze stanowisk badawczych opracowanym w ramach projektu badawczego MNiSW nr NN502 336936 „Opracowanie konstrukcji i badania doświadczalne mechatronicznego obrabiarkowego zespołu posuwowego z napędem sterowanym inteligentnym modułowym układem wykonawczym”. Stanowisko to z użyciem komercyjnie dostępnego serwonapędu cyfrowego, dzięki implementacji cyklicznej komunikacji z nadrzędnym systemem sterowania opartej na technologii deterministycznego Ethernetu przemysłowego (w tym przypadku protokołu Ethernet Powerlink) pozwala na szybkie testowanie nowych koncepcji algorytmów regulacji prędkości i/lub położenia silnika synchronicznego ze wzbudzeniem od magnesów trwałych. Są to badania wstępne, gdyż dla celów opracowania nowych algorytmów z technologicznego punktu widzenia konieczne jest zbudowanie całkowicie nowej konstrukcji. W artykule dokonano podziału na prototypowanie z użyciem narzędzi naukowych oraz z zastosowaniem docelowego sprzętu (stosowanego w przemyśle). Pomimo bliskości metodologicznej obu podejść istnieje pomiędzy nimi istotna z punktu widzenia czasu wdrażania nowych algorytmów rozbieżność.

**Słowa kluczowe:** silniki synchroniczne, PMSM, szybkie prototypowanie

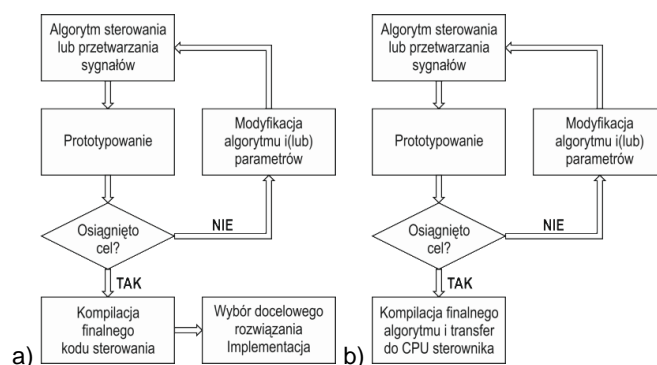
### Wprowadzenie

Pod pojęciem szybkiego prototypowania algorytmów regulacji automatycznej współcześnie rozumiemy szereg technik (od badań symulacyjnych, poprzez wirtualne prototypowanie, badania *hardware-in-the-loop*, po kompilację kodu w urządzeniu sterującym czasu rzeczywistego, patrz Rys. 1), prowadzących do wytworzenia nowego produktu – nowego kontrolera urządzenia rzeczywistego (fizycznego) [5].



Rys. 1. Etapy szybkiego prototypowania algorytmów sterowania.

W związku z zagadnieniem szybkiego prototypowania algorytmów regulacji współcześnie promowane są dwie ścieżki postępowania. Pierwsza z nich zgodna z podejściem naukowym kończy procedurę szybkiego prototypowania na etapie implementacji kodu algorytmu w urządzeniu specjalistycznym – karcie z systemem czasu rzeczywistego, realizującym funkcje sterujące. Aby wykonać nowy produkt konieczne jest przyniesienie tak opracowanego kodu do mikroprocesora, mikrokontrolera, układu reprogramowalnego lub do układu typu ASIC, produkowanych następnie w wielu egzemplarzach. Oznacza to jednak, iż opracowywany algorytm nie jest testowany w docelowych warunkach działania, chociaż bardzo zbliżonych (Rys. 2a).



Rys. 2. Podejścia do szybkiego prototypowania algorytmów sterowania.

Podejście drugie, promowane przez środowiska przemysłowe (Rys. 2b), współpracujące z uczelniami (jak również przez naukowców, mających doświadczenie w pracach dla przemysłu) stanowi o tym, że prototypowane algorytmy powinny być już na etapie projektu, rozwoju testowane w docelowym rozwiązaniu systemu sterowania. Jest to daleko trudniejsze w praktycznej realizacji, gdyż wymaga od producenta systemu sterowania odpowiedniego pakietu narzędzi, umożliwiających automatyczne generowanie kodu programu na podstawie modeli takich programów modelowania i symulacji jak np. Matlab/Simulink firmy Mathworks. Jest ich coraz więcej, zaś szczegółowo zagadnienie to omówiono w pracy [7].

### Napędy posuwu obrabiarek CNC

Współcześnie technologia silników synchronicznych ze wzbudzeniem od magnesów trwałych jak również synchronicznych silników liniowych wypiera z zastosowań w napędach posuwu obrabiarek sterowanych numerycznie silniki asynchroniczne [1;3]. Podstawowym interfejsem sprzężenia zwrotnego od silnika do układu regulacji jest sprzężenie położeniowe absolutne (lub rzadziej relatywne), zwykle bardzo wysokiej rozdzielczości.

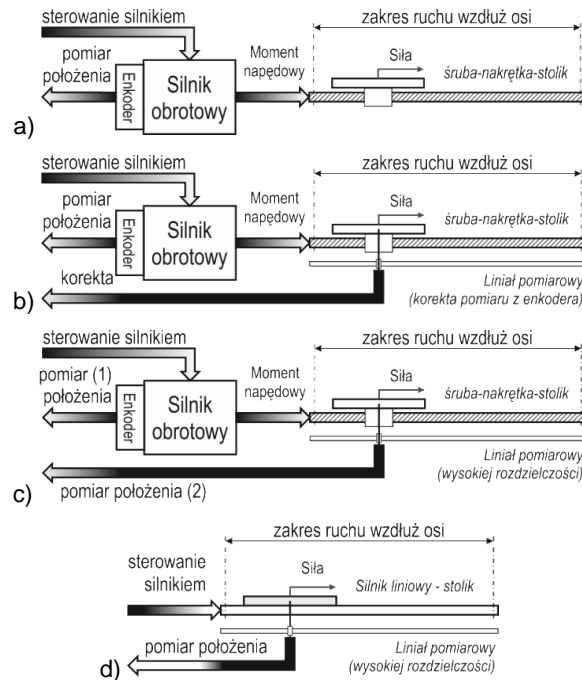
Spotykane w przemyśle układy sterowania napędami w osiach posuwu obrabiarek można podzielić (z punktu widzenia zastosowanych elementów pomiaru położenia) na: otwarte (Rys. 3a), półzamknięte (Rys. 3b) i zamknięte (Rys. 3c,d).

W przypadku układów otwartych zastosowanie mają silniki obrotowe z przeniesieniem napędu za pomocą układu śrubowego kulowo-tocznego. Układ tego typu, przedstawiony na Rys. 3a, nosi nazwę otwartego z powodu pomiaru położenia silnika napędzającego układ mechaniczny, znajdującego się za silnikiem. Na układ ten składają się: przeniesienie napędu (układ *sprzęgło-śruba-nakrętka*) oraz poruszany element (np. stolik frezarki).

Układ półzamknięty (Rys. 3b) to rozwiązanie z silnikiem obrotowym, z przeniesieniem napędu identycznym jak w przypadku układu otwartego. Istotną różnicą jest tutaj dodatkowo zamontowany układ cyklicznie kontrolujący poprawność odwzorowania

## VII Lubuska Konferencja Naukowo-Techniczna – i-MITEL 2012

położenia poruszanego układu mechanicznego przez enkoder zamontowany (tak jak w układzie otwartym) na wałku silnika. Ten układ to liniał pomiarowy, zwykle niskiej rozdzielczości, celem uniknięcia znaczącego wzrostu kosztów spowodowanego powielaniem układów pomiarowych w systemie sterowania.



Rys. 3. Typy układów sterowania napędem w osiach posuwu frezarek.

W przypadku układu zamkniętego mamy do czynienia z zastosowaniem silnika liniowego (Rys. 3d), zwykle ze wzbudzeniem od magnesów trwałych, synchronicznego, prądu zmiennego wraz z liniałem pomiarowym absolutnym bardzo wysokiej rozdzielczości i dokładności.

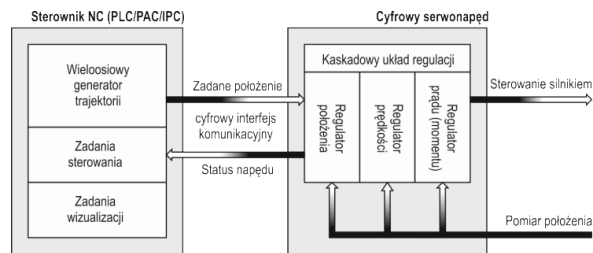
Przy odpowiednim oprogramowaniu systemowym możliwe jest również zastosowanie architektury układu zamkniętego dla silnika obrotowego (Rys. 3c).

### Kaskadowy układ regulacji cyfrowego serwonapędu silnika PMSM

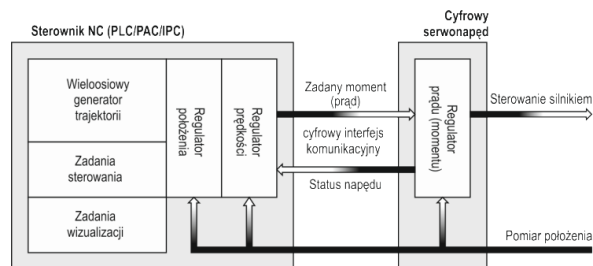
Typową współcześnie stosowaną (w przeciwieństwie do rozwiązań historycznych [2]) jest struktura kaskadowego układu regulacji położenie-prędkość-moment (prąd) cyfrowego serwonapędu, przestawiona na Rys. 4.

Istotnym elementem jest tutaj cyfrowy deterministyczny interfejs komunikacyjny pomiędzy sterownikiem nadrzędnym (zawierającym generator trajektorii) a procesorem wewnętrznym serwonapędu, realizującym funkcje kaskadowego układu regulacji.

Coraz większa liczba producentów sprzętu automatyki z uwagi na rosnące moce obliczeniowe procesorów stosowanych ich systemach sterowania jak również coraz krótsze (sięgające 100 mikrosekund) czasy cyklu w ramach deterministycznych przemysłowych protokołów komunikacyjnych zmienia architekturę układu regulacji na taką, w której pętle regulacji położenia i prędkości realizowane są sposób programowy zaś sam serwonapęd zyskuje funkcję falownika z regulatorem prądu. Architekturę taką przedstawiono na Rys. 5.



Rys. 4. Kaskadowy układ regulacji położenie-prędkość-moment (prąd).



Rys. 5. Programowy układ regulacji położenie-prędkość – regulator prądu w module energoelektronicznym.

### Programowanie cyfrowego serwonapędu – PLCopen Motion Control

W ramach standardu PLCopen Motion Control zdefiniowano stany, określające działanie cyfrowego serwonapędu. Stany te w graficzny sposób odzwierciedlają aktualny stan układu napędowego – ruch, zatrzymanie, oczekiwanie na komendę ruchu z nadrzędnego systemu sterowania, uruchomienie i zatrzymanie ruchu osi napędowych.

Poszczególne stany odzwierciedlają sytuację, w jakiej znajduje się dana oś ruchu: *Disabled* – regulator napędu jest wyłączony; *Standstill* – napęd nie wykonuje ruchu – oczekuje na akcję, oś silnika utrzymywana jest w aktualnej pozycji, *Homing* – bazowanie – w przypadku enkoderów absolutnych nie wywołuje procedury ruchu; *ErrorStop* – stan wstrzymania po wystąpieniu błędu, konieczne jest potwierdzenie wszystkich błędów; *Stopping* – wstrzymywanie aktualnie wykonywanego ruchu; *Discrete Motion* – ruch na zadane położenie, ruch ze zdefiniowanym punktem końcowym; *Continuous Motion* – ruch bez określonego położenia zadanego, ruch bez punktu końcowego.

Przejęcie pomiędzy stanami odbywa się poprzez wywołanie odpowiednich funkcji, takich jak np. *MC\_MoveAbsolute* (ruch absolutny), czy *MC\_MoveAdditive* (ruch relatywny w określonym kierunku).

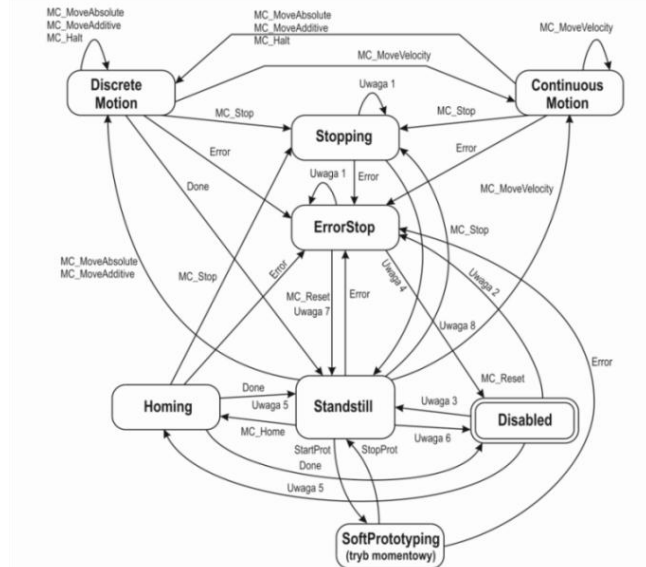
Na Rys. 6 pojawia się kilka uwag: (1) w stanach *Stopping*, *Errorstop*, *Disabled* oraz *Homing* nie można wywołać żadnej z procedur ruchu. W przypadku osi rzeczywistej nie możliwe jest wykonywanie jakichkolwiek ruchów (wywoływanych z poziomu stanu *Standstill*), dopóki oś nie zostanie zbazowana; (2) *MC\_Power.Enable = TRUE* z wystąpieniem błędu osi; (3) *MC\_Power.Enable = TRUE* oraz brak błędu osi; (4) *MC\_Stop.Done = TRUE* i *MC\_Stop.Execute = FALSE*; (5) przy wystąpieniu *MC\_Home.Done = TRUE* (zakończenie bazowania osi) oś powraca do stanu, z którego było wywołane bazowanie: *Standstill* lub *Disabled*; (6) *Power.Enable = FALSE*; (7) *MC\_Reset.Done = TRUE* i *MC\_Power.Enable = TRUE* oraz brak błędu osi; (8) *MC\_Reset.Done = TRUE* i *MC\_Powet.Enable = FALSE* lub wystąpienie błędu osi.

### Podsystem sterowania napędami zespołów posuwu obrabiarki

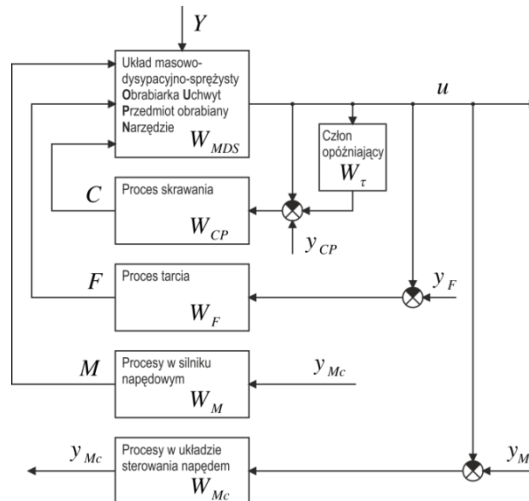
Na Rys. 7 przedstawiono jednowymiarowy przypadek dynamicznego układu OUPN [4], z uwzględnieniem procesów zachodzących w silniku napędowym oraz w systemie

**VII Lubuska Konferencja Naukowo-Techniczna – i-MITEL 2012**

sterowania napędem posuwu. Na potrzeby niniejszego artykułu pominięto analizę procesów związanych ze skrawaniem  $C, W_{CP}, y_{CP}, W_{\tau}$  oraz procesy związane z tarciami w elementach konstrukcyjnych obrabiarki  $F, W_F, y_F$ . Symbol sumatora na kolejnych rysunkach  $\otimes$  w polu zaciemnionym oznacza przyjęcie wartości sygnału ze znakiem minus.



Rys. 6. Modyfikacja diagramu normy PLCopen Motion Control o stan, w którym dokonywane jest programowe prototypowanie (SoftPrototyping).

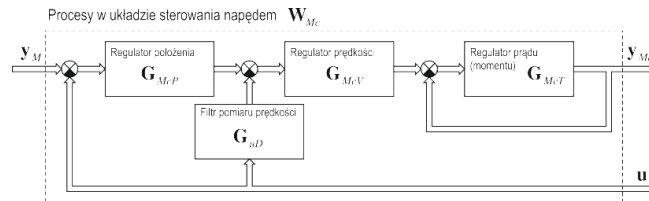


Rys. 7. Dynamiczny układ OUPN z wyszczególnieniem podsystemu sterowania napędem.

Równanie, opisujące działanie układu z Rys. 7 (z pominięciem operatora  $s$ )

$$u = \left\{ 1 + W_{MDS} \left[ W_{CP} (1 - W_\tau) + W_F + W_M W_{Mc} \right] \right\}^{-1} W_{MDS} \begin{bmatrix} W_{CP} & 0 & 0 \\ 0 & W_F & 0 \\ 0 & 0 & W_M W_{Mc} \end{bmatrix} \begin{bmatrix} y_{CP} \\ y_F \\ y_M \end{bmatrix} + \left\{ 1 + W_{MDS} \left[ W_{CP} (1 - W_\tau) + W_F + W_M W_{Mc} \right] \right\}^{-1} Y \quad (1)$$

Na Rys. 8 przedstawiono typową kaskadową strukturę sterowania cyfrowych serwonapędów zespołu posuwowego obrabiarki.



Rys. 8. Typowa struktura kaskadowa serwonapędu obrabiarki

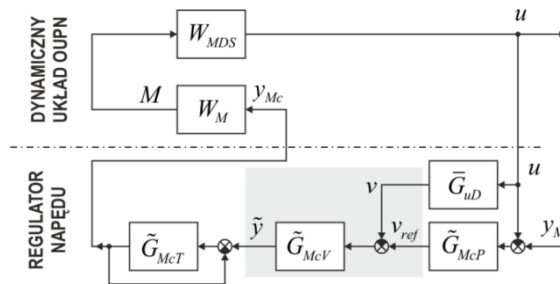
$$y_{Mc} = (\mathbf{I} + \mathbf{G}_{McT})^{-1} \mathbf{G}_{McT} \mathbf{G}_{McV} \begin{bmatrix} \mathbf{G}_{McP} & -\mathbf{G}_{McP} - \mathbf{G}_{uD} \end{bmatrix} \begin{bmatrix} y_M \\ u \end{bmatrix} \quad (2)$$

Na potrzeby niniejszej pracy założono, że regulator prądu posiada nieznaną strukturę i parametry, stanowiąc tym samym element obiektu sterowania, zaś sygnałem wymuszającym jest w tym przypadku sygnał zadanego momentu, lub inaczej zadanej wartości prądu w osi  $q$ , zgodnie z przyjętą strategią regulacji prądu [3].

#### Odporna regulacja prędkości w serwonapędzie silnika PMSM

Klasyczna jednopętlowa struktura układu regulacji, przedstawiona na Rys. 9 opisana jest następującym równaniem (3):

$$v_{PID}(s) = \frac{W_{M-MDS}(s) \tilde{G}_{McV}(s) \bar{G}_{uD}(s)}{1 + W_{M-MDS}(s) \tilde{G}_{McV}(s) \bar{G}_{uD}(s)} v_{ref}(s) \quad (3)$$



Rys. 9. Prosty regulator prędkości.

gdzie  $y_M(s)$  – wartość zadana układu regulacji (tutaj prędkość),  $W_{M-MDS}(s)$  – zastępcza transmitancja sterowanego obiektu w torze prąd zadany w osi  $q \rightarrow \omega$  (prędkość obrotowa silnika, obciążonego konstrukcją zespołu posuwu),  $\tilde{G}_{McV}(s)$  – transmitancja regulatora prędkości,  $v(s)$  – sygnał wyjściowy (prędkość) obiektu,  $\bar{G}_{uD}(s)$  – znana, założona funkcja przekształcająca wartość położenia  $u(s)$  na wartość prędkości. Cyklicznie przekazywana wartość prądu zadanego w układzie z Rys. 5

**VII Lubuska Konferencja Naukowo-Techniczna – i-MITEL 2012**

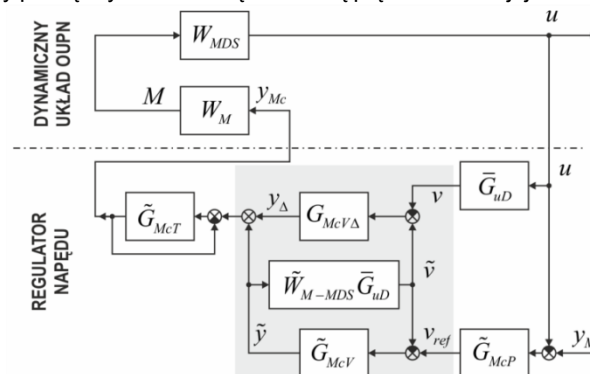
wynosi  $\tilde{y}$ . Ważne jest, aby w tym miejscu zaznaczyć, iż sterowany obiekt  $W_{M-MDS}(s)$  różni się od swojego nominalnego modelu  $\tilde{W}_{M-MDS}(s)$ , na podstawie którego zaprojektowano regulator  $\tilde{G}_{McV}(s)$ , o nieznane, lecz ograniczone w zakresie częstotliwości pracy układu perturbacje  $\Delta(s)$ :

$$W_{M-MDS}(s) = [1 + \Delta(s)] \tilde{W}_{M-MDS}(s), \quad \Delta(s)|_{s=j\Omega} < 1, \quad \Omega \leq \Omega_{gr} \quad (4)$$

Uwzględniając zależność (4) w równaniu (3), oraz zakładając pełną znajomość modelu ( $\Delta(s) = 0$ ) można wprowadzić pojęcie nominalnej pętli układu regulacji, danej równaniem:

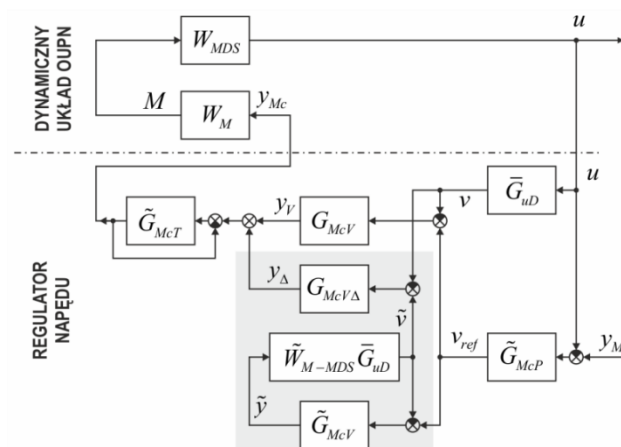
$$\tilde{v}(s) = \frac{\tilde{W}_{M-MDS}(s) \tilde{G}_{McV}(s) \bar{G}_{uD}(s)}{1 + \tilde{W}_{M-MDS}(s) \tilde{G}_{McV}(s) \bar{G}_{uD}(s)} v_{ref}(s) \quad (5)$$

Bazując na zależności (5), na rysunku kolejnym przedstawiono strukturę odpornego regulatora typu model-following [6;8;9] prędkości, możliwą do zaimplementowania w cyfrowym serwonapędzie (Rys. 5), działającym w trybie cyklicznego zadawania prądu w trybie momentowym. Wartość tego prądu wynosi w tym przypadku  $\tilde{y} + y_{\Delta}$ , i wyznaczana jest jako złożenie dwóch składowych sterowania: głównego sygnału  $\tilde{y}$ , wyznaczanego dla nominalnego modelu sterowanego procesu oraz korekty sygnału prądu  $y_{\Delta}$ , wyznaczonej na podstawie różnicy pomiędzy oczekiwaną wartością prędkości  $\tilde{v}$  a jej wartością rzeczywistą.



Rys. 10. Regulator MFC prędkości w serwonapędzie obrabiarki

Na Rys. 11 przedstawiono całkowicie nową strukturę układu regulacji prędkości (w tym konkretnym rozwiązaniu stanowiącym zaawansowany mechanizm korekcyjny prądu sterującego silnikiem, który istotnie poprawia jakość pracy podczas ruchów osi obrabiarki z małymi prędkościami, gdzie istotne znaczenie mają złożone zjawiska związane z tarcieniem w elemencie wykonawczym – silniku, ale również w samej konstrukcji osi posuwu obrabiarki). Należy tutaj zwrócić uwagę, iż wartość prędkości nominalnej  $\tilde{v}$  w układzie z Rys. 11 wyliczana jest na podstawie równania (5), zaś regulator  $G_{McV\Delta}$  „pomaga” regulatorowi głównemu  $G_{McV}$  (którego użytkownik nie musi w żaden sposób modyfikować) osiągnąć zadaną prędkość ruchu  $v_{ref}$ , pomimo różnic, jakie występują pomiędzy modelem a sterowanym obiektem.



Rys. 11. Pomocniczy regulator prędkości typu model-following

### Podsumowanie

W artykule zaprezentowano stanowisko badawcze wiążące podejście szybkiego prototypowania w docelowym przemysłowym systemie sterowania z nową programową architekturą kaskadowego układu regulacji cyfrowego serwonapędu silnika synchronicznego (PMSM) oraz wskazano sposób implementacji w ramach diagramu stanów normy PLCopen Motion Control.

Przedstawiono i omówiono trzy (w tym jedną klasyczną oraz dwie odporne, bazujące na informacji o nominalnym modelu sterowanego procesu) struktury układów regulacji prędkości. Wskazano na możliwości ich praktycznej weryfikacji w opracowanym środowisku sprzętowo – programowym, stanowiącym rozwinięcie funkcjonalności opisanej w normie PLCopen Motion Control.

### Literatura

1. Gieras J.F., *Advancements in electric machines*, Springer, 2009.
2. Kosmol J., *Serwonapędy obrabiarek sterowanych numerycznie*. Wydawnictwa Naukowo-Techniczne, 1998.
3. Krishnan R., *Permanent Magnet Synchronous and Brushless DC Motor Drives*. Boca Raton: CRC Press, Taylor and Francis Group, LLC, 2010.
4. Marchelek K., *Dynamika obrabiarek*. Warszawa: Wydawnictwa Naukowo-Techniczne, 1991.
5. Petko M., *Wybrane metody projektowania mechatronicznego*. Kraków, Radom: Wydawnictwo Naukowe Instytutu Technologii Eksploatacji – PIB, 2008.
6. Pietruszewicz K., Dworak P., Misztal H., *Szybkie prototypowanie struktur układów regulacji temperatury w sterownikach programowalnych*, *Przegląd Elektrotechniczny*, Nr 5, str. 304 – 306, 2010.
7. Pietruszewicz K., Urbański Ł., *Balancing PLCs, PACs and IPCs. What controller fits your application?*, *Control Engineering*, Nr 1, str. 28 – 32, 2011.
8. Skoczowski S., *Odporny układ regulacji z wykorzystaniem modelu obiektu*, *Pomiary Automatyka Kontrola*, Nr 9, str. 2 – 4, 1999.
9. Skoczowski S., Domek S., Pietruszewicz K., Broel-Plater B., *A method for improving the robustness of PID control*, *IEEE Transactions on Industrial Electronics*, Vol. 52, Nr 6, str. 1669 – 1676, 2005.

**Autor:** dr inż. Krzysztof Pietruszewicz; Wydział Elektryczny, Katedra Automatyki Przemysłowej i Robotyki, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie, ul. Sikorskiego 37, 70-313 Szczecin, e-mail: krzysztof.pietruszewicz@zut.edu.pl