

doi:10.15199/48.2019.09.41

Czasy realizacji algorytmów PID2DOF w sterownikach serii S7

Streszczenie. Celem artykułu jest analiza szybkości działania algorytmu PID napisanego w języku Structured Control Language (SCL) i porównanie go z dostępnym w TIA Portal rozwiązaniem. Porównanie zostało wykonane na kilku platformach CPU S7 oraz na symulatorze sterowników PLCSIM. Jednocześnie artykuł powinien zachęcić producentów PLC do wdrożenia nowych prostszych rozwiązań algorytmów regulacyjnych oraz sterujących.

Abstract. The aim of the article is to analyze the speed of the PID algorithm written in Structured Control Language (SCL) and compare it with the available solution in the TIA Portal. The comparison was made on several CPU S7 platforms and controllers simulator PLCSIM. At the same time, the article should encourage PLC manufacturers to implement new simpler solutions, ie many available algorithms control algorithms. "The times of PID2DOF algorithms calculation in S7 controller". **Speed of the PID algorithm written in Structured Control Language (SCL)**

Słowa kluczowe: PLC, Siemens S7, PID anti-windup, czas obliczeń algorytmu

Keywords: PLC, Siemens S7, PID anti-windup, time of algorithm calculation

Wstęp

W automatyce przemysłowej mogą wystąpić przypadki, gdzie rozbudowana funkcjonalność algorytmu jest wadą, ponieważ przede wszystkim liczy się czas realizacji kolejnych obliczeń. Stosowanie dostępnych na rynku rozwiązań, m.in. sterowników serii S7 firmy Siemens [1], pozwala na realizację wymagających zadań sterowania w najprostszym sposobie. Celem artykułu jest analiza szybkości działania algorytmu PID napisanego w języku Structured Control Language (SCL) i porównanie go z dostępnym w TIA Portal rozwiązaniem – PID_Compact V2. Czas obliczenia sterowania (sygnał wyjściowy regulatora) przez algorytm regulatora jest przedmiotem analizy.

Ciągły rozwój sterowników przemysłowych (PLC) oraz stosowanie najnowszych rozwiązań technologicznych doprowadza do skracania czasów wykonywania poszczególnych instrukcji arytmetyczno-logicznych na różnych typach zmiennych – zestawienie podstawowych parametrów jednostek centralnych firmy Siemens przedstawiono w tabeli 1.

Układy analogowe typu „wejścia-wyjścia” stanowią pewne ograniczenia dla zastosowania szybkich sterowników PLC w układach sterowania, np. dla napędów elektrycznych. Moduły te są ciągle rozwijane i udoskonalane, przez co w niedługiej przyszłości ich szybkość może dorównać układom CPU. Na chwilę obecną czasy konwersji w modułach AQ i AI są jednak zbyt długie (tab. 2), aby w pełni wykorzystać możliwości najwydajniejszych sterowników PLC.

Najprawdopodobniej za kilkanaście lat specjalizowane sterowniki do silników elektrycznych nie będą już potrzebne, ponieważ moduły uniwersalne PLC uzyskują wystarczającą szybkość.

Oprócz zastosowań przemysłowych sterowników, w literaturze można odszukać prace naukowe dotyczące zagadnień regulacji PID [2-6], a nawet sterownia układami o niepewnych parametrach [7].

W artykule skoncentrowano się na analizie pracy regulatora PID_compact V2, który jest regulatorem o dwóch stopniach swobody (PID2DOF). Szybkość obliczeń tego regulatora zestawiono z napisanym od podstaw algorytmem w języku SCL. W badaniach porównawczych wykorzystano sterowniki: 1517F-3 PN/DP, S7-1515-2 PN, S7-1512, S7-1211C.

Regulacja cyfrowa PID i implementacja w S7

W artykule przyjęto oznaczenia zgodne ze sterownikami S7 firmy Siemens [1] – PID_Compact V2, tj. w jest

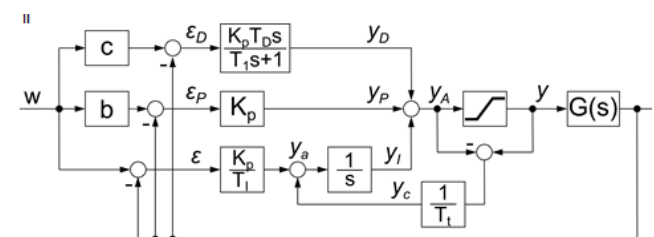
wartością zadaną, y sterowaniem, a x wielkością regulowaną (sygnałem wyjściowym obiektu regulacji).

W celu poprawienia jakości regulacji często wprowadza się ideę różnego przetwarzania sygnału wartości zadanej w oraz sygnału wyjściowego x . Podejście takie prowadzi do częściowego odseparowania zadania nadążania od zadania stabilizacji [8-11], tzn. współczynniki wagowe mają wpływ na odpowiedź skokową x przy wymuszeniu w , ale nie mają wpływu na proces stabilizacji. Dodatkowo przyjmuje się wolnozmiennność wartości zadanej w i modyfikuje formułę sterowania do postaci **regulatora o dwóch stopniach swobody (PID2DOF)**:

$$(1) \quad y(s) = K_R \left(\frac{bw(s) - x(s)}{\varepsilon_p} + \frac{1}{T_i s} \frac{(w(s) - x(s))}{\varepsilon} + \frac{T_d s}{T_i s + 1} \frac{(cw(s) - x(s))}{\varepsilon_d} \right)$$

gdzie b i c są współczynnikami wagowymi dla działania członu proporcjonalnego i różniczkującego. Najczęściej $0 < b \leq 1$ oraz $0 < c \leq 1$.

Kolejnym krokiem polepszającym pracę regulatora PID jest zastosowanie techniki **anti-windup** [10], która polega na ograniczeniu całkowania w regulatorze (rys.1).



Rys. 1. Schemat blokowy układu regulacji z regulatorem typu PID2DOF

Technika ta częściowo poprawia błędy projektanta systemu automatyki, tzn. zły dobór nastaw regulatorów oraz przyspiesza reakcję regulatora pracującego przy nasyceniu sygnału wyjściowego.

W układzie anti-windup wykorzystywane jest dodatkowe sprzężenie zwrotne $1/T_i$ nazywane **torem śledzącym** i w przypadku gdy sygnał y_A przekracza wartość ograniczenia (y_{min} lub y_{max}), na wejście integratora $1/s$ podawany jest dodatkowy sygnał o przeciwnym znaku. Sygnał ten

spowalnia proces całkowania lub całkowicie go blokuje. Przedstawione rozwiązanie polepsza pracę układu poprzez zmniejszenie oscylacji w procesie przejściowym zwłaszcza przy dużych wartościach zadanych. Oznacza to, że parametrami regulatora są: $K_p, T_I, T_D, T_1, T_2, b, c$. Mają one

bezpośredni wpływ na działanie zamkniętego układu regulacji i powinny zostać dobrane w procesie optymalizacji parametrycznej regulatora.

Tabela 1. Podstawowe parametry wybranych sterowników PLC firmy Siemens

Seria	Model	Pamięć prog.	Pamięć zmiennych	Pamięć trwała	Czas operacji na zmiennych			
					bit	word	fixed point	Floating point
S7-1500	6ES7 512-1AK00-0AB0	150 KB	1 MB	128 KB	60 ns	72 ns	96 ns	384 ns
	6ES7 513-1AL02-0AB0	300 KB	1.5 MB	128 KB	40 ns	48 ns	64 ns	256 ns
	6ES7 515-2AM01-0AB0	500 KB	3 MB	512 KB	30 ns	36 ns	48 ns	192 ns
	6ES7 517-3FP00-0AB0	3 MB	8 MB	768 KB	2 ns	3 ns	3 ns	12 ns
S7-1200	6ES7 211-1AD30-0XB0	25 KB	25 KB	2048 B	100 ns	1200 ns	1800 ns	-
	6ES7 214-1AG31-0XB0	75 KB	4 MB	10KB	85 ns	1700 ns	2500 ns	-
	6ES7 214-1AG40-0XB0	100 KB	4 MB	19KB	80 ns	1700 ns	2300 ns	-

Tabela 2. Podstawowe parametry wybranych modułów wejść/wyjść firmy Siemens

Rodzaj	Model	Ilość kanałów	Rozdzielczość przetwornika	Czas konwersji	Błąd liniowy
wejście	6ES7 531-7NF10-0AB0	8	16 Bit	65 500 ns	± 0,02 %
wyjście	6ES7 532-5HF00-0AB0	8	16 Bit	50 000 ns	± 0,02 %
wejście/wyjście	6ES7 534-7QE00-0AB0	4/2	16 Bit	9 000 000 ns/500 000 ns	±0,02%/0,05%

Aby przeprowadzić dyskretyzację regulatora o działaniu ciągłym – w celu uzyskania dyskretnej transformacji $G(z)$, należy za argument s w transmitancji $G(s)$ podstawić s' , gdzie:

$$(2) \quad s' = \frac{z-1}{T_s} \quad \text{metoda Eulera}$$

$$(3) \quad s' = \frac{2}{T_s} \frac{z-1}{z+1} \quad \text{metoda Tustina}$$

stąd $G(z) = G(s')$.

Postać algorytmiczną cyfrowego regulatora uzyskuje się po dokonaniu dyskretyzacji i obliczeniu odwrotnej transformacji Z odpowiednio dla integratora jak i dla różniczkarki rzeczywistej. Wybór metody dyskretyzacji praktycznie nie wpływa na złożoność algorytmu, dlatego można zapisać ogólną postać dla k -tego kroku algorytmu ($\alpha_I, \alpha_D, \beta_I, \beta_D$ - współczynniki zależne od parametrów regulatora, metody oraz czasu próbkowania T_s), odpowiednio dla integratora:

$$(4) \quad y_I(k) = \alpha_I(y_a(k-1) + \beta_I y_a(k)) + y_I(k-1)$$

i dla części różniczkującej:

$$(5) \quad y_D(k) = \alpha_D(\varepsilon_D(k-1) - \varepsilon_D(k)) + \beta_D y_D(k-1)$$

Zestawienie współczynników zamieszczono w tabeli 3.

Tabela 3. Zestawie współczynników algorytmów członów I (4) oraz D (5) dla różnych metod dyskretyzacji

	Współczynniki	Metoda Eulera	Metoda Tustina
I	α_I	T_s	$T_s/2$
	β_I	0	1
D	α_D	$K_p T_D / T_1$	$2K_p T_D / (2T_1 + T_s)$
	β_D	$(T_1 - T_s) / T_1$	$(2T_1 - T_s) / (2T_1 + T_s)$

Zgodnie ze schematem z rysunku 1 oraz algorytmami (4) i (5) napisano procedurę regulacji dla sterownika S7:

$$"eD" := ("c" * "w-in") - "x-in";$$

```
"ep" := ("b" * "w-in") - "x-in";
"e" := "w-in" - "x-in";
"yD" := "alpha D" * ("eD" - "eD-1") + "beta D" * "yD-1";
"yP" := "Kp" * "ep";
"ya_1" := ("Kp" / "T1") * "e" + "yc";
"yl" := "alpha I" * ("ya_1-1" + ("beta I" * "ya_1")) + "yl-1";
"yA" := "yD" + "yP" + "yl";
"y" := "yA";
IF "yA" > "ymax" THEN
  "y" := "ymax";
END_IF;
IF "yA" < "ymin" THEN
  "y" := "ymin";
END_IF;
"yc" := ("y" - "yA") / "Tt";
"eD-1" := "eD";
"yD-1" := "yD";
"y-1" := "y";
"ya_1-1" := "ya_1";
"yl-1" := "yl";
```

gdzie współczynniki $\alpha_I, \alpha_D, \beta_I, \beta_D$ są obliczane przy pierwszym uruchomieniu sterownika PLC:

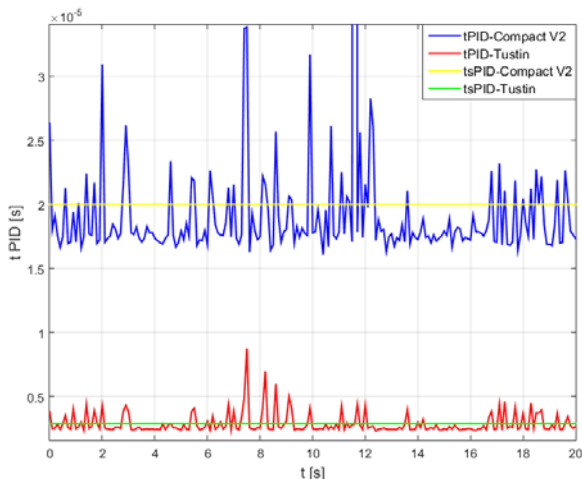
```
IF "FirstScan" = TRUE THEN
  "ymax" := 10.0;
  "ymin" := -10.0;
  "Kp" := 3.0;
  "TD" := 0.4;
  "T1" := 0.25;
  "Ts" := 0.1;
  "Tt" := 1.0;
  "c" := 1.0;
  "b" := 1.0;
  "T1" := 10.0;
  "x-in" := 0.0;
  "w-in" := 0.0;
  "alpha D" := (2 * "Kp" * "TD") / ((2 * "T1") + "Ts");
  "beta D" := ((2 * "T1") - "Ts") / ((2 * "T1") + "Ts");
  "alpha I" := "Ts" / 2;
  "beta I" := 1;
END_IF;
```

Porównanie czasów wykonania algorytmów PID

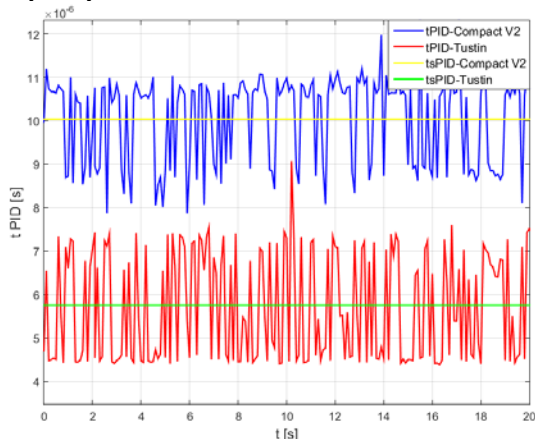
Czas przetwarzania sygnałów firmowego rozwiązania technologicznego regulatora PID firmy Siemens (PID_Compact V2) został porównany z czasem przetwarzania sygnałów regulatora PID wyznaczonego przy

pomocy metody Tustina (PID_Tustin). Do tego celu wykorzystano funkcję RUNTIME sterownika S7, która pozwoliła zmierzyć czas wykonania indywidualnego bloku bądź sekwencji instrukcji.

W pierwszym etapie dokonano porównania z wykorzystaniem symulatora S7-PLCSIM, a otrzymane wyniki zostały przedstawione na rysunku 2. Czas próbkowania regulatorów w każdym z przedstawionych przypadków wynosił $T_s=0.1s$. Na wykresach przyjęto następujące oznaczenia tPID – czas wykonania algorytmu w bieżącej chwili, tsPID – średni czas wykonania algorytmu dla 200. wywołań algorytmu.



Rys. 2. Czasy wykonywania algorytmów regulatora PID podczas symulacji na symulatorze S7-PLCSIM



Rys. 3. Czasy wykonywania algorytmów regulatora PID na sterowniku z CPU1517

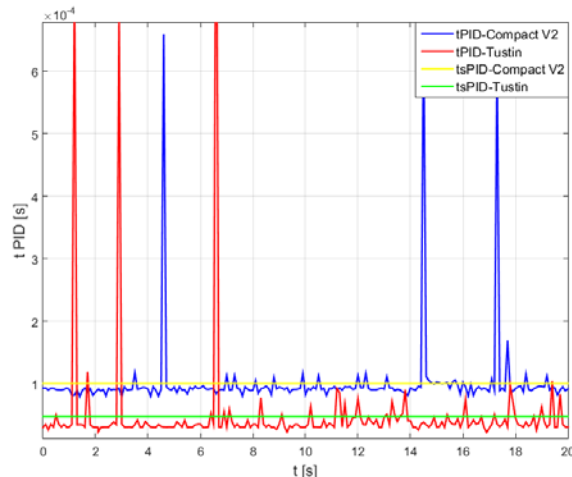
Dodatkowo, na tle przeprowadzonych symulacji zamieszczono średnie wartości czasów przetwarzania danych dla obu algorytmów. Czas przetwarzania firmowego regulatora (PID_Compact V2) wynosił $2 \cdot 10^{-5}$ sekundy, zaś dla zaproponowanej metody Tustina $2,91 \cdot 10^{-6}$ sekundy. Oznacza to że, drugi z algorytmów (PID_Tustin) jest około siedmiokrotnie szybszy od firmowego rozwiązania. Różnica ta wynika ze znacznego rozbudowania algorytmu PID_Compact V2 [1], którego pełne możliwości często są nie wykorzystywane. Znacząca nierównomierność czasu wykonywania obliczeń prawdopodobnie spowodowana jest pracą symulatora w systemie WINDOWS.

W drugim etapie badań symulowane algorytmy zostały wgrane kolejno do sterowników S7-1517F-3 PN/DP, S7-1515-2 PN, S7-1512, S7-1211C, na których dokonano porównania czasów wykonania obu procedur.

Na rysunku 3 przedstawiono wyniki pomiaru czasów realizacji algorytmów dla sterownika z CPU1517. Można

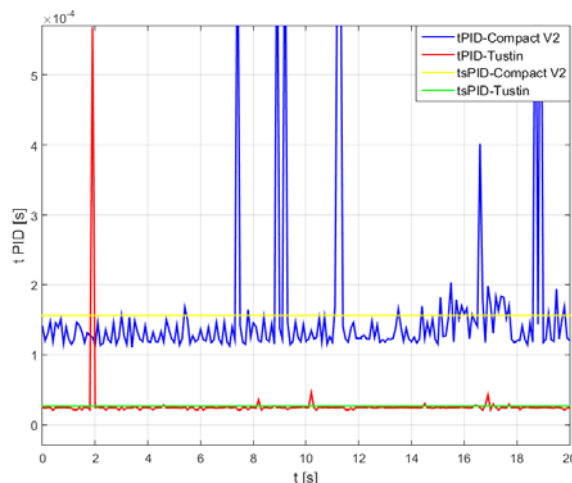
zauważyć, że średnie czasy obliczeń są krótsze w porównaniu z przeprowadzoną symulacją (rys.2), jednak w tym przypadku zdyskretyzowany przy pomocy metody Tustina regulator okazał się 1,75 razy szybszy w porównaniu do regulatora firmowego PID_Compact V2.

Dla sterownika z jednostką CPU1515 wyniki przedstawiono na rysunku 4. W tym przypadku algorytm regulacji oparty o metodę Tustina był średnio 2,1 razy szybszy od rozwiązania PID_Compact V2.



Rys. 4. Czasy wykonywania algorytmów regulatora PID na sterowniku z CPU1515

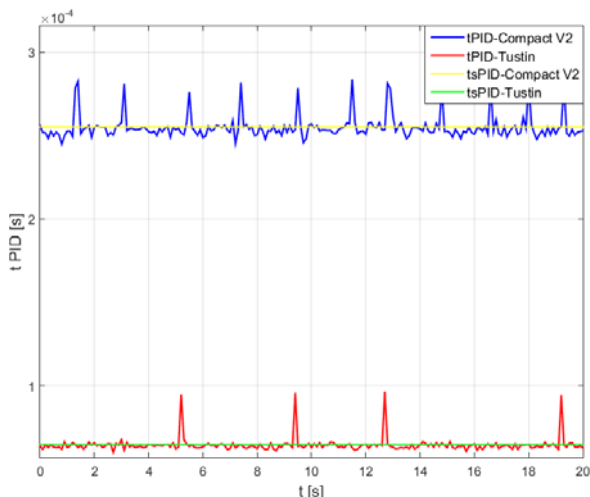
W podobny sposób przeprowadzono badania dla sterownika z procesorem CPU1512 (rys. 5), gdzie w porównaniu do algorytmu PID_Compact V2, algorytm PID_Tustin był niemal sześciokrotnie szybszy.



Rys. 5. Czasy wykonywania algorytmów regulatora PID na sterowniku z CPU1512

Ostatni z analizowanych sterowników wyposażony był w jednostkę CPU1211 (rys.6). W tym przypadku przetwarzanie sygnałów było niespełna 4. krotnie wolniejsze dla algorytmu Siemens - PID_Compact V2.

Podczas pomiaru czasów wykonania algorytmów regulacji, sterowniki były połączone z oprogramowaniem TIA Portal w trybie Online. Takie połączenie wymusza komunikację w przerwaniu pomiędzy urządzeniami. Skutkuje to pojawieniem się na przedstawionych wykresach pojedynczych skoków wartości czasu obliczeń



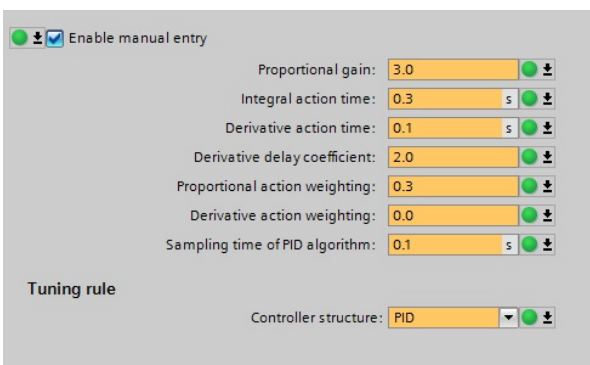
Rys. 6. Czasy wykonywania algorytmów regulatora PID na sterowniku z CPU1211

Weryfikacja poprawności działania algorytmów PID

W celu sprawdzenia poprawności działania przedstawionych regulatorów PID2DOF na obiekcie rzeczywistym, wykonano badania na dwóch niezależnych obwodach elektrycznych, zbudowanych z elementów pasywnych RC ($R=100k\Omega$, $C=1\mu F$) o transmitancji:

$$G(s) = \frac{1}{0,1s + 1}$$

Powyższa transmitancja jest przykładem elementu inercyjnego 1. rzędu, gdzie $T=RC$ jest stałą czasową obwodu, natomiast jego wzmocnienie wynosiło 1. Do badań wykorzystano sterownik PLC z jednostką CPU1513, który współpracował z modułem analogowym AI 4xU//RTD/TC/ AQ 2xU// ST. Pomiaru wykonano dla dwóch wartości czasu próbkowania ($T_s=0,1s$ i $T_s=0,05s$).

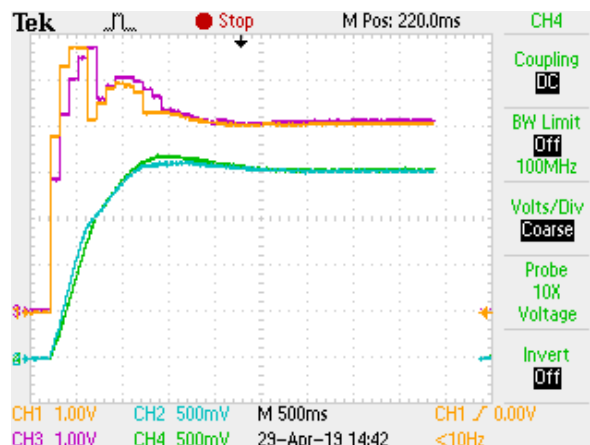


Rys. 7. Parametry regulatora PID_Compact V2

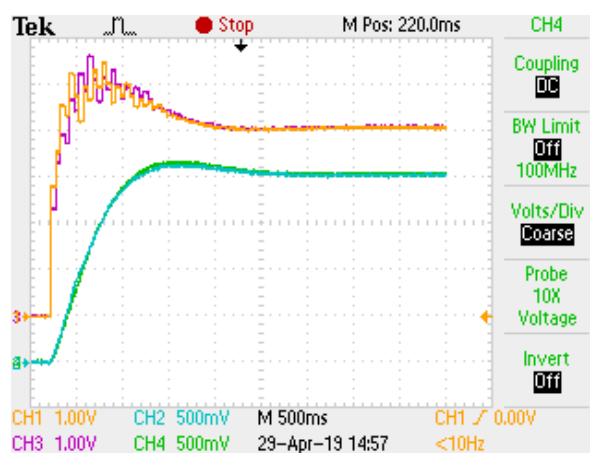
Pomiarów dokonano z wykorzystaniem oscyloskopu Tektronix TDS 2014B, gdzie do kanałów CH1, CH2 podłączono sterowanie i pomiar odpowiedzi dla algorytmu PID_Compact V2. Analogicznie, algorytm sterowania z wykorzystaniem metody Tustina jest monitorowany na kanałach CH3 i CH4. Zarejestrowane wyniki przedstawiono na oscylogramach (Rys.8, Rys.9). Natomiast parametry regulatora, które zostały wpisane do bloku PID_Compact V2 (Rys.7) wprowadzono również do zaproponowanego algorytmu PID_Tustin.

Działanie przedstawionych algorytmów można ocenić przy pomocy bezpośrednich wskaźników jakości, gdzie dla czasu próbkowania $T_s=0,05s$ czas odpowiedzi i przeregulowanie są niemalże identyczne dla obu regulatorów. Natomiast dla czasu próbkowania $T_s=0,1s$

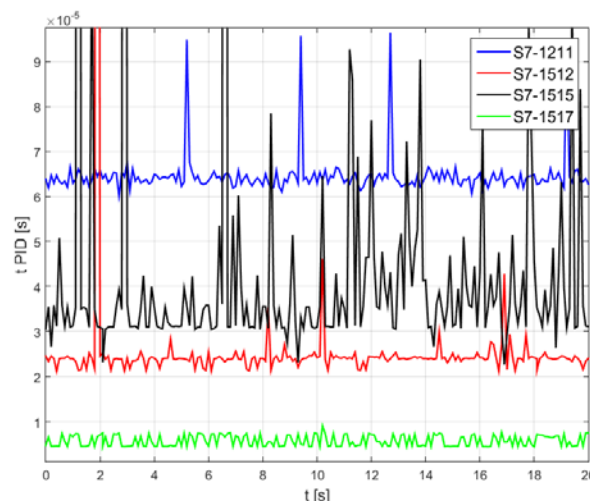
różnica pomiędzy oboma czasami odpowiedzi wynosi około 2%, a przeregulowania różnią się o 5%.



Rys. 8. Zestawienie sygnałów sterujących i odpowiedzi skokowych dla czasu próbkowania $T_s=0,1s$



Rys. 9. Zestawienie sygnałów sterujących i odpowiedzi skokowych dla czasu próbkowania $T_s=0,05s$.



Rys. 10. Czasy wykonania algorytmu PID_Tustin na wybranych jednostkach sterowników PLC serii S7

Najprawdopodobniej przyczyną tych niewielkich różnic jest inna implementacja (dyskretyzacja) algorytmu PID. Oznacza to, że oba algorytmy zbliżone są do siebie w działaniu, zatem można porównywać czasy obliczeń – czas wyznaczenia sterowania na podstawie pomiaru wartości zadanej.

Podsumowanie

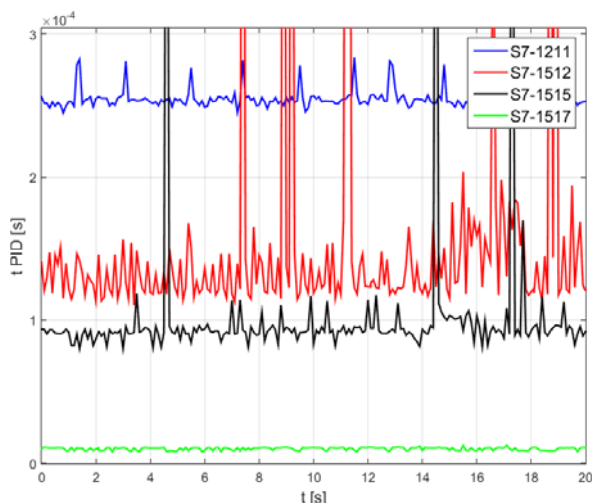
Zestawienie wyników z przeprowadzonych badań dla różnych jednostek sterowników PLC przedstawiono na rysunku 10 (czasy wykonania kolejnych wywołań bloku FC - PID_Tustin) i rysunku 11 (czasy wykonania kolejnych kroków algorytmu PID_Compact V2).

Srednie wartości czasów wykonania jednego kroku algorytmu PID2DOF zamieszczono w tabeli 4.

Tabela 4. Zestawienie wartości czasów wykonania jednego kroku algorytmu PID2DOF

	PID_Tustin	PID_Compact V2	$\frac{\text{PID_Tustin}}{\text{PID_Compact}}$
CPU1517	5,7 μs	10 μs	1,75
CPU1515	48 μs	100 μs	2,1
CPU1512	26 μs	160 μs	5,9
CPU1211	65 μs	250 μs	3,9

Widać, że czas obliczeń zaproponowanego algorytmu (PID_Tustin) dla każdego sterownika był krótszy w porównaniu z firmowym rozwiązaniem dla S7. Dodatkowo algorytm ten zaimplementowany na jednostce CPU1211 (najprostszy model) jest jedynie wolniejszy od najbardziej zaawansowanej jednostki z CPU1517, na której uruchomiono algorytm PID_Compact V2.



Rys. 11. Czasy wykonania algorytmu PID_Compact V2 na wybranych jednostkach sterowników PLC serii S7

Jak zasugerowano we wstępie artykułu, możliwe jest uzyskanie zdecydowanie krótszych czasów obliczeń

algorytmów PID napisanych samodzielnie w stosunku do rozwiązania firmowego Siemens - PID_Compact V2.

Podziękowania: Składamy serdeczne podziękowania F.H.U. Auspol za udzielone wsparcie techniczne w postaci dostarczenia programowalnych sterowników PLC oraz za wypożyczenie programatora wraz z pakietem oprogramowania TIA Portal v15. Adres sieciowy firmy: <http://www.auspol.com.pl>

Autorzy: mgr inż. Józef Gromba, e-mail: jozefgromba@gmail.com, dr inż. Grzegorz Sieklucki, e-mail: siekl@agh.edu.pl; mgr inż. Sylwester Sobieraj, e-mail: sobieraj2@gmail.com; AGH w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, Katedra Energoelektroniki i Automatyki Systemów Przetwarzania Energii, al. Adama Mickiewicza 30, 30-059 Kraków.

LITERATURA

- [1] SIEMENS, SIMATIC S7-1200, S7-1500 PID control, Function Manual, 10/2018, 74-97, 252-287.
- [2] Rooholahi B., Reddy P.L., "Concept and application of PID control and implementation of continuous PID controller in Siemens, Indian Journal of Science and Technology, Vol. 8(35): 1-9, December 2015
- [3] Grecu D.-L., Popescu S.-L., Hard/Soft simulator for electric engines controlled with Siemens automates, ECAI International Conference, 109-116, 2015
- [4] Li Y., Tang C., Liu K., PID parameter self-setting method base on S7-1200 PLC, International Conference on Electrical and Control Engineering, ICECE, 274-277, 2011.
- [5] Sałat R., Winiczenko R., Awtoniuk M., Dobór nastaw regulatora PID zaimplementowanego w sterowniku PLC przy wykorzystaniu algorytmów genetycznych, PAK, 58(2012), nr 5, 427-430.
- [6] Oprzędkiewicz K., Programowe testy spełnienia wymagań czasu rzeczywistego na sterowniku PLC z wykorzystaniem języka LD, PAR 3/2011, 71-75.
- [7] Oprzędkiewicz K., Implementacja algorytmów sterowania systemami o niepewnych parametrach na platformach PLC, Automatyka, Tom 13(2): 531-538, 2009.
- [8] Åström K.J., Hägglund T., PID Controllers: Theory, Design, and Tuning, International Society for Measurement and Control, 1995.
- [9] Araki M., Taguchi H., Two-Degree-of-Freedom PID Controllers, IJCAS, Vol. 1(4):401-411, 2003.
- [10] Åström K.J., Hägglund T., Advanced PID Control, ISA-The Instrumentation, Systems, and Automation Society, 2006.
- [11] Guzman J.L., Åström K.J., Dormido S., Hägglund T., Berenguel M., Piguet Y., Interactive learning modules for pid control, Control Systems, IEEE, 28(5): 118-134, 2008