

mgr inż. Krzysztof Skura, dr inż. Zbigniew Smalec
Instytut Technologii Maszyn i Automatykacji PWR
mgr inż. Tadeusz Nowak, Microtech Int. Ltd W-w

Stanowisko do badań symulacyjnych układów regulacji z wykorzystaniem techniki HWILS

Streszczenie: Wskazano na znaczenie stosowania metod RP i HWILS w badaniach projektowanych układów sterowania i regulacji. Omówiono zbudowane stanowisko do badań symulacyjnych układów regulacji procesów ciągłych z wykorzystaniem techniki HWILS oraz przykład jego praktycznego wykorzystania.

A Stand for simulation research of control system using HWILS technology

Summary: *Special methods of RP and HWILS and their significance in design of control systems are indicated in this article. In particular design and work of the simulation stand for control systems of continuous processes are described. The practical aspect of simulation stand and its development is explained.*

1. WSTĘP

Układy sterowania i regulacji, ze względu na często występujące w nich zjawiska nieliniowe takie jak: tarcie, luzy, zmiana masy układu itp., które w istotnym stopniu wpływają na zachowanie się sterowanych obiektów, powinny być wszechstronnie przebadane. Zastosowanie nowoczesnej techniki obliczeniowej umożliwia szybkie testowanie układów sterowania i regulacji już podczas ich projektowania. Dzięki temu możliwe jest wykrywanie i usuwanie występujących w nich błędów, co może być wykonane jeszcze przed wdrożeniem do produkcji. I tak np. w prototypowaniu metodą RP (*Rapid Prototyping*) badania prowadzone są na rzeczywistym obiekcie regulacji, natomiast w sposób symulacyjny odwzorowywane są algorytmy sterowania. Coraz częściej w badaniach, monitorowaniu i prototypowaniu układów sterowania są obecnie wykorzystywane techniki HWILS (ang. *Hardware in the Loop Simulation*).

2. TECHNIKI SYMULACJI UKŁADÓW REGULACJI

Pod pojęciem terminu "*Rapid Prototyping*" (RP) były pierwotnie rozumiane zastosowania z obszaru technik kształtowania mechanicznego. Jednak obecnie w obszarze złożonej elektroniki, która składa się z hybrydowych systemów sprzętu i oprogramowania, dla skrócenia cyklu rozwojowych i zwiększenia ich efektywności rozszerza się zastosowanie metod RP [5]. System taki można przy tym w wielu przypadkach podzielić na składniki i obwody do obsługi, sterowania / regulacji itp. Przy pomocy narzędzi CASE (*Computer Aided Software Engineering*), w zależności od wymagań i nakładów, z wykorzystaniem modelu komputerowego można odwzorować albo tylko poszczególne składniki albo też cały system i jego własności. W modelach symulacyjnych nie mogą być oczywiście uwzględniane wszystkie rzeczywiste oddziaływania. Do testowania funkcjonalności układu sterowania lub regulacji do układów symulacji są pomocne *rzeczywiste obwody*.

Sformalizowany opis systemów technicznych za pomocą CASE-Tools, w porównaniu ze zwykłymi sposobami postępowania przy specyfikowaniu wymagań zapewnia następujące korzyści:

- uzyskuje się sformalizowany i tym samym jednoznaczny przebieg opisu,
- następuje redukcja prób i zmniejszenie ilości budowanych prototypów (wzorców funkcjonalnych), co oszczędza koszty i czas rozwoju,

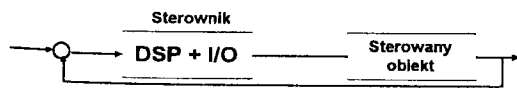
- wewnętrzne wielkości stanu systemu są dostępne dla projektanta,
- możliwe jest szybkie dopasowanie parametrów i struktury (optymalizacja);
- następuje wczesne rozpoznanie błędów rozwojowych i planowania,
- w przypadku szerokiego wykorzystywania w przedsiębiorstwie systemów RP powstają biblioteki modułów, które umożliwiają ponowne wykorzystanie już wypróbowanych algorytmów,
- wiele CASE-Tools oferuje możliwość generowania kodu w języku C z algorytmów utworzonych najczęściej graficznie. Te kody za pomocą *cross-compiler*'a można skompilować na docelowy sprzęt. Dzięki temu uzyskuje się „przechodność” formalnego opisu przez symulację aż do implementacji.

Problematyczna jest jednak symulacja systemów, które są opisywane niedokładnie tylko matematycznie lub też są bardzo złożone. Prowadzi to do niepewności w modelu i do wysokich nakładów czasowych podczas analizy systemu. Może tutaj pomóc implementacja systemu HWILS:

- złożone modele systemu są zastępowane realnym środowiskiem,
- wielkości mierzone można uzyskiwać za pomocą czujników (*Post-Processing*), które służą następnie do walidacji lub polepszenia modeli symulacyjnych,
- przebiegi prób można powtarzać metodą *teach-in* lub przez wprowadzanie wartości zadanych.

Oprócz wymienionych już zalet należy także w wielu przypadkach wymienić również wysokie koszty zestawienia systemu złożonego z *CASE-Tool* i sprzętu. Wysoka elastyczność takiego systemu jest uzyskiwana dzięki odpowiednio dużej mocy obliczeniowej procesu symulacji i wysokiej funkcjonalności narzędzi programowych, ale wskutek tego wzrastają także koszty.

W symulacji układów regulacji posługiwano się dotychczas najczęściej metodą *Rapid Prototyping* ("szybkie prototypowanie"), którą można krótko scharakteryzować jako metodę w której komputer czasu rzeczywistego, z różnego rodzaju urządzeniami wejścia / wyjścia, jest połączony ze sterowanym obiektem. Za pomocą komputera symulowane jest zachowanie się sterownika, a celem symulacji jest sprawdzenie poprawności algorytmu sterowania lub doboru nastaw regulatora. Schemat prototypowania układów sterowania według tej metody został przedstawiony na rys. 1.



Rys. 1 Klasyczna metoda prototypowania układów sterowania techniką RP [3]

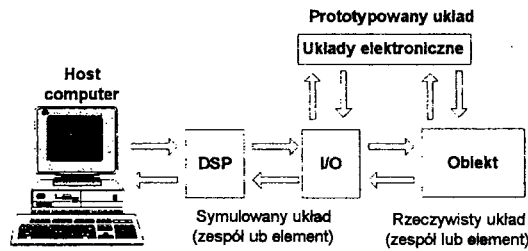
W ostatnich kilku latach, dzięki rozwojowi sprzętu informatycznego i oprogramowania, coraz większego znaczenia nabiera technika HWILS. Koncepcja metody HWILS jest odwrotna do metod RP (rys. 2). Sterownik jest tutaj układem rzeczywistym, natomiast symulowany jest obiekt sterowany, którego koszt wykonania jest najczęściej wielokrotnie wyższy niż koszt wykonania (lub zakupu) sterownika. Umożliwia to prototypowanie układów sterowania w taki sposób, że rolę obiektu sterowanego spełnia jego model matematyczny, symulowany w czasie rzeczywistym. Metoda ta polega zatem na zastąpieniu rzeczywistego obiektu podczas badań poprawności działania sterownika, przez symulację w czasie rzeczywistym modelu obiektu z uwzględnieniem napędów, sensorów itp.



Rys. 2 Zasada prototypowania układów sterowania metodą HWILS [3]

Sygnaly sterujące są wysyłane przez sterownik do interfejsów komputera, za pomocą którego realizowana jest symulacja sterowanego obiektu w czasie rzeczywistym. Podobna zasada dotyczy sygnałów przesyłanych z czujników na obiekcie do sterownika. Z punktu widzenia badanego sterownika "odczuwa" on symulowany model w taki sposób jak rzeczywisty obiekt, o ile sygnały generowane w procesie symulacji wiernie odwzorowują sygnały występujące w rzeczywistym

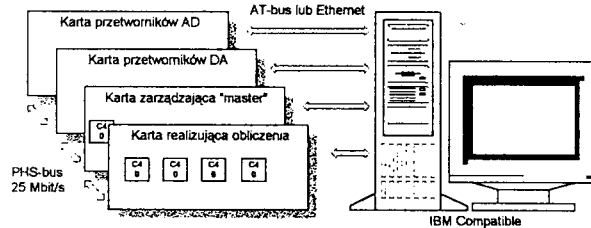
obiekcie. Warunkami zgodności tych sygnałów są m.in. adekwatność modelu i obiektu oraz wystarczająco duża moc obliczeniowa komputera, na którym jest realizowana symulacja, tak aby zapewniona była możliwość realizacji symulacji w czasie rzeczywistym.



Rys.3 Symulacja części układu sterowania

elementów sterowanego obiektu (rys.3). Wraz ze wzrostem złożoności symulowanego obiektu rosną także wymagania odnośnie do mocy obliczeniowej. Proces symulacji wymaga zatem zastosowania dobrej klasy sprzętu o wysokich parametrach technicznych. Ponieważ tej klasy sprzęt jest drogi, dlatego też możliwe jest zastosowanie specjalizowanych kart procesorów sygnałowych DSP umieszczanych w komputerach klasy PC. Karta procesora sygnałowego przejmuje wówczas całość obliczeń, a komputer służy jedynie do komunikacji z użytkownikiem i między różnego rodzaju interfejsami. Jednym z przodujących producentów kart z procesorami sygnałowymi jest niemiecka firma dSPACE. Firma ta, opierając się na procesorach sygnałowych TEXAS Instruments, opracowała wiele kart oraz oprogramowanie do ich obsługi (rys.4), które są szeroko stosowane w przemyśle lotniczym, samochodowym, maszynowym itp. do realizacji procesu HWILS.

W procesie symulacji sygnały sterujące wysyłane przez sterownik są wprowadzane do urządzenia na którym odbywa się symulacja (np. komputer z odpowiednim oprogramowaniem). Z drugiej strony w czasie symulacji obiektu są również generowane sygnały, które są podawane na wejście sterownika. Symulowane obiekty mogą być również częścią złożonego zespołu, co pozwala na badania symulacyjne wybranych



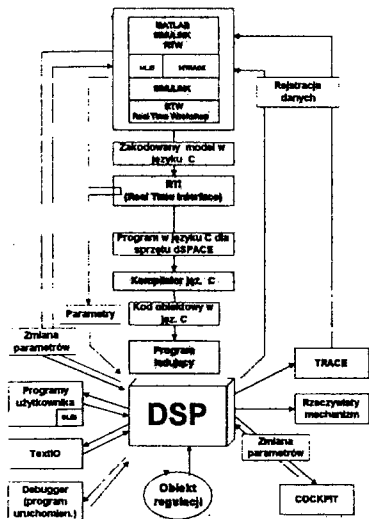
Rys.4 Przykładowy zestaw firmy dSPACE do realizacji procesu HWILS wg [2]

Na rysunku 5 pokazano natomiast przykład środowiska do symulacji metodą HWILS w oparciu o produkty firmy MathWorks.

3. STANOWISKO DO BADAŃ SYMULACYJNYCH

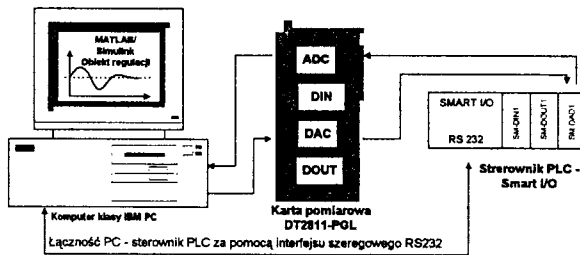
3.1. Opis stanowiska do symulacji

W budowie stanowiska do symulacji układów regulacji procesów ciągłych zastosowano wariant rozwiązania programowo-sprzętowego. Wynikało to z optymalnych kosztów wykonania takiego stanowiska oraz możliwości fizycznego wydzielenia sygnałów obiektowych i sterujących. Założono, że obiekty będą symulowane na odpowiednio oprogramowanym komputerze klasy PC, pracującym pod kontrolą systemu operacyjnego DOS oraz interfejsu graficznego WINDOWS 3.1 lub systemu operacyjnego WINDOWS 95. Programem umożliwiającym symulację układów regulacji jest pakiet MATLAB®/Simulink firmy MathWorks. Natomiast rolę regulatora spełnia sterownik swobodnie programowalny PLC (ang. *Programmable Logical Controller*) typu SMART I/O firmy PEP Modular Computers, który może być programowany przy pomocy systemu ISaGRAF.



Komunikacja między sterownikiem SMART I/O a komputerem odbywa się za pomocą interfejsu szeregowego RS 232 oraz za pomocą karty pomiarowej z 12-bitowymi przetwornikami D/A, A/D (cyfrowo-analogowe i analogowo-cyfrowe) oraz DIN i DOUT (wejściami i wyjściami dwustanowymi). Zastosowana karta pomiarowa typu DT2811-PGL umożliwia również podłączenie innych urządzeń takich jak oscyloskop, mierniki itp. do obserwacji przebiegów symulowanego procesu. Na rysunku 6 pokazano konfigurację zestawionego stanowiska do badań symulacyjnych.

Rys.5 Kompletnie środowisko programowo-sprzętowe do realizacji symulacji HWILS [2]



Rys.6 Schemat zbudowanego stanowiska do badań symulacyjnych układów regulacji

3.2. Program Matlab® / SIMULINK, pakiet RTW i tryb zewnętrzny

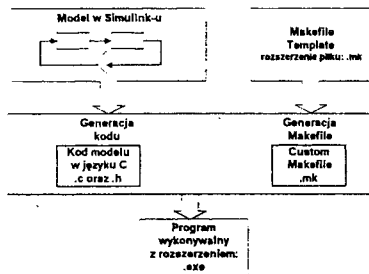
Do badań symulacyjnych regulacji procesów ciągłych w zestawionym stanowisku wykorzystano program MATLAB® firmy MathWorks wraz z pakietem symulacyjnym SIMULINK. Program ten pracuje w środowisku graficznym WINDOWS 3.1 lub WINDOWS 95, lecz także może pracować w systemach UNIX-owych. Program MATLAB® umożliwia budowanie i dołączanie własnego lub istniejącego specjalizowanego oprogramowania (ang. *toolbox*). Ponadto program ten zapewnia duże możliwości wizualizacji danych i wyników obliczeń (różnorodne wykresy oraz grafika dwu- i trójwymiarowa), a także wybór i zmianę platformy sprzętowej: od komputerów klasy IBM-PC, poprzez stacje robocze UNIX do superkomputerów. Program ten pozwala również na wykonywanie zaawansowanych operacji w zakresie obliczeń numerycznych (m.in. przetwarzanie i analiza danych, interpolacja, aproksymacja, całkowanie numeryczne, rozwiązywanie zagadnień początkowych dla równań różniczkowych zwyczajnych, metody liniowej algebry numerycznej) oraz umożliwia budowę i analizę modeli dynamicznych dla potrzeb sterowania, konstrukcji układów regulacji i kontroli ich własności. Znaczne uproszczenie symulacji układów automatycznej regulacji uzyskuje się za pomocą interaktywnego pakietu SIMULINK. Umożliwia on tworzenie wielopoziomowych schematów blokowych. Bloki funkcyjne są zobrazowane w postaci ikon, a ich wzajemne relacje w postaci połączeń. SIMULINK może współpracować między innymi z pakietem RTW (ang. *Real Time Workshop*), który umożliwia generowanie programów działających w czasie rzeczywistym i zawiera generator języka C i asemblera dla różnych procesorów, w tym sygnałowych DSP. SIMULINK

może być stosowany do modelowania i symulacji zarówno układów dynamicznych jak i do projektowania oraz testowania urządzeń i systemów sterujących, np. metodą HWILS. Budowanie obiektów w tym programie wymaga wybrania z odpowiednich grup bibliotek bloków i połączeń. W skład biblioteki bloków wchodzi następujące grupy: *sources* (źródła) - różnego rodzaju bloki wejściowe, generatory; *sinks* (odbiorniki) - bloki wyjściowe, oscyloskopy, bloki zapisujące wyniki symulacji do przestrzeni roboczej (macierzy); *discrete* (bloki dyskretne) - wykorzystywane w modelach dyskretnych; *linear* (bloki liniowe) - wykorzystywane w modelach ciągłych; *nonlinear* (bloki nieliniowe); *connection* (bloki połączeniowe) - zapewniające przekazywanie sygnałów oraz *extras* (biblioteki dodatkowe) - bloki regulatorów PID, bloki przerzutników sterowanych zegarem itp.

Pakiet RTW jest generatorem kodu języka C (wersja ANSI lub K&R) dla środowiska SIMULINK. Umożliwia on tworzenie kodu w języku C bezpośrednio z graficznego modelu SIMULINK'a i automatyczne utworzenie programów wykonywalnych, które mogą być później uruchamiane w czasie rzeczywistym w różnych środowiskach. Ogólnie aplikacje tworzone za pomocą RTW można podzielić na dwie kategorie: RT (*real-time*) i nonRT (*nonreal-time*), czyli działające i nie działające w czasie rzeczywistym. W systemach czasu rzeczywistego są możliwe następujące warianty:

- wygenerowanie kodu C z projektowanego modelu przez posłużenie się systemem MATLAB-SIMULINK i załadowanie go później do sterownika mikroprocesorowego,
- HWILS, czyli możliwość naśladowania przez SIMULINK symulowanego obiektu i dokonywania na nim pomiarów w czasie rzeczywistym, po zainstalowaniu w systemie odpowiedniej karty pomiarowej oraz
- interaktywne dostrajanie parametrów regulatora bez przerywania jego pracy.

Pakiet RTW można też wykorzystać w aplikacjach nie działających w czasie rzeczywistym do wygenerowania kodu. Zaletą takiego rozwiązania jest znacznie zwiększona szybkość działania takiej aplikacji. Istnieje również możliwość automatycznej generacji kodu za pomocą okna graficznego SIMULINK'a, w którym ustawia się wymagane dla programu *make* opcje, natomiast bardziej złożone zadania realizuje się z poziomu systemu operacyjnego - wówczas wymagana jest znajomość parametrów wywołań kompilatora i *linker*'a (programu łączącego). Poszczególne kroki tworzenia wykonywalnego programu pokazano na rys.7. Do utworzenia zbioru *Makefile* wykorzystywane są zbiory skryptowe MATLAB-a, tzw. M-zbiory: *make_rt.m*, dla aplikacji RT oraz *make_nrt.m* dla aplikacji nonRT.



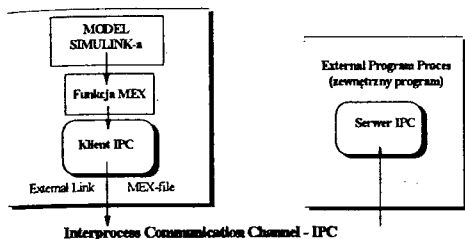
Rys.7 Proces tworzenia programu *Real Time*

Oprócz tego brane są pod uwagę informacje zawarte w oknie dialogowym generowania kodu SIMULINK'a, które pozwalają na określenie, które zbiory powinny być skompilowane i połączone. Istnieje również możliwość pracy w trybie komunikacyjnym poprzez specjalny kanał komunikacyjny IPC (*Interprocess Communication Channel*). Jest to mechanizm umożliwiający współpracę między SIMULINK'iem i zewnętrznym (samodzielnym) programem oraz zmianę parametrów w modelu SIMULINK'a bez przerywania pracy i powtórnej kompilacji.

W terminologii klient/serwer SIMULINK jest wówczas klientem, a program zewnętrzny serwerem (rys.8). Przy zmianie parametrów w modelu wywoływany jest *external link MEX-file*, który ma za zadanie przekazanie przez IPC parametrów jako argumentów do zewnętrznego programu.

Do kontroli przerw wykorzystywany jest układ zegara systemowego 8254 (*Counter/Timer*). Jest to układ 16-bitowy, taktowany z częstotliwością 1,193 MHz i umożliwia on uzyskanie częstotliwości próbkowania w zakresie 18 Hz - 400 kHz, co odpowiada okresowi 0,056 - 2,5 10⁻⁶ s. Razem z

pakiem RTW dostarczana jest biblioteka programów obsługujących komunikację z kartami pomiarowymi np. DT2811.



Rys.8 Symulacja w trybie zewnętrznym

3.6. Program ISaGRAF

Kolejnym programem, który wykorzystano w budowie stanowiska do symulacji układów regulacji jest ISaGRAF. ISaGRAF jest programem mającym następujące cechy: umożliwia on pisanie programów dla sterowników PLC, m.in. PEP Smart I/O, śledzenie wykonywania programu, pracę krokową, podgląd zmiennych podczas wykonywania programu, symulację programu

przed załadowaniem go do sterownika, wykonywanie cyklicznych programów w języku SFC (ang. *Sequential Function Chart*), możliwość programowania w języku C, wielozadaniowość, jak również wykonywanie pojedynczych zadań. W systemie ISaGRAF są do dyspozycji języki programowania sterowników PLC wg IEC 1131-3 tzn.: SFC (ang. *Sequential Function Chart*) - wykorzystuje się tutaj kroki (programowane w językach ST, IL, LD i FDB) i tranzycje (logiczne powiązania między krokami); FBD (ang. *Function Block Diagram*) - język graficzny wykorzystywany przy złożonych cyklicznych operacjach, przy czym budowanie programu odbywa się przez wybór odpowiednich bloków funkcyjnych z biblioteki i połączenie ich w dany schemat; LD (ang. *Ladder Diagram*) - graficzny język drabinkowy wykorzystywany tylko do operacji logicznych, wykorzystujący kontakty (wejścia argumentów) oraz wyjścia umieszczane między dwoma szynami zasilania; ST (ang. *Structured Text*) - język wysokiego poziomu wykorzystywany we wszystkich cyklicznych operacjach przy tworzeniu złożonych procedur automatycznego przetwarzania, które nie dają się przedstawić za pomocą języków graficznych oraz IL (ang. *Instruction List*) - język podobny do programowania mnemonicznego wykorzystywany w programowaniu niskiego poziomu, który jest wysoce efektywny przy małych aplikacjach lub do optymalizacji części aplikacji.

3.7. Sterownik micro PLC oraz komputer czasu rzeczywistego PEP SMART I/O

SMART I/O jest urządzeniem zawierającym w sobie sterownik micro PLC oraz komputer czasu rzeczywistego. Przeznaczony jest on do różnych zastosowań w przemyśle. SMART I/O można programować za pomocą środków narzędziowych programu ISaGRAF, wykorzystując do tego celu języki programowania PLC wg IEC 1131-3 oraz kompilator Ultra-C ANSI-C (DOS, OS-9) do programowania w czasie rzeczywistym. SMART I/O ma także interfejs RS 485 PROFIBUS, umożliwiając wykorzystanie tego sterownika w sieci PROFIBUS. SMART I/O jest oparty na procesorze Motorola MC68302. Procesor ten składa się z dwóch *chip*'ów mikroprocesorowych. Pierwszy to 68HC000 taktowany zegarem 20 MHz, a drugi to procesor typu RISC zorientowany głównie na komunikację. Sterownik ten składa się z głównego modułu SMART-BASE oraz z modułów dodatkowych SMART-EXT, dołączanych na wspólnej szynie. Sterownik SMART I/O (część SMART-BASE), ze względu na zakres operacji potrzebnych do zrealizowania zadań symulacji, został wyposażony w układy SM-DIN1, SM-DOUT1 i SM-DAD1.

4. PRZYKŁAD SYMULACJI UKŁADU REGULACJI

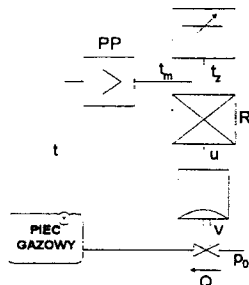
4.1. Wstęp

Sprawdzenie przydatności zestawionego stanowiska do badań symulacyjnych przeprowadzono na przykładzie regulacji temperatury pieca gazowego. Symulacja została przeprowadzona z wykorzystaniem we / wy analogowych sterownika PLC PEP typu Smart I/O. Karta pomiarowa

obsługiwała we / wy jako sygnały bipolarne, ale ponieważ w jej przypadku zakres dopuszczalnych napięć wynosi $\pm 5V$, to występowało niedopasowanie poziomu napięć między kartą a sterownikiem PLC. Problem ten rozwiązano za pomocą dodatkowego układu dopasowującego sygnały we/wy. Obiekt regulacji, tzn. piec gazowy, został zamodelowany w SIMULINK'u, natomiast rolę regulatora PID pełnił odpowiednio oprogramowany (za pomocą ISaGRAF'a) sterownik PLC PEP Smart I/O. Karta pomiarowa DT2811-PGL, przez odpowiednie sekcje przetworników ADC i DAC, umożliwiała natomiast dopasowanie sterownika PLC do programu Matlab®/SIMULINK.

4.2. Opis układu symulacji regulacji temperatury pieca gazowego

Obiekt regulacji jest piec gazowy do produkcji szkła stosowanego w wytwarzaniu kineskopów telewizyjnych, którego schemat pokazano na rys.9. Wielkością regulowaną jest temperatura w piecu, która jest mierzona za pomocą przetwornika pomiarowego PP. Natomiast głównym zakłóceniem występującym podczas pracy pieca jest zmiana ciśnienia gazu p_0 . Ciśnienie gazu dopływającego do pieca jest regulowane przez zawór. Sygnał wychodzący z regulatora steruje ustawieniem trzpienia zaworu doprowadzającego gaz i w ten sposób reguluje temperaturę w piecu.

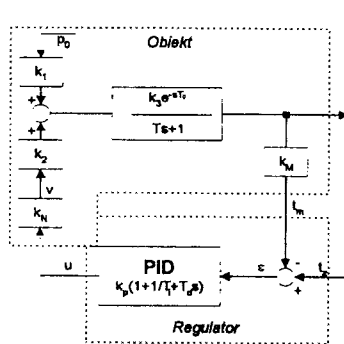


gdzie:

- PP - przetwornik pomiarowy temperatury,
- Q - natężenie przepływu gazu,
- p_0 - ciśnienie gazu,
- T_0 - opóźnienie obiektu,
- R - regulator,
- v - ustawienie zaworu,
- t_z - temperatura zadana,
- t_m - temperatura mierzona,
- t - temperatura pieca

Rys.9 Schemat układu regulacji temperatury pieca gazowego

Na rysunku 10 pokazano schemat blokowy symulowanego układu regulacji temperatury pieca.



$$k_1 = 1,96 \cdot 10^{-2} \frac{\frac{m^3}{kN}}{\frac{s}{m^2}} \quad k_2 = 6,66 \frac{\frac{m^3}{s}}{m}$$

$$k_3 = 1,53 \cdot 10^2 \frac{^\circ C}{\frac{m^3}{s}} \quad k_M = 0,784 \frac{\frac{kN}{m^2}}{^\circ C}$$

$$k_N = 10^{-3} \frac{\frac{m}{mA}}{s} \quad T_0 = 15s, T = 60s$$

Rys.10 Schemat blokowy układu regulacji temperatury pieca gazowego

Od tego układu wymaga się, żeby uchyb ustalony temperatury spowodowany trwałą zmianą ciśnienia był możliwie mały oraz żeby, ze względu na jakość szkła, czas regulacji t_r (czas trwania stanu nieustalonego) nie przekraczał 400 s. Dopuszcza się przeregulowanie 20%. W układzie tym występują następujące wielkości: k_1 - współczynnik wrażliwości przepływu gazu na zmiany ciśnienia zasilania, k_2 - współczynnik wrażliwości przepływu gazu na zmiany położenia trzpienia zaworu, k_3 -

współczynnik wzmocnienia pieca, k_M - współczynnik wzmocnienia czujnika i przetwornika pomiarowego, k_N - współczynnik wzmocnienia nastawnika, T_o - opóźnienie obiektu, T - stała czasowa obiektu. Na podstawie tych wymagań i danych zawartych w [1] zastosowano regulator PID o następujących nastawach:

$$k_p = \frac{1,2}{k_{ob} \frac{T_o}{T}} = \frac{1,2}{3,149 \cdot \frac{15}{60}} = 1,52 \quad T_i = 2,0T_o = 2,0 \cdot 15 = 30 \quad T_d = 0,4T_o = 0,4 \cdot 15 = 6$$

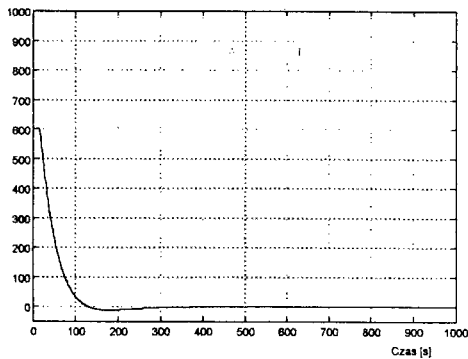
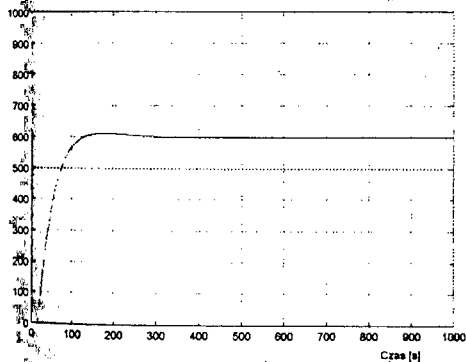
przy czym całkowity współczynnik wzmocnienia obiektu wynosi:

$$k_{ob} = (k_N k_2 + k_1) k_3 k_M = k_M k_N k_2 k_3 + k_M k_1 k_3 = 3,149$$

$$\left[\frac{kN}{m^2 \text{ } ^\circ C} \frac{m}{mA} \frac{m^3}{s} \frac{^\circ C s}{m^3} + \frac{kN}{m^2 \text{ } ^\circ C} \frac{m^3 m^2}{s} \frac{^\circ C s}{kN} \frac{1}{m^3} \right] = \left[\frac{kN}{m^2 mA} \right]$$

Przyjęto wartość opóźnienia obiektu $T_o=15$ s, natomiast wartość zadaną temperatury $t_z=600^\circ C$.

Na rys.11 pokazano przebieg sygnału regulowanego, a na rys.12 przebieg uchybu regulacji, które uzyskano podczas symulacji układu regulacji na zestawionym stanowisku badawczym. Jak można zauważyć zadanie regulacji zostało spełnione: osiągnięto wartość zadaną temperatury po czasie nie przekraczającym 400 s, przy przeregulowaniach mniejszych niż 20 %.



Rys.11 Przebieg sygnału wielkości regulowanej

Rys.12 Przebieg uchybu regulacji temperatury

5. ZAKOŃCZENIE

W ostatnich latach symulatory układów sterowania i regulacji zdobywają sobie coraz większą popularność, która zawdzięcza między innymi wymaganiom zmniejszenia czasu i kosztów prac rozwojowych. Poprzez zastosowanie nowoczesnej techniki obliczeniowej stało się możliwe szybkie testowanie projektowanych układów sterowania, wykrywanie występujących w nich błędów oraz ich usuwanie jeszcze przed wdrożeniem do produkcji. Badania symulacyjne przeprowadzone na przykładzie regulacji temperatury pieca gazowego w pełni potwierdziły przydatność zestawionego stanowiska do testowania układów regulacji.

6. LITERATURA

- [1] Fndeisen W.: *Poradnik Inżyniera - Automatyka*, WNT, 1973
- [2] *Materiały reklamowe firm MathWorks i dSPACE*
- [3] Uhl T.: *Hardware-in-the-loop-simulation - nowa metoda prototypowania układów sterowania i monitorowania*, PAK, 5/1994
- [4] Węgrzyn W., *Badania symulacyjne układów sterowania i regulacji z wykorzystaniem techniki HWILS i logiki rozmytej*, Praca dyplomowa, Wyd. Mech. PWr, Wrocław 1997
- [5] Beckmann G., Varchim, Kusiek A., Hesselbach, *Einsatz eines Rapid-Prototyping-Systems als Entwicklungshilfe neuer Prozesse*, SPC/IPC/Drives - Messe und Kongres, Nürnberg 1997