1. Dominik BODORA[1], 2. Tomasz KŁOPOT[2], 3. Krzysztof STEBEL[3]

Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki (1), Politechnika Śląska, Katedra Automatyki i Robotyki (2),(3)
ORCID: 2. 0000-0003-1247-5399; 3. 0000-0003-2912-6742

doi:10.15199/48.2024.01.17

Wykorzystanie chmur obliczeniowych do sterowania obiektami przy pomocy złożonych obliczeniowo algorytmów

Streszczenie. *Mechanizm zaimplementowany w chmurze obliczeniowej dokonuje doboru nastaw regulatora DMC, jeżeli nastawy nie znajdują się w bazie danych. Wyznaczone wartości przesyłane są do sterownika PLC, na którym zaimplementowane jest prawo sterowania regulatora DMC oraz model obiektu. Dobrane nastawy będą weryfikowane przy pomocy modelu fragmentu instalacji dystrybucji ciepła dostępnej w laboratorium Katedry Automatyki i Robotyki, który będzie pełnił rolę obiektu regulacji.*

Abstract*. The mechanism implemented in the computing cloud parameterizes settings of the DMC controller if the settings are not in the database. The computed values are sent to the PLC controller on which the control law of the DMC controller and the object model are implemented. The selected settings are going to be verified on a model of a fragment of the heat distribution system available in the laboratory located in the Department of Automation and Robotics, which is going to act as a control facility. (**The use of cloud computing to control objects using computationally complex algorithms**)*

Słowa kluczowe: regulator DMC, chmura obliczeniowa, parametryzacja, PLC
Keywords: DMC controller, cloud computing, parametrization, PLC

Wstęp

Przemysł 4.0 jest szeroko analizowany i prezentowany we współczesnych publikacjach [1]. Głównym powodem takiej sytuacji jest postęp technologiczny, który niesie ze sobą kolejna rewolucja przemysłowa. Postęp ten wpływa na wiele sektorów, w szczególności na sektor przemysłowy dając przy tym wiele zauważalnych korzyści [2][3].

Pojęcie chmur obliczeniowych jest blisko powiązane z Przemysłem 4.0. Zalicza się ono bowiem do grupy jego filarów, w której znajduje się miedzy innymi: Przemysłowy Internet Rzeczy (IIOT), bezpośrednia komunikacja maszyna-maszyna (M2M), czy przetwarzanie Big Data [4]. Stosowanie chmur obliczeniowych oferuje szereg korzyści dla organizacji usprawniając ich działalność. Pomaga obniżyć koszty organizacji związane z wdrożeniem wielu najnowocześniejszych technologii usług informacyjnych. Usługi są skalowane do zapotrzebowania klienta. Dodatkowo proces wdrożenia przebiega bardzo szybko, ponieważ zasoby są zarządzane przez oprogramowanie. Przetwarzanie obniża bariery IT dla innowacji, dzięki czemu powstaje coraz więcej obiecujących startupów [5]. Dostępne artykuły rozwijają tematykę związaną z wykorzystaniem chmur obliczeniowych, między innymi w inteligentnych domach [6][7], bądź podczas rozwiązywania problemów optymalizacji [8].

W tym artykule przedstawiono wykorzystanie chmur obliczeniowych do parametryzacji regulatora sterującego obiektem. Wykorzystany algorytm sterowania należy do grupy algorytmów złożonych obliczeniowo. W takim przypadku wykorzystanie chmur obliczeniowych jest bardziej uzasadnione, niż bezpośrednia implementacja algorytmu na sterowniku PLC.

Koncepcja układu sterowania

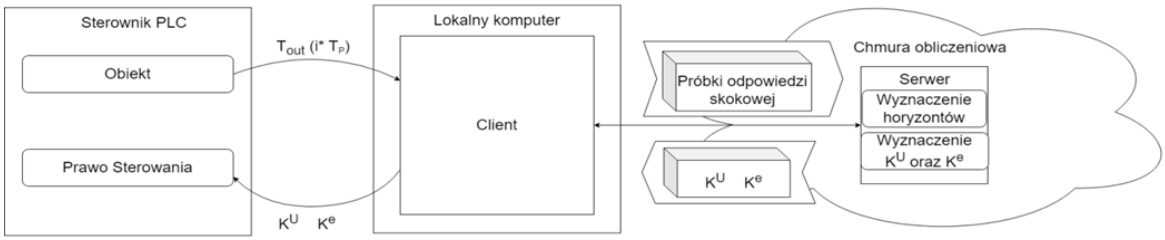
Koncepcję układu sterowania z wykorzystaniem chmury obliczeniowej można zobaczyć na rysunku 1. Każdy z trzech modułów jest odpowiedzialny za konkretne zadania w procesie parametryzacji.

Na sterowniku zostało zaimplementowane prawo sterowania, dzięki któremu możliwe jest sterowania obiektem. Weryfikacja działania parametryzacji regulatora będzie jest przeprowadzana na obiekcie.

Na lokalnym komputerze wdrożony jest klient, który jest węzłem pośrednim w procesie wymiany danych pomiędzy sterownikiem, a serwerem. Zadaniem klienta jest zbieranie potrzebnych próbek z symulatora, zapis danych w postaci wektora oraz przesłanie zbioru do chmury. Klient wysyła żądanie do serwera o obliczenie potrzebnych parametrów, które z kolei trafiają z powrotem do sterownika PLC.

Ostatnim węzłem jest chmura obliczeniowa, na której zaimplementowano serwer. Po otrzymaniu żądania i potrzebnego wektora próbek program dokonuje parametryzacji regulatora. W pierwszej kolejności wyznacza horyzonty, które są ściśle powiązane z algorytmem regulacji. Ich znaczenie zostanie wyjaśnione w późniejszej części artykułu. Następnie obliczane są wektory K^U oraz K^e, które są istotne dla prawa sterowania.

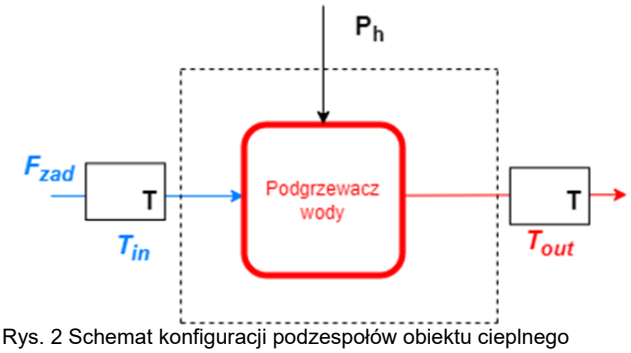
Zaimplementowane programy na lokalnym komputerze i chmurze obliczeniowej zostały napisane w języku Python. Python dostarcza gotową bibliotekę o nazwie snap7 umożliwiającą komunikację ze sterownikiem. Komunikacja pomiędzy klientem, a serwerem oparta jest o połączenie gniazdowe. Użycie takiego sposobu komunikacji wymaga jednak poprawnego skonfigurowania chmury obliczeniowej, co pokazano w dalszej części artykułu.



Rys.1 Schemat ideowy układu sterowania

Model obiektu

W celu weryfikacji poprawności działania układu sterowania wykorzystany został model zbudowany na podstawie obiektu dystrybucji ciepła, którego przedstawiono na rysunku 2.



Rys. 2 Schemat konfiguracji podzespołów obiektu cieplnego

Do pogrzewacza o pojemności V = 1,6 L wpływa woda miejska o temperaturze T_{in}, °C z zadanym przepływem F_{zad}, L/min. W obiekcie przepływ na wejściu podgrzewacza nie jest równy bezpośrednio przepływowi zadanemu i wymagana jest jego stabilizacja. W symulatorze nie występuje taka sytuacja, bowiem przepływ zapisany jest w zmiennej programu. Z tego powodu rysunek 2 przedstawia tylko część całej instalacji. Moc modyfikowana jest za pomocą zmiennej P_h w zakresie od 0-100 % mocy nominalnej grzałki P_{nom} równej 12kW. Woda o podwyższonej temperaturze wypływa z podgrzewacza wody z mierzoną temperaturą T_{out}, °C .

Model obiektu potrzebny do symulacji zaczerpnięto z artykułu [9]. Pierwsza część modelu jest nieliniowym równaniem dynamicznym pierwszego rzędu (1) .

$$\frac{dT_{out}^{*}}{dt} = \frac{F}{60V}(T_{in} - T_{out}^{*}) + \frac{P_h P_{nom}}{100 c_S \rho_S V} \tag{1}$$

gdzie: c_s – ciepło właściwe równe 4200, J/kg°C, ρ_s – gęstość wody równa 1, kg/l oraz T^{*}_{out} – niemierzalna temperatura, °C.

Uwzględnienie dynamiki czujników, podgrzewacza wody i innych pominiętych zjawisk występujących w obiekcie pozwala na określenie mierzalnej temperatury. W tym celu należy zastosować (2)

$$K(s) = \frac{T_{out}(s)}{T_{out}^{*}(s)} = \frac{k(P_h)}{(1 + s\tau_1(F))(1 + s\tau_2(F))} e^{-s\tau_0} , \tag{2}$$

gdzie

$$k(P_h) = -0,0002347 \cdot P_h + 1,012 , \tag{3}$$

$$\tau_1(F) = \tau_2(F) = 19,08 \cdot F^{-0,4293} - 4,042 , \tag{4}$$

$$\tau_0 = 11,93 \cdot F^{-0,78} + 2,73 , \tag{5}$$

Wzmocnienie (3) jest zależne od aktualnej wartości procentowej mocy podanej na grzałkę. Stałe czasowe (4) t_1 i t_2 reprezentują dodatkową dynamikę zależną od przepływu, a t_0 jest opóźnieniem (5).

Algorytm DMC

Algorytm DMC należy do rodziny algorytmów regulacji predykcyjnej. Jest to szeroka klasa algorytmów sterowania dyskretnego wykorzystująca bezpośrednio model matematyczny sterowanego obiektu. W przypadku algorytmu DMC obiekt jest aproksymowany modelem pierwszego rzędu z opóźnieniem (FOPDT). Głównym zadaniem algorytmu DMC jest wyznaczenie ciągu przyszłych zmian wielkości sterującej dla danej chwili przy minimalizacji wskaźnika jakości sterowania J dla podanego horyzontu predykcji H_P (6)

$$J = \sum_{p=1}^{H_P} (y_{k+p}^{zad} - y_{k+p}^{pred})^2 + \lambda \cdot \sum_{P=0}^{H_C-1} (\Delta u_{k+p})^2 , \tag{6}$$

gdzie: y_{k+p}^{zad} – wartość zadana w chwili k+p, y_{k+p}^{pred} – wartość wielkości regulowanej predykowana w chwili k+p, Δu_{k+p} – przyrost wielkości sterującej, H_P – horyzont predykcji, H_C – horyzont sterowania, λ- współczynnik wagowy.

We wzorze 6 można zauważyć wpływ niektórych horyzontów, o których wspomniano we wcześniejszej części artykułu. Horyzonty określają w głównej mierze wielkości poszczególnych sum, tzn. ilość sumowanych zmiennych w danej sumie. Dodatkowo występuje parametr λ, który jest pewną karą nakładaną na przyrost wartości wielkości sterującej. Zmiana wartości λ pozwala na kształtowanie dynamiki regulacji.

Szczegółowe kroki realizacji algorytmu DMC dla obiektu jednowymiarowego (SISO) przedstawiono w [10]. W artykule wykorzystano zasady strojenia, tzn. sposób doboru nastaw zaproponowane przez autora artykułu [11].

Parametryzacja regulatora DMC opiera się w główniej mierze na operacjach macierzowych, które są złożone obliczeniowo. Uzasadnione jest w takim wypadku zastosowanie chmur obliczeniowych, do przeprowadzenia procesu parametryzacji. Macierze te wcześniej zostały wypełnione zgodnie z metodą opisaną w [10] próbkami odpowiedzi skokowej pobranymi z okresem próbkowania T_P. Końcowym wynikiem podanych operacji są dwa wektory K^U oraz K^e. Wektory są wykorzystywane do wyznaczenia przyrostu wartości wielkości sterującej (7).

$$\Delta u(k) = K^e \cdot [y^{zad}(k) - y(k)] - \sum_{j=1}^{H_D} K_j^U \cdot \Delta u(k - j) , \tag{7}$$

Równanie (7) zaimplementowano bezpośrednio na sterowniku, ze względu na jego małą złożoność obliczeniową. Wyznaczenie przyrostu jest potrzebne do określenia wartości sterującej w danej chwili (8).

$$u(k) = u(k - 1) + \Delta u(k) , \tag{8}$$

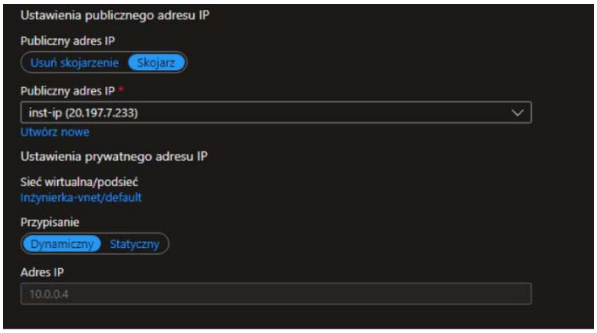
Chmura obliczeniowa

Chmura obliczeniowa wykorzystywana w artykule jest w postaci maszyny wirtualnej, która została utworzona na platformie Microsoft Azure. Wykorzystanie maszyny wirtualnej umożliwia w łatwy sposób wdrażanie aplikacji. Parametry maszyny takie jak: ilość procesorów oraz wielkość pamięci Ram zostały odpowiednio dobrane. System operacyjny utworzonej maszyny to Ubuntu Server 20.04 LTS – Gen2, który oparty jest na wersji 5.4 jadra Linux.

Istnieje parę sposobów na połączenie się z maszyną i uzyskanie dostępu do klienta. Najwygodniejszym sposobem jest połączenie za pośrednictwem protokołu SSH. Wykorzystanie takiego sposobu połączenia wymaga klucza, który jest plikiem do pobrania na stronie dostawcy. Ścieżkę pliku należy podać w wierszu poleceń w wybranej powłoce wiersza poleceń.

Domyślnie utworzona maszyna wirtualna wymaga pewnych dodatkowych konfiguracji, aby umożliwić utworzenie połączenia gniazdowego pomiędzy maszyną wirtualną, a lokalnym komputerem. W tym celu należy utworzonej maszynie wirtualnej przypisać statyczny adres IP, który domyślnie jest dynamiczny (Rys. 3). Taką

konfigurację należy wykonać w interfejsie sieciowym, który jest tworzony razem z maszyną wirtualną.



Rys. 3. Konfiguracja adresu IP

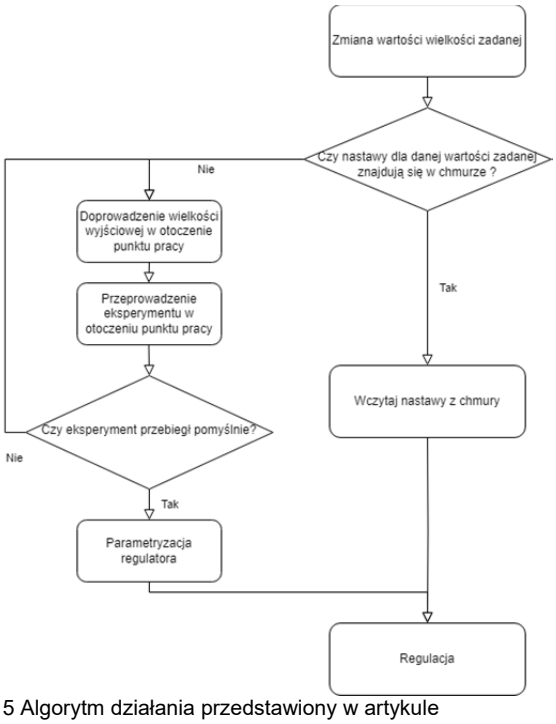
Kolejnym krokiem jest dodanie reguły dla portu, tzn. umożliwienie komunikacji przez konkretny port maszyny wirtualnej. Każda reguła posiada priorytet, którego wartość nie może być niższa niż 300. Taka wartość jest przypisywana dla portu, po którym następuje połączenie za pomocą protokołu SSH. Niektóre wartości portu są zabronione dla użytkowania. Najbezpieczniej jest wybrać port o numerze większym od 1024. Możliwe jest także określenie protokołu transmisji, jednak nie jest to wymagane. Przykład dodania reguły można zobaczyć na rysunku 4.

Priorytet	Nazwa	Port	Protokół
300	SSH	22	TCP
310	Komunikacja	1025	Dowolny

Rys. 4 Reguły portów

Zaproponowane rozwiązanie

Początkowe etapy badań przedstawione w artykule zrealizowano w [12]. Zaproponowany algorytm działania w [12] posiada pewną wadę, bowiem dla każdej zmiany wartości wielkości zadanej przeprowadzana jest parametryzacja regulatora. Parametryzacja regulatora wymaga przeprowadzenia eksperymentu w otoczeniu punktu pracy, przez co zakłócany jest proces technologiczny.



Rys. 5 Algorytm działania przedstawiony w artykule

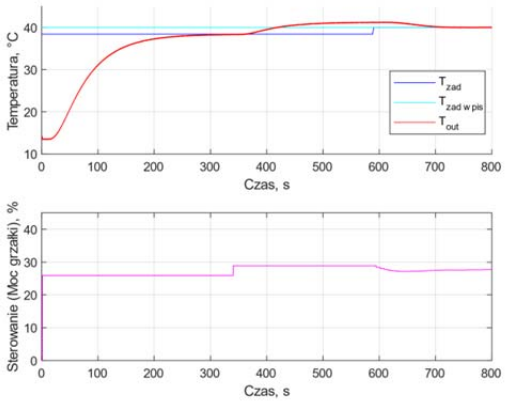
W artykule przedstawiono rozwiązanie, które stara się minimalizować ilość wykonywanych eksperymentów. Modyfikacje algorytmu można zobaczyć na rysunku 5.W tym przypadku wyznaczone wcześniej nastawy regulatora dla podanego punktu pracy są przechowywane w chmurze. Pozwoli to na wczytanie nastaw bezpośrednio z bazy danych, jeśli punkt pracy zostanie powtórnie zadany. Pomija się w takim przypadku proces związany z przeprowadzeniem eksperymentu.

Dodatkowo zaimplementowano specjalne mechanizmy, które zapewniają, że proces parametryzacji zostanie przeprowadzony poprawnie. Przykłady mechanizmów zostaną przedstawione w dalszej części artykułu.

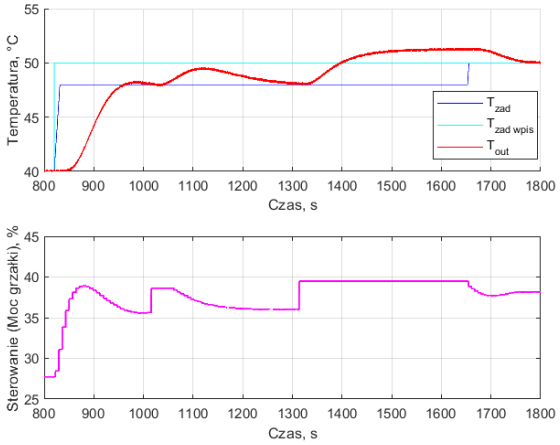
Wyniki oraz wnioski

Symulacja rozpoczyna się od procesu inicjalizacji (pierwsza faza 0-340 s), tzn. wstępnego określenia nastaw dla punktu pracy 40 °C oraz przepływu nominalnego 1,8 l/min (Rys. 6). Z racji braku początkowych nastaw w regulatorze układ w pierwszej fazie pracuje w otwartej pętli regulacji. Początkowo wartość wielkości procesowej doprowadza się do wartości 96% wpisanej wartości zadanej T_{zad_wpis}. Stan ustalony dla podanej fazy występuje jeśli wartość wielkości procesowej mieści w zakresie $0{,}9T_{zad} \div 1{,}1T_{zad}$ przez określony czas.

Po wystąpieniu stanu ustalonego następuje przeprowadzanie eksperymentu (druga faza 340-595 s). W fazie tej wykonuje się skok o wartości dU. Wartość skoku jest odpowiednio dobierana, aby wartości skoku dobrze pokryły otoczenie punktu pracy. Odpowiedź skokowa otrzymywana w tej fazie służy do parametryzacji regulatora DMC. Jeśli proces parametryzacji przebiegnie pomyślnie, układ przejdzie do regulacji w pętli zamkniętej (trzecia faza 595 s).



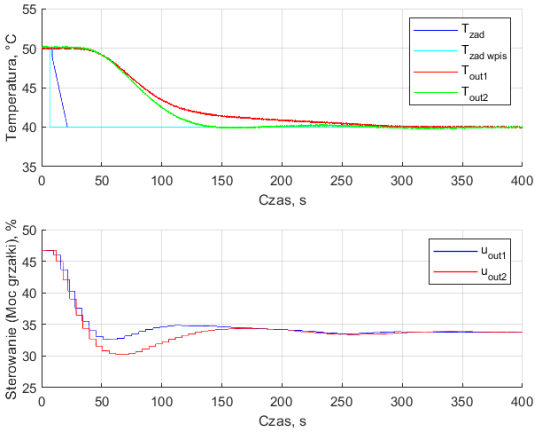
Rys. 6 Przebieg procesu inicjalizacji



Rys. 7 Zmiana punktu pracy

Podczas zmiany wartości zadanej T_{zad_wpis} algorytm na początku sprawdza, czy w bazie danych nie znajdują się już nastawy regulatora dla danego punktu pracy. Jeśli nie istnieją to przeprowadzany jest ponownie proces parametryzacji (Rys. 7). Cały proces przebiega podobnie do jak w przypadku procesu inicjalizacji. Jedyna różnica występuje podczas doprowadzenia wartości procesowej w otoczenie punktu pracy. Regulator pracuje w tym przypadku w trybie nadążania z parametrami wypracowanymi dla poprzedniego punktu pracy.

Na rysunku 7 dodatkowo można zauważyć działanie mechanizmu doboru odpowiedniej wartości skoku. Pierwsza próba nie przebiegła pomyślnie, co spowodowało zwiększenie wartości dU dla następnej próby.



Rys. 8 Przebiegi dla różnych nastaw

Na rysunku 8 przedstawiono sytuację, w której dokonano zmiany wartości wielkości zadanej. Nastawy dla podanego punktu pracy znajdują się w bazie danych, przez co pomijany jest proces parametryzacji. Należy nadmienić, że przepływ podczas przeprowadzania zmiany punktu pracy wynosi 2,2 l/min.

Porównano przebiegi regulacji dla dwóch zestawu nastaw, dla podanej wartości zadanej temperatury. Parametryzacja podanych nastaw została przeprowadzona dla różnych przepływów: T_{out1} - 1,8 l/min, a T_{out2} - 2,2 l/min.

Pomimo, że wczytano nastawy dla odpowiedniego punktu pracy przebiegi różnią się znacząco od siebie. W szczególności różna jest ich dynamika. Różnica pomiędzy osiągnięciem stanu ustalonego dla rozpatrzonych przepływów wynosi, aż 150 sekund. Sterowanie u_{out2} także jest bardziej skuteczne niż u_{out1}.

Z podanych wyników można wywnioskować, iż punkt pracy nie może być charakteryzowany tylko przez samą temperaturę, a przez temperaturę oraz przepływ. Szukanie nastaw musi wtedy uwzględniać obie zmienne. Przy braku nastaw dla danego punktu pracy, można wczytać inne nastawy, znajdujące się w zakresie ± 2°C i ± 0,1 l/min podanego punktu pracy, ponieważ nie wystąpi znacząca różnica dynamiki.

Wnioski

Użycie chmur obliczeniowych umożliwia syntezę skutecznego, wydajnego oraz w miarę taniego algorytmu sterowania obiektem.

Wprowadzenie mechanizmu przechowywania w bazie danych nastaw regulatora dla konkretnych punktów pracy pozwala na znaczącą redukcję przeprowadzanych eksperymentów. Dzięki czemu proces technologiczny jest rzadziej zakłócany.

Mimo przechowywania nastaw należy pamiętać, że zmiana niektórych zmiennych może wpływać znacząco na proces. W takim przypadku użycie nieodpowiednich nastaw może wpłynąć na pogorszenie się jakości regulacji. Aby zapobiec takim sytuacjom, należy umożliwić operatorowi podjęcie decyzji co do korekty nastaw w regulatorze po przez przeprowadzenie parametryzacji weryfikacyjnej.

***Autorzy**: inż. Dominik Bodora, Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki, ul. Akademicka 16, 44-100 Gliwice, E-mail: domibod596@student.polsl.pl; dr hab. inż. Tomasz Kłopot, Politechnika Śląska, Katedra Automatyki i Robotyki, ul. Akademicka 16, 44-100 Gliwice, E-mail: tomasz.klopot@polsl.pl; dr hab. inż. Krzysztof Stebel, Politechnika Śląska, Katedra Automatyki i Robotyki, ul. Akademicka 16, 44-100 Gliwice, E-Mail: krzysztof.stebel@polsl.pl*

LITERATURA

[1] Pereira, A.C.; Romero, F. A review of the meanings and the implications of the Industry 4.0 concept. *Procedia Manufact.*, 13, (2017), 1206–1214.
[2] Gajdzik, B.; Wolniak, R. Influence of Industry 4.0 Projects on Business Operations: Literature and Empirical Pilot Studies Based on Case Studies in Poland. *J. Open Innov. Technol. Mark. Complex.,* 8, (2022), 44
[3] Tjahjono B., Esplugues C., Ares E., Pelaez G., What does Industry 4.0 mean to Supply Chain?, *Procedia Manufact.,* 13 (2017), 1175-1182
[4] Chiarell F., Trivelli L., Bonaccorsi A., Fanroni G., Extracting and mapping Industry 4.0 technologies using Wikipedia, *Computers in Industry.*, 100, (2018), 244-257
[5] Martson S., Li Z., Bandyopadhyay S., Zhang J., Ghalsasi A., Cloud computing – The business perspective, *Decision Support Systems*, 51, (2011), 176-189
[6] Kareem A., Ahmed B., Abdul R., Design and Implementation of Humidity and Temperature Automation and Monitoring for Public Buildings Comfortable Climate Based on Cloud Platform, *Przegląd Elektrotechniczny*, 04, (2022), 94-100
[7] Kazaroan A., Teslyuk V., Tykhan M., Mashevska M., Usage Of SaaS Software Delivery Model In Inteligent House Systems, *Przegląd Elektrotechniczny*, 07, (2019), 38-41
[8] Sawicki B., Krupa A., Optimization of coil geometry using Monte Carlo method HTCondor and Microsoft Azure technologies, *Przegląd Elektrotechniczny*, 05, (2019), 113-118
[9] Czeczot J., Nowak P., Frątczak M., Grelewicz P., ADRC-Based Habituating Control of Double-Heater Source, *Energies*, 15 issue 14, (2022), 1-17
[10] Tatjewski P., Advanced control of industrial processes: structures and algorithms, *London:Springer-Verlag*, (2007)
[11] Kłopot T., Skupin P., Metzger M., Grelewicz P., Tunning strategy for dynamic matrix control with reduced horizons, *ISA Transactions*, 76, (2018), 145-154
[12] Bodora D., Parametryzacja regulatora DMC za pomocą chmury obliczeniowej, *Politechnika Śląska*, (2023)