

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP

Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

ZESPÓŁ AUTOMATYKI ELEKTRONICZNEJ

440

PRACOWNIA REGULATORÓW ELEKTRONICZNYCH

BE 10

Główny wykonawca

mgr inż. Z. Pietrusiński

Pietrusiński

Wykonawcy

mgr inż. G. Kazimiński

mgr inż. A. Cichy

Konsultant

Nr zlecenia

S-1307

Badania symulacyjne algorytmów autostrojenia regulatorów.

Etap 2. Przeprowadzenie badań symulacyjnych algorytmów autostrojenia, weryfikacja programów.

Zleceniodawca

Praca statutowa PIAP

Pracę rozpoczęto dnia

1992.10.16

zakończono dnia 1992.12.15.

Kierownik Pracowni

Z-ca Dyr. d/s

Kierownik Zespołu

Pietrusiński

Bad. Rozwojowych

J. Korytkowski

mgr inż. Z. Pietrusiński

dr inż. J. Jankowski

doc. dr inż. J. Korytkowski

Praca zawiera:

Rozdzielnik - ilość egz:

64 stron

Egz. 1 BOINTE

27 rysunków

Egz. 2 ZAE

fotografii

Egz. 3 ZAE-2

5 tabel

Egz. 4 ZAE-2

tablic

Egz. 5

załączników

Egz. 6

Nr rejestr.

6902

Analiza deskrytorowa

REGULATORY ELEKTRONICZNE + ALGORYTMY + OPROGRAMOWANIE + ADAPTA-
CJA : BADANIA

Analiza dokumentacyjna

Praca zawiera wyniki badań symulacyjnych algorytmów autostrojzenia regulatorów ciągłych PID. Badania wykonano w oparciu o programy przygotowane w poprzednim etapie pracy.

Wyniki badań pozwoliły na dokonanie odpowiednich korekt i na weryfikację programów wyjściowych.

Tytuły poprzednich sprawozdań

1. "Algorytmy autostrojzenia regulatorów i ich implementacja do oprogramowania mikroprocesorowych regulatorów ciągłych wolnozmiennych procesów przemysłowych". - Sprawozdanie z pracy PIAP Nr rejestr.6851.
2. "Badania symulacyjne algorytmów autostrojzenia regulatorów. Etap 1. Opracowanie programów do badań". - Sprawozdanie z pracy PIAP Nr rejestr.6882.

SPIS TREŚCI

1. Wstęp.
2. Ogólna charakterystyka badań symulacyjnych.
3. Omówienie badań autostrojania jednorazowego.
4. Omówienie badań autostrojania ciągłego.
5. Poprawione programy algorytmów autostrojania.
6. Wnioski.

1. WSTEP

Badania symulacyjne algorytmów autostrojenia zostały przeprowadzone w oparciu o wykonane w poprzednim etapie oprogramowanie. Oba etapy stanowią kontynuację tematu "algorytmy autostrojenia regulatorów i ich implementacja do oprogramowania mikroprocesorowych regulatorów ciągłych wolnozmiennych procesów przemysłowych.

Obecnie autostrojenie i adaptacja stały się już niemal powszechne w nowych opracowaniach mikroprocesorowych regulatorów procesów wolnozmiennych. Jednak rozwiązania stosowane w poszczególnych aparatach znacznie różnią się w szczegółach, które często decydują o jakości algorytmu.

Prace prowadzone w ZAE dotyczą dwu sposobów autostrojenia :

- autostrojenia opartego na metodzie cyklu granicznego i odpowiedzi skokowej;
- autostrojenia dokładnego, opartego na własnych przemyśleniach, obejmującego strojenie regulatora zarówno w wypadku zmieniających się parametrów obiektu jak i zmieniających się zakłóceń oddziałujących na proces regulacji.

Prace prowadzone w niniejszym etapie miały na celu :

- zbadanie poprawności zaproponowanej koncepcji realizacji algorytmu autostrojenia;
- stwierdzenie prawidłowości działania opracowanych programów autostrojenia;
- zbadanie wpływu czynników zewnętrznych takich jak : sygnały zakłócające na wejściu obiektu, ograniczenie sygnału, parametry obiektu itp. na pracę układów autostrojenia;
- wprowadzenie odpowiednich poprawek i korekt oraz zweryfikowanie programów opracowanych w poprzednim etapie pracy.

2. OGÓLNA CHARAKTERYSTYKA BADAŃ SYMULACYJNYCH

Badania symulacyjne zostały przeprowadzone na komputerze PC-386 w oparciu o programy napisane w języku C. Programy te umożliwiały symulację regulatora, obiektu oraz zakłóceń na niego oddziałujących.

Uproszczony schemat układu regulacji z zastosowaniem regulatora samonastrajalnego przedstawiono na rys.1.

Zgodnie z założeniami przyjęto obiekt o transmitancji typu

$$\frac{K_{ob}}{1 + sT_F} e^{-sT}$$

Program algorytmu regulatora ciągłego umożliwiał nastawę, a także przestrajanie wszystkich istotnych parametrów tj. :

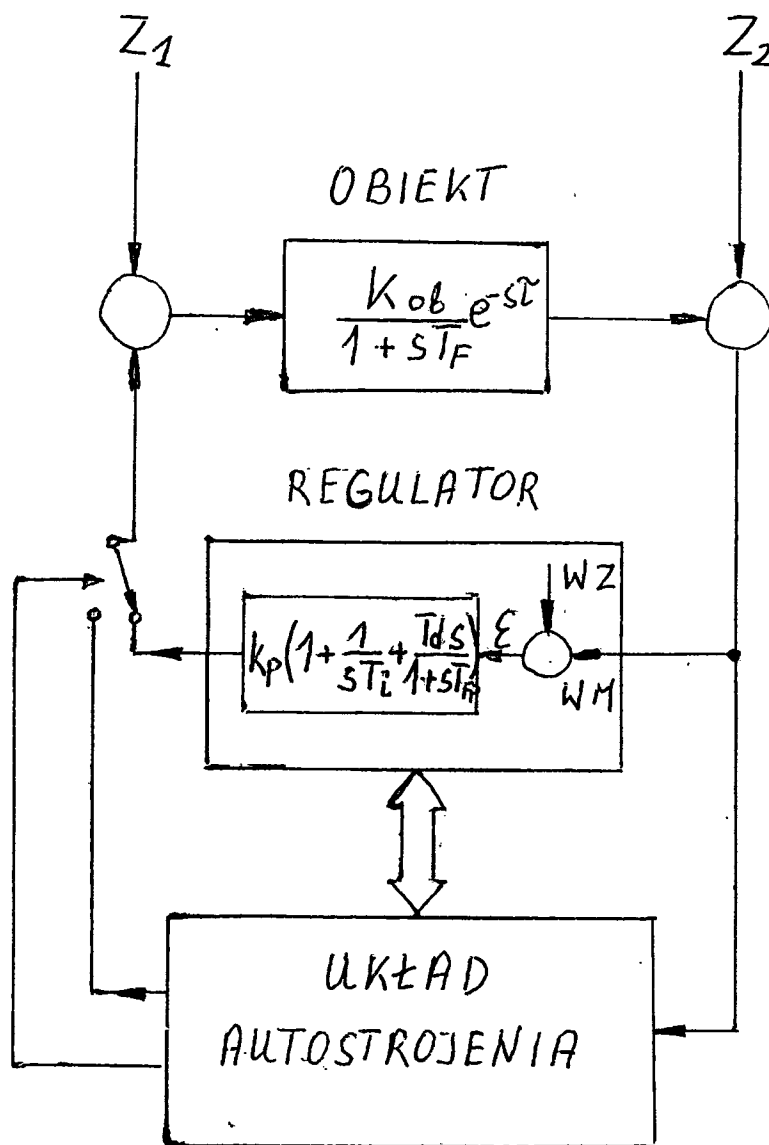
- wartości zadanej WZ,
- wzmocnienia proporcjonalnego K_p ,
- stałej czasowej całkowania T_I ,
- stałej czasowej różniczkowania T_D ,
- stałej czasowej filtracji sygnału różniczki T_{FP} .

Wyniki badań były zapisywane do zbioru dyskowego, po każdym kolejnym eksperymencie. Oprogramowanie zapewniało możliwość prezentacji wyników na ekranie bądź drukarce. Istnieje możliwość prezentacji wszystkich charakterystycznych sygnałów w układzie regulacji takich jak : sygnał wejściowy, odchyłka regulacji, sygnał regulowany, a także wartości parametrów regulacji przed i po autostrojeniu, wskaźnik jakości regulacji, wybrane charakterystyczne sygnały układu autostrojenia.

Badaniom podlegały dwie procedury autostrojenia : strojenie zgrubne oraz strojenie precyzyjne.

Algorytm strojenia zgrubnego jest dość dobrze znany od strony teoretycznej i stosowany z większymi lub mniejszymi zmianami, w wielu produkowanych regulatorach mikroprocesorowych. Zgodnie z oczekiwaniami badania potwierdziły przydatność tego algorytmu do zastosowań przemysłowych. Zaproponowany algorytm strojenia ^{dokładnego} jest rozwiązaniem oryginalnym i jakkolwiek badania wskazują na poprawność założeń teoretycznych, to jednak zastosowanie praktyczne

wymaga jeszcze przewyciężenia szeregu trudności dla uzyskania pewności pracy w różnych warunkach zewnętrznych.



Rys.1. Uproszczony schemat układu regulacji z zastosowaniem regulatora z algorytmem autostrojzenia.

3. OMÓWIENIE BADAŃ AUTOSTROJENIA JEDNORAZOWEGO.

Podstawą teoretyczną strojenia zgrubnego jest zasada Zieglera-Nicholsa, której istota sprowadza się do następującego praktycznego przepisu :

aby otrzymać wartości nastaw, dla kalsycznego regulatora PID, włącza się w pętle ujemnego sprzężenia zwrotnego tylko część proporcjonalną regulatora i zmieniając jego wzmocnienie doprowadza układ zamknięty : obiekt-regulator do drgań niegasnących. Okres drgań odpowiada punktowi przecięcia charakterystyki amplitudowo-fazowej obiektu z ujemną częścią osi rzeczywistej. Na podstawie odczytanego wzmocnienia regulatora K_{kr} i zmierzonego okresu drgań T_{kr} oblicza się nastawy regulatora np. według zależności Zieglera-Nicholsa. Doprowadzenie układu do drgań niegasnących jest czynnością żmudną i czasem wcale nie tak prostą. Proces ten można automatyzować w opisany niżej sposób.

Włączając w pętle ujemnego sprzężenia zwrotnego przekaźnik dwupołożeniowy zamiast regulatora, otrzymuje się w układzie drgania nietłumione.

Amplituda i okres tych drgań związane są ściśle z tzw. punktem krytycznym, tzn. punktem na charakterystyce amplitudowo - fazowej, w którym przecina się ona z ujemną częścią osi rzeczywistej. Okres drgań jest równy okresowi drgań w punkcie krytycznym T_{kr} . Wzmocnienie krytyczne K_{kr} da się otrzymać wykorzystując metodę funkcji opisującej.

Warunek drgań niegasnących określa zależność

$$1 + K_o(jw_{kr})K_{kr} = 0$$

gdzie:

$K_o(jw_{kr})$ - transmitancja obiektu dla częstotliwości drgań niegasnących w_{kr} .

Korzystając z metody funkcji opisującej dla przekaźnika bez histerezy o amplitudzie przełączeń B przy amplitudzie drgań sinusoidalnych na jego wejściu wynoszącej A otrzymujemy :

$$K_{kr} = \frac{4B}{\pi A}$$

Jeśli drgania sygnału na wyjściu obiektu nie mają kształtu sinusoidalnego, to dla polepszenia dokładności obliczeń należy wartość A obliczyć posługując się filtrem pasmowo przepustowym.

Podstawowe wyniki badań przedstawione są w formie wykresów na rysunkach rys.2 do rys.25. Podano na nich przebiegi sygnałów :

- sygnału regulowanego X_M ,
- sygnału odchyłki regulacji ε oraz
- sygnału wyjściowego regulatora

przed rozpoczęciem autostrojenia, w trakcie procesu autostrojania oraz po dostrojeniu. Na osi odciętej naniesiona jest skala czasowa w sekundach, przy czym próbkowanie odbywa się z okresem równym jednej sekundzie.

Początkowa część przebiegu odpowiada ustaleniu się sygnałów bezpośrednio po włączeniu regulatora co wiąże się ze skokiem wartości zadanej i odpowiadającym mu przebiegiem przejściowym. Następnie pokazana jest odpowiedź układu regulacji na skokowe 10-procentowe zakłócenie na wejściu obiektu, przed przystąpieniem do procesu autostrojania. Po ustaleniu się przebiegu przejściowego rozpoczyna się proces autostrojania tzn. zostają wymuszone nietłumione drgania prostokątne wokół poprzednio ustalonego punktu pracy. Proces autostrojania kończy się kiedy pomierzone 3 kolejne amplitudy wahań sygnału regulowanego nie różnią się od siebie więcej niż o zadaną wartość Δ (w badaniach przyjmowano wartość Δ w granicach 0,01.....0,5%).

Dalsza część przebiegu to przebieg przejściowy związany z powrotem sygnału wyjściowego regulatora do stanu sprzed procesu autostrojania, przy czym regulator posiada już nastawy wyliczone w oparciu o przeprowadzony proces autostrojania. Dla zmniejszenia zakłócenia w stanie przejściowym sygnały z bloku proporcjonalnego i różniczkującego są zerowane w momencie zakończenia procesu autostrojania.

Ostatnia faza przebiegu przedstawia odpowiedź układu regulacji,

zawierającego regulator z dobranymi w procesie autostrojania nastawami parametrów dynamicznych na 10 procentowe zakłócenie skokowe na wejściu obiektu. Pozwala to na porównanie przebiegów przed i po procesie autostrojania.

Przebiegi przedstawione na rys.2 do 25 pozwalają ponadto na :

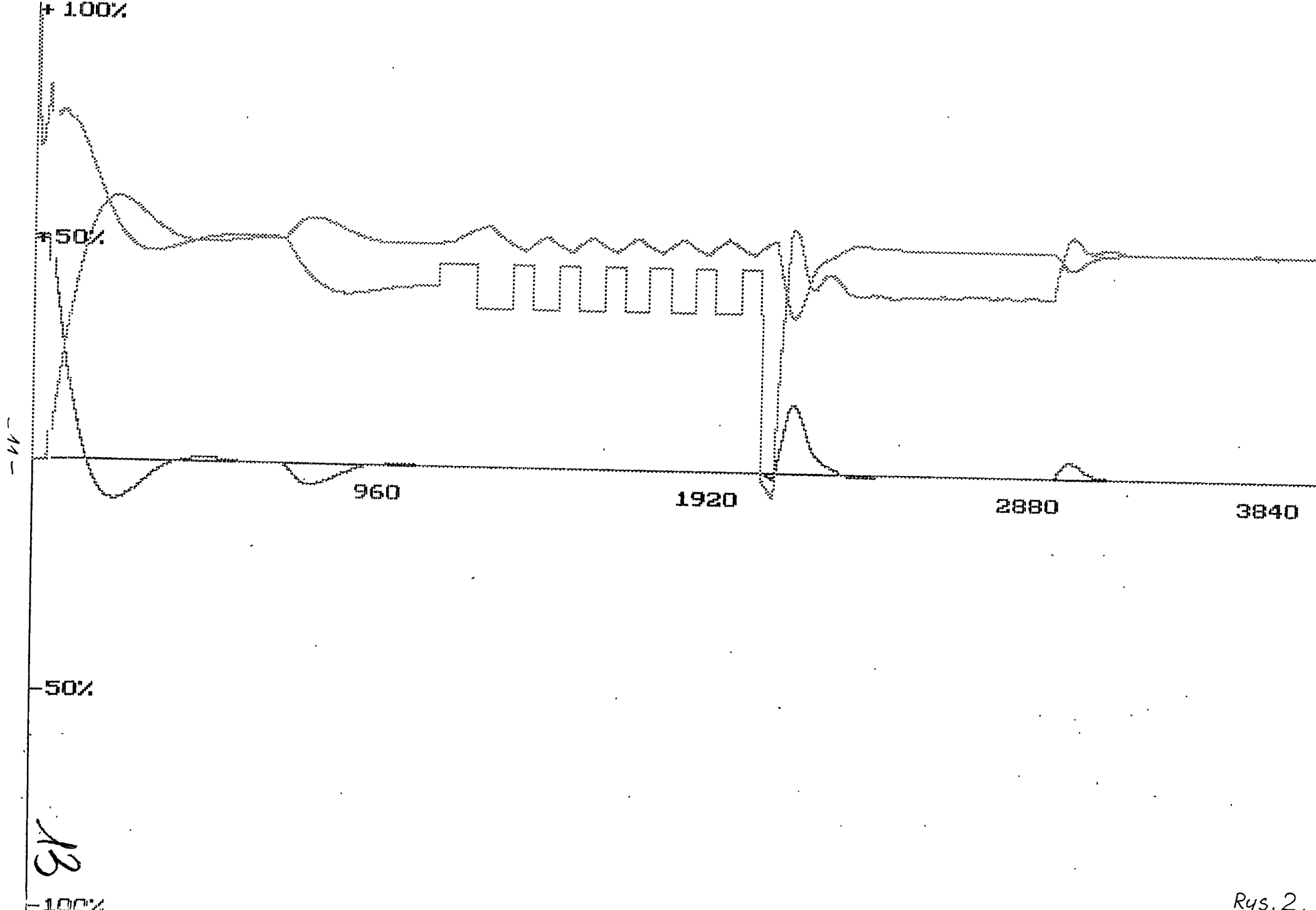
- porównanie jakości autostrojania przy obliczeniach wykonanych w obecności filtra dolno przepustowego (rys.4 do 25) oraz jakości autostrojania bez filtra (przebiegi na rys.2 i 3);
- sprawdzenie skuteczności autostrojania przy różnych charakterystykach obiektu regulacji tzn. przy różnym stosunku opóźnienia T_o do stałej czasowej inercji obiektu T_f (rys.4 do 7);
- określenie wpływu poziomu zakłóceń stochastycznych oraz wymaganej dokładności porównania amplitud przebiegów, przy drganiach niegasnących w procesie autostrojania, na czas procesu autostrojania oraz na dokładność identyfikacji (rys.8 do 13);
- zbadanie wpływu amplitudy wymuszenia przekąźnikowego przy drganiach niegasnących w procesie autostrojania na dokładność i czas autostrojania w obecności stałych zakłóceń stochastycznych (rys.15 do 20);
- zbadanie możliwości wykorzystania autostrojania zgrubnego do doboru nastaw regulatora wg różnych wskaźników regulacyjnych (rys.21 do 23);
- stwierdzenie możliwości i jakości autostrojania przy jednoczesnym spełnieniu dodatkowego warunku odnośnie nieprzekraczania przez sygnał regulowany ostatniej wartości przed uruchomieniem procesu autostrojania (rys.24 i 25).

Zestawienie najważniejszych wyników badań podano w Tabeli 1. Badania dostarczyły szeregu istotnych informacji, które omówione są poniżej.

Stwierdzono między innymi, że :

- Stosowanie pasmoprzepustowego filtra, przeznaczonego dla filtracji sygnału regulowanego w procesie autostrojania jest uzasadnione, gdyż wyraźnie podnosi dokładność obliczeń.
- Metoda autostrojania daje dobre wyniki w pełnym badanym zakresie charakterystyk obiektu regulacji.
- Dokładność identyfikacji ustalenia się przebiegu drgań niegasnących nie musi być zbyt wysoka, gdyż w granicach 0,1...0,5% nie obserwuje się jeszcze wyraźnej zmiany w dostrojeniu parametrów regulacyjnych. Z drugiej strony zbyt duża dokładność powoduje znaczne wydłużenia czasu identyfikacji. Przy dużych zakłóceniach stochastycznych oddziaływujących na obiekt może się nawet zdarzyć, że algorytm nie potrafi zakończyć procesu autostrojania z uwagi na zbyt dużą wymaganą dokładność porównania amplitud drgań niegasnących.
- Wartość amplitudy wymuszenia przekąźnikowego, po przekroczeniu pewnej minimalnej strefy, zależnej od zakłóceń zewnętrznych, nie ma już większego wpływu na dokładność i czas autostrojania. Wydaje się, że amplituda rzędu 5 - 10% jest optymalna, gdyż gwarantuje dobrą dokładność, a jednocześnie sprawia, że zakłócenia pracy obwodu regulacji wprowadzane przez proces autostrojania są stosunkowo nieduże.
- Stosując powyższą metodę można uzyskać dobór nastaw wg różnych kryteriów doboru. Można zarówno zapewnić nastawy dla uzyskania przebiegów o tłumieniu aperiodycznym krytycznym jak i nastawy odpowiadające przebiegom silnie oscylacyjnym, drogą prostego przeliczenia zalecanych wartości nastaw przy różnych wskaźnikach jakości regulacji. Algorytm może więc zapewnić użytkownikowi możliwość wyboru sposobu samodostrajania się regulatora zgodnie z potrzebami procesu regulacji.
- Poprawne wyniki autostrojania można również uzyskać doprowadzając obwód regulacji do drgań nietłumionych, nie koniecznie względem aktualnego punktu pracy obwodu lecz powyżej lub poni-

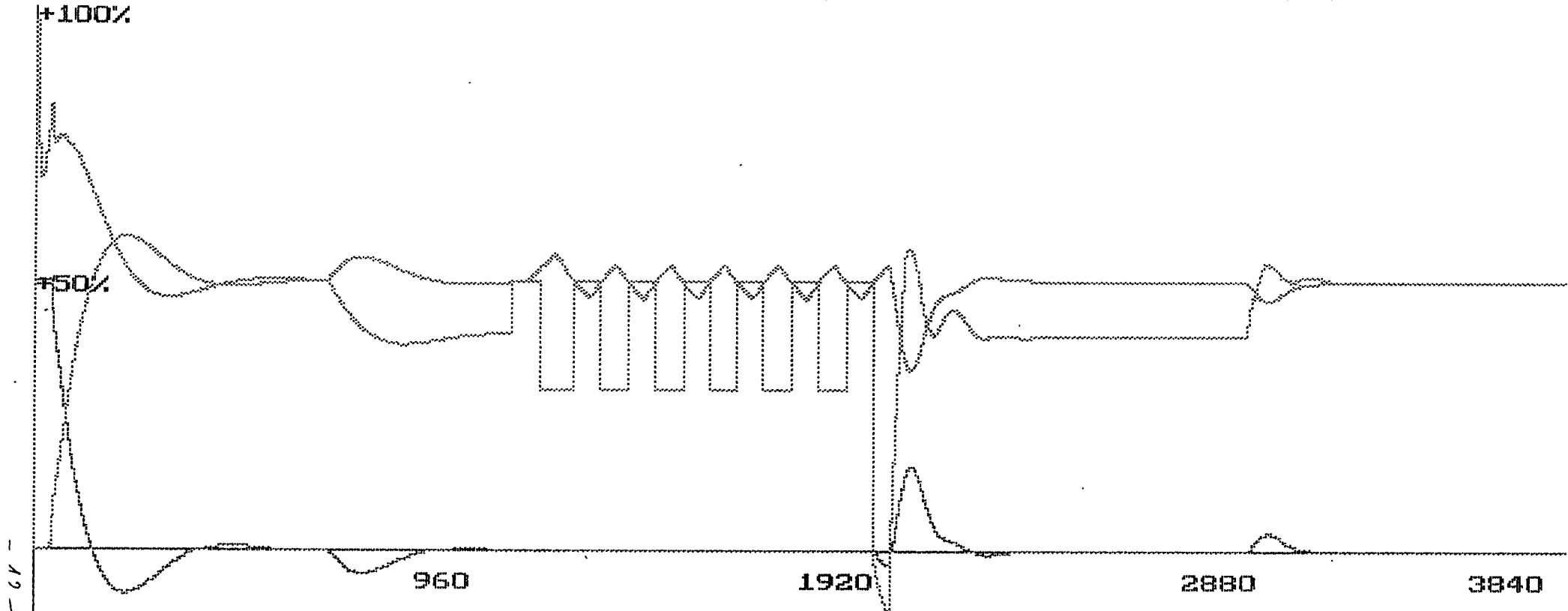
żej punktu pracy. Dla niektórych procesów technologicznych nie dopuszczających przekroczeń określonych granic przez wielkość regulowaną ma to istotne znaczenie.



13

Rys. 2.

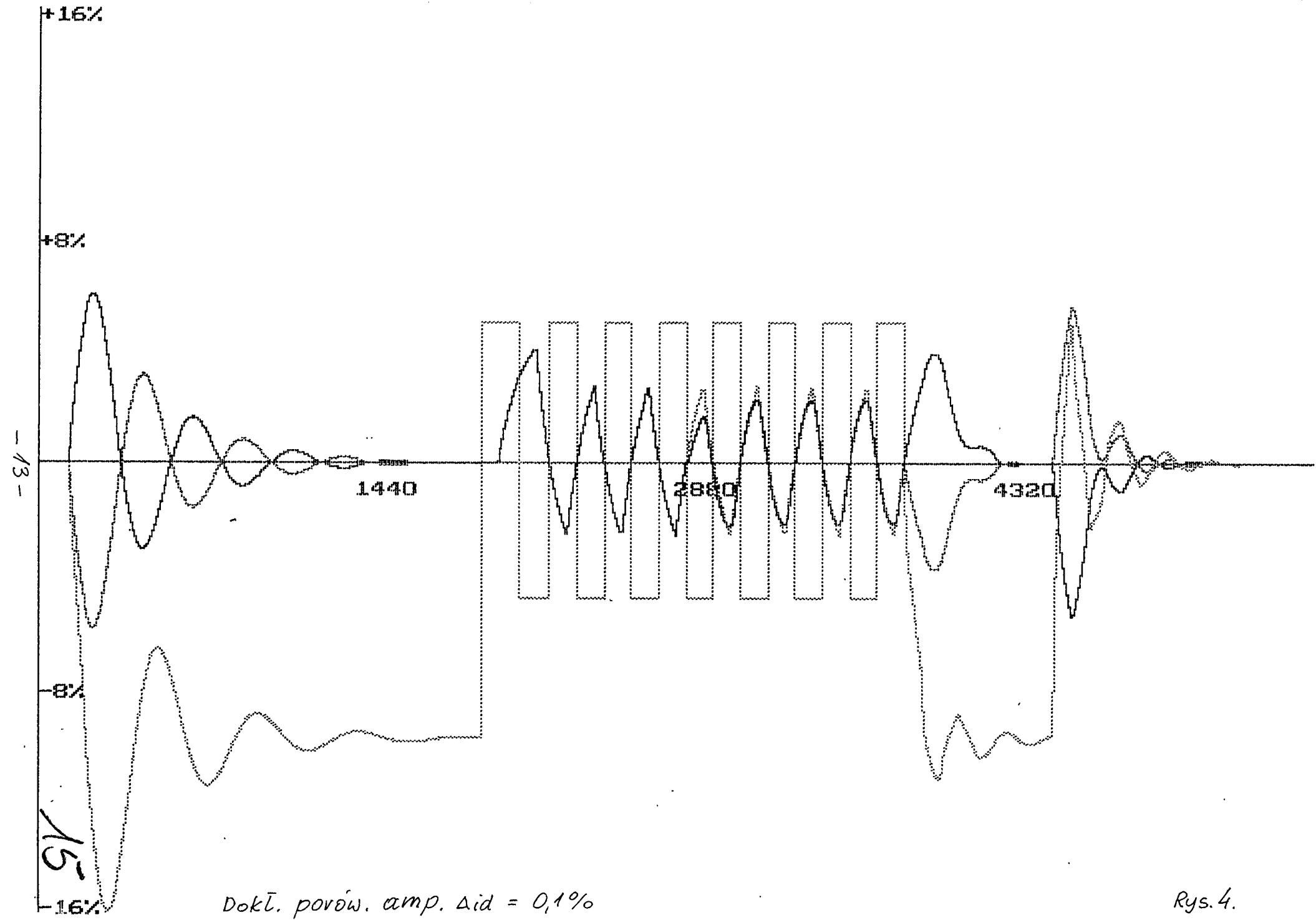
Regulator: $K_p=1.00/2.00$; $T_i=62.00/65.00$; $T_d=10.00/17.00$; $\text{Człekt. } T_c=40.00, T_f=100.00$
 Sygnaly: $S_p=50.00\%$; $Zakłocenia=10.00\%; 0.00\%$; $\text{Amplituda sygn. ident}=5.00\%$



14

Dokł. porów. amplitud $\Delta id = 0,1\%$; Amp. zakł. stoch. = 2%
 Regulator: $Kp=1,00/2,95, Ti=50,00/68,00; Td=10,00/17,00, Dp=0,00$; $Tow=40,00, Tf=100,00$
 Sygnaly: $\%s=50,00\%$; Zakłocenia= 10,00%; 0,00%; Amplituda sygn. ident=10,00%

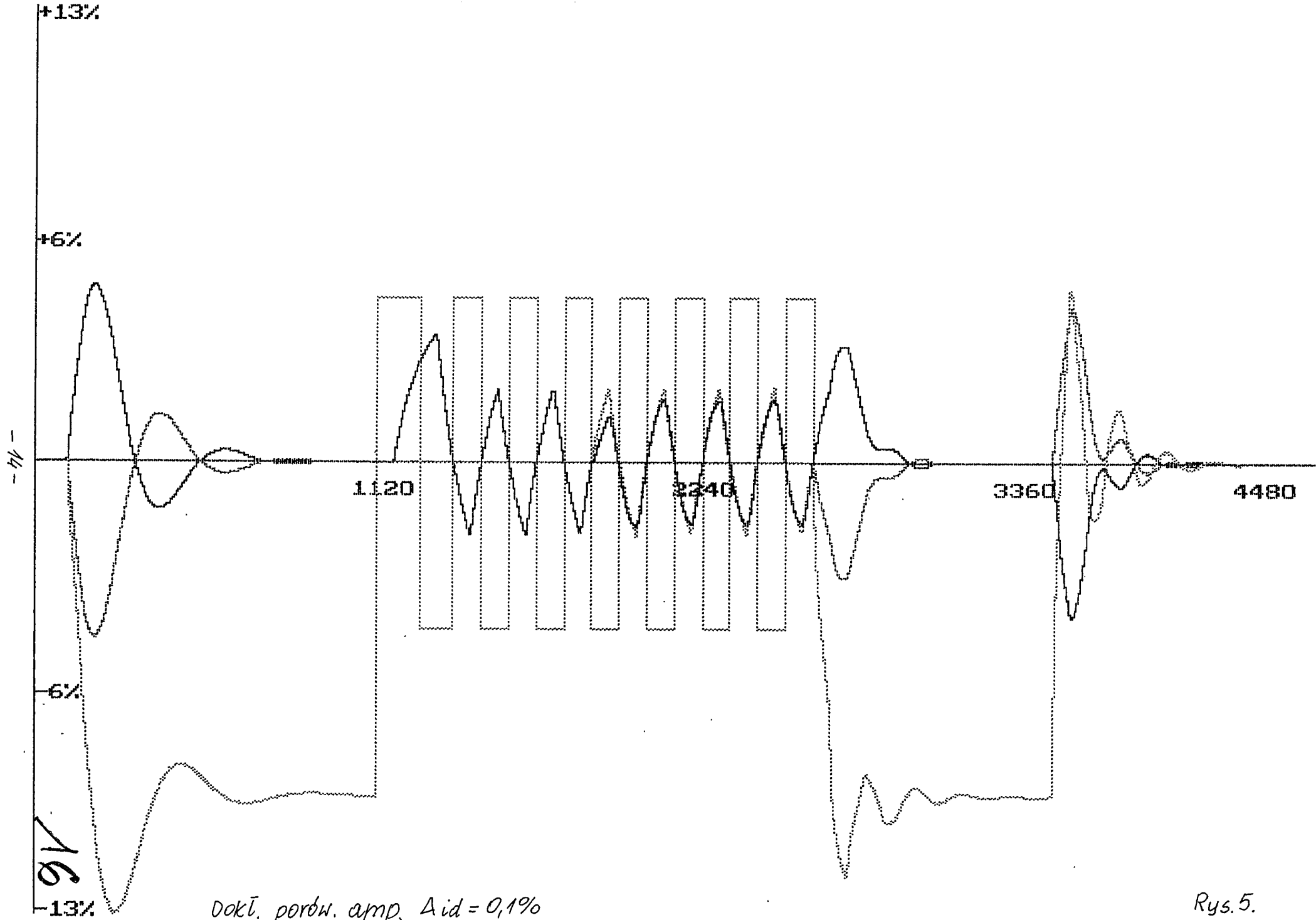
Rys.3.



Dokł. porów. amp. $\Delta id = 0,1\%$

Rys. 4.

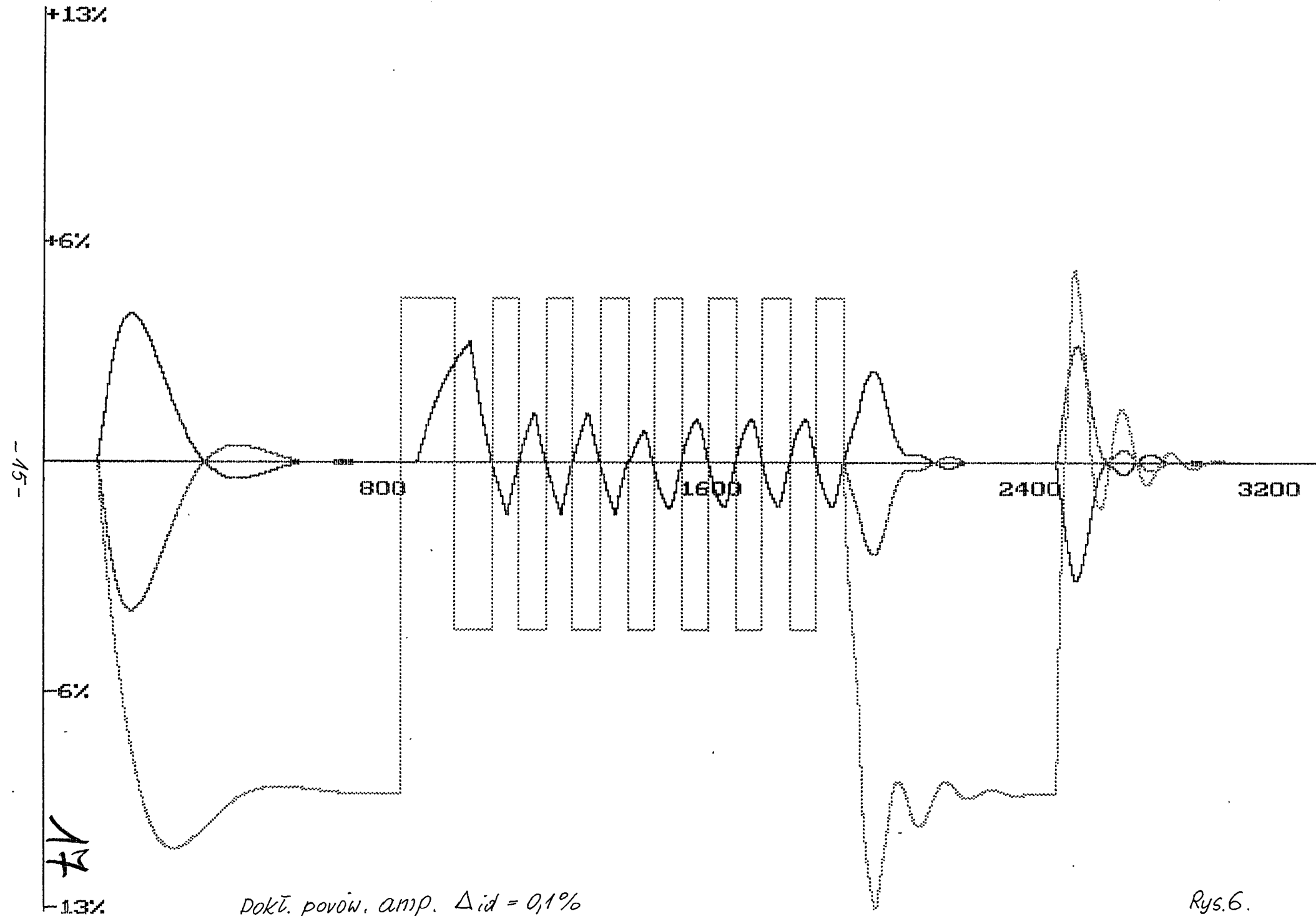
Regulator: $\kappa_p=1,00/1$ $\& \kappa_i=50,00/125,00$, $T_c=10,00/30,75$, σ_{ext} $T_c=85,00$, $T_f=100,0$
 Sygnały: $S_p=0,00\%$; Zakłócenia= $10,00\%$; $0,00\%$; Amplituda.sygnał.ident= $5,00\%$



00k1. por6w. amp. Aid = 0,1%

Rys. 5.

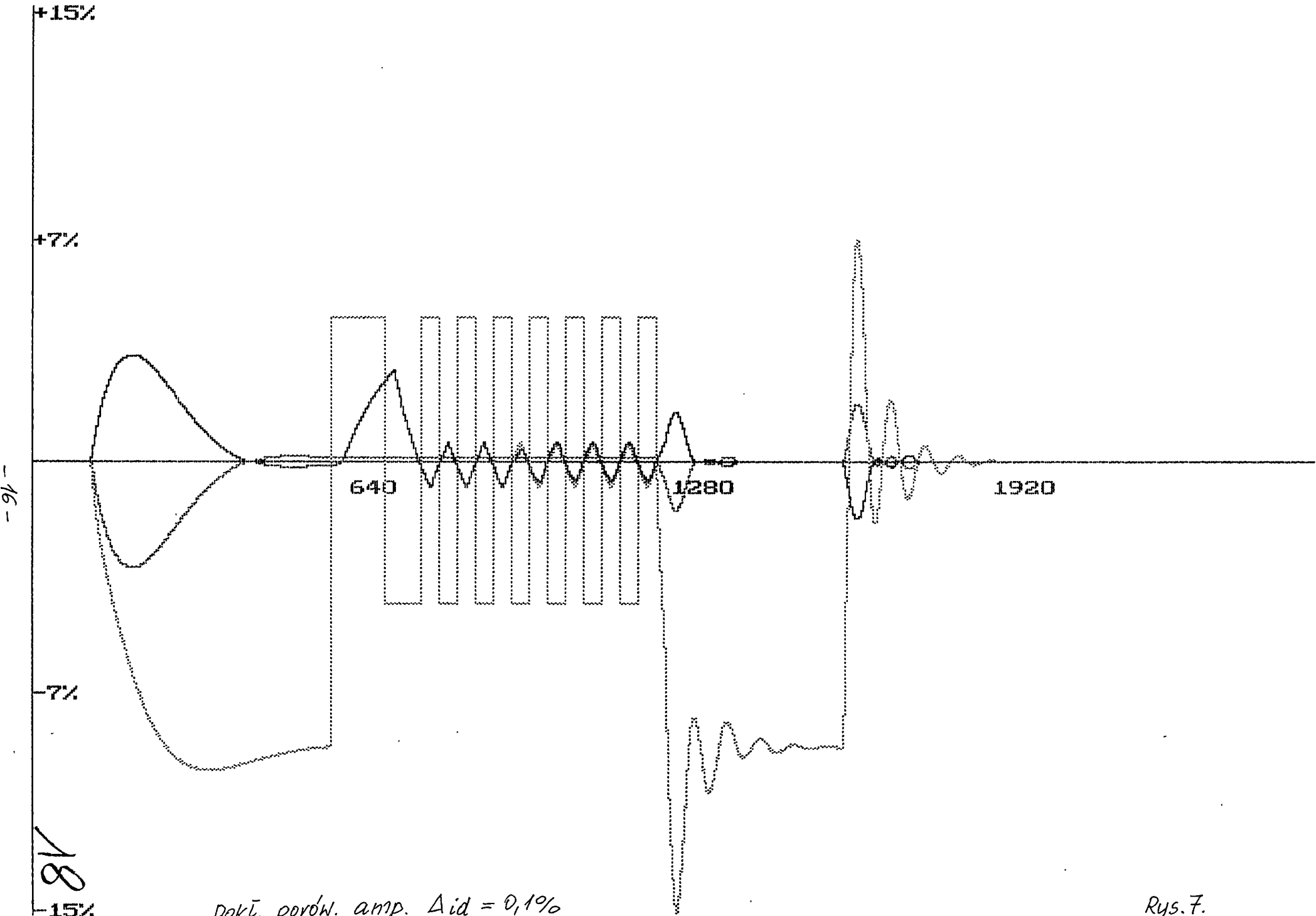
Regulator: $K_p=1.55/2.05$, $T_i=50.00/96.50$, $T_d=10.00/24.12$, Obiekt: $T_s=60.00$, $T_f=100.00$
 Sygnaly: $S_p=0.00\%$; Zakłocenia= $10.00\%; 0.00\%$; Amplituda.sygñ.ident=5.00%



Dokł. powiń. amp. $\Delta id = 0,1\%$

Rys.6.

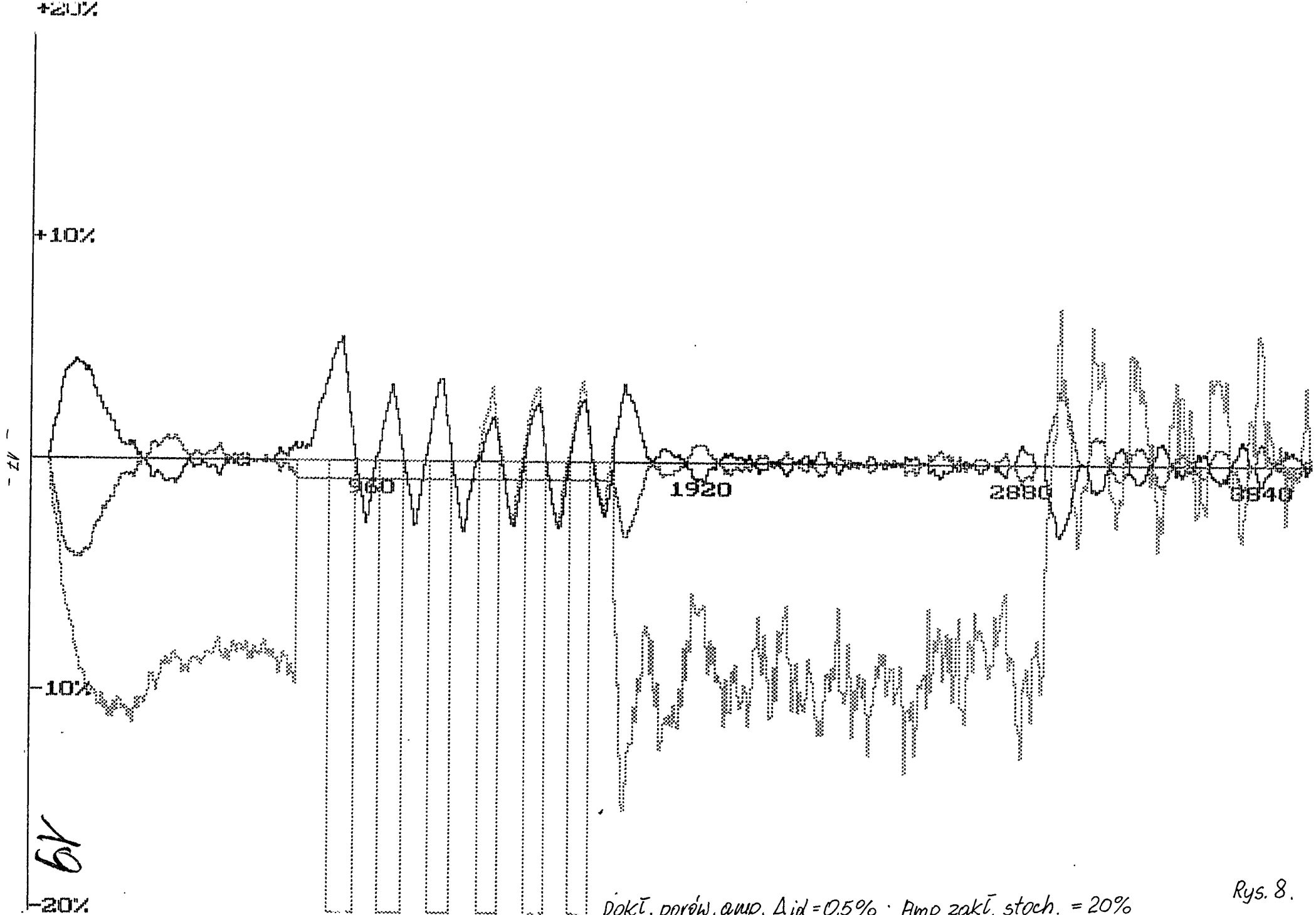
Regulator: $K_p=1,00/2,81$; $T_i=50,00/67,50$; $T_d=10,00/16,85$; Obiekt: $T_0=40,00$; $T_f=100,00$
 Sygnaly: $S_p=0,00\%$; Zakłócenia= $10,00\%; 0,00\%$; Amplituda sygn.ident= $5,00\%$



Dokł. porówn. amp. $\Delta id = 0,1\%$

Rys. 7.

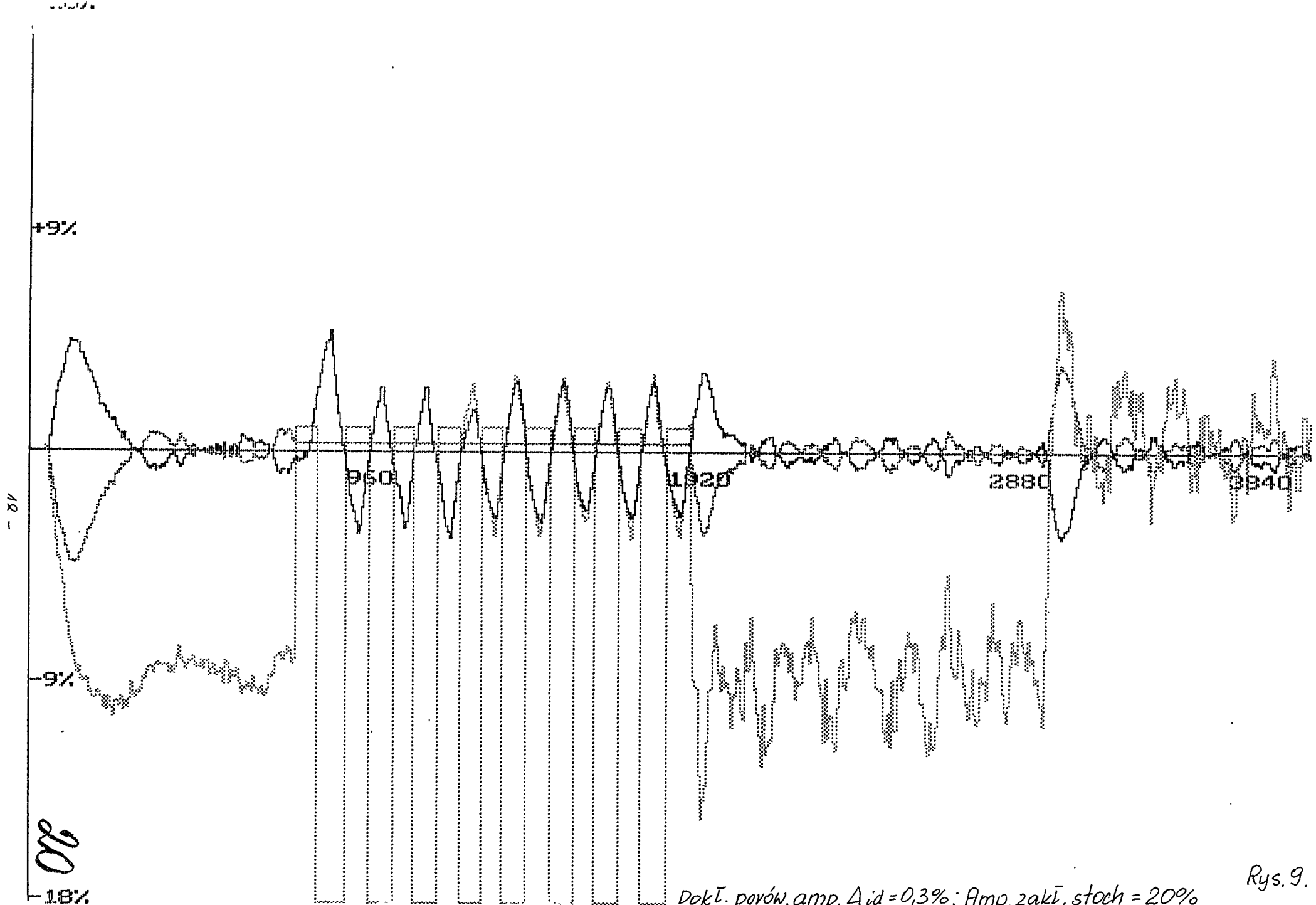
Regulator: $K_p=1,00/5,21$; $T_i=50,00/36,00$; $T_d=10,00/9,00$; Obiekt: $T_c=20,00$, $T_f=100,00$
 Sygnały: $S_p=0,00\%$; Zakłócenia= $10,00\%$; $0,00\%$; Amplituda.sygn.ident= $5,00\%$



Dokł. porów. amp. $\Delta id = 0,5\%$; Amp. zakł. stoch. = 20%

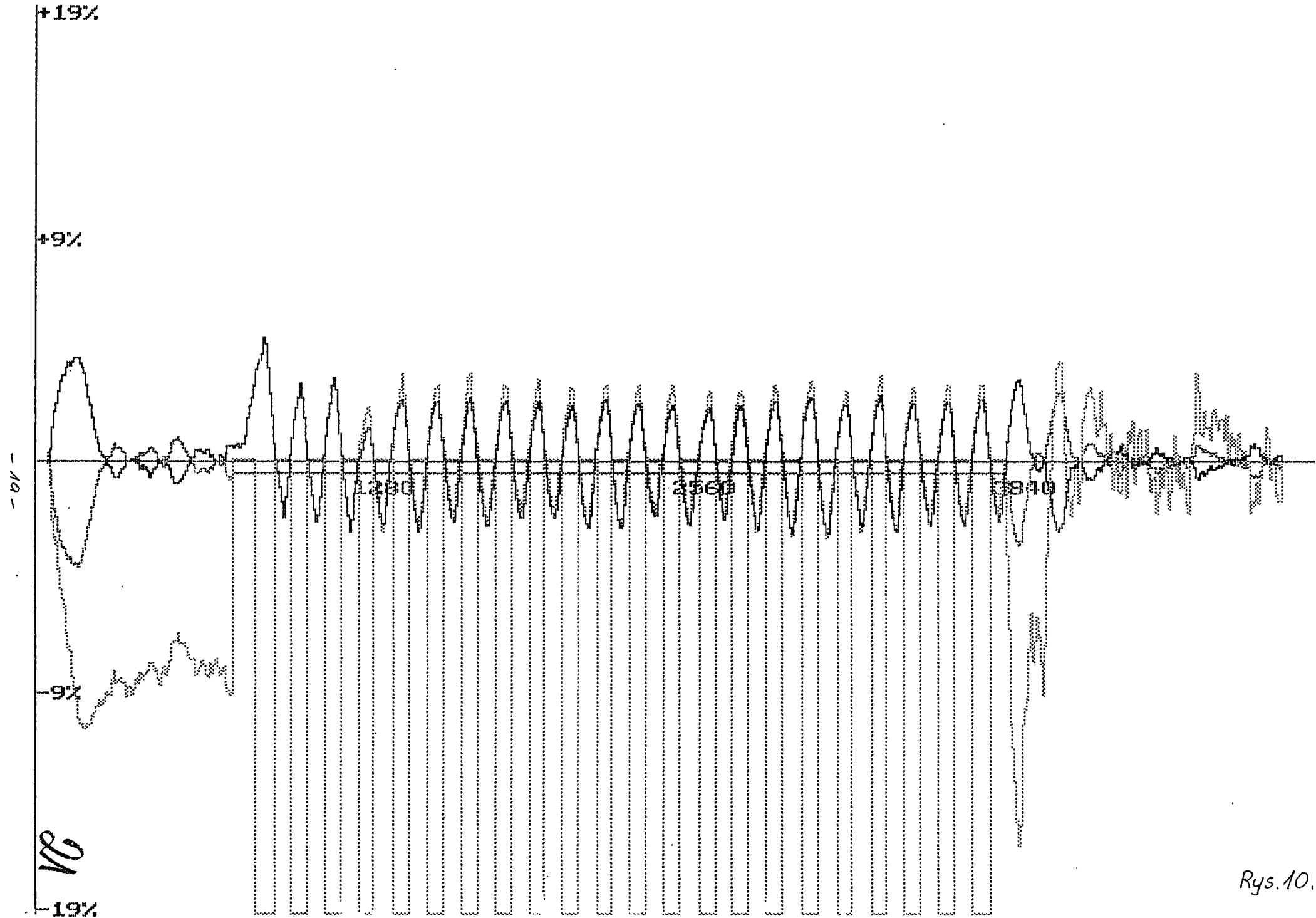
Rys. 8.

Regulator: $K_p=1.00/2.50$; $T_i=50.00/58.00$; $T_d=10.00/17.00$; Obiekt: $T_o=40.00$; $T_f=100.00$
 Sygnaly: $S_p=0.00\%$; Zakłocenia= 10.00%; 0.00%; Amplituda sygn. ident=10.00%



Rys. 9.

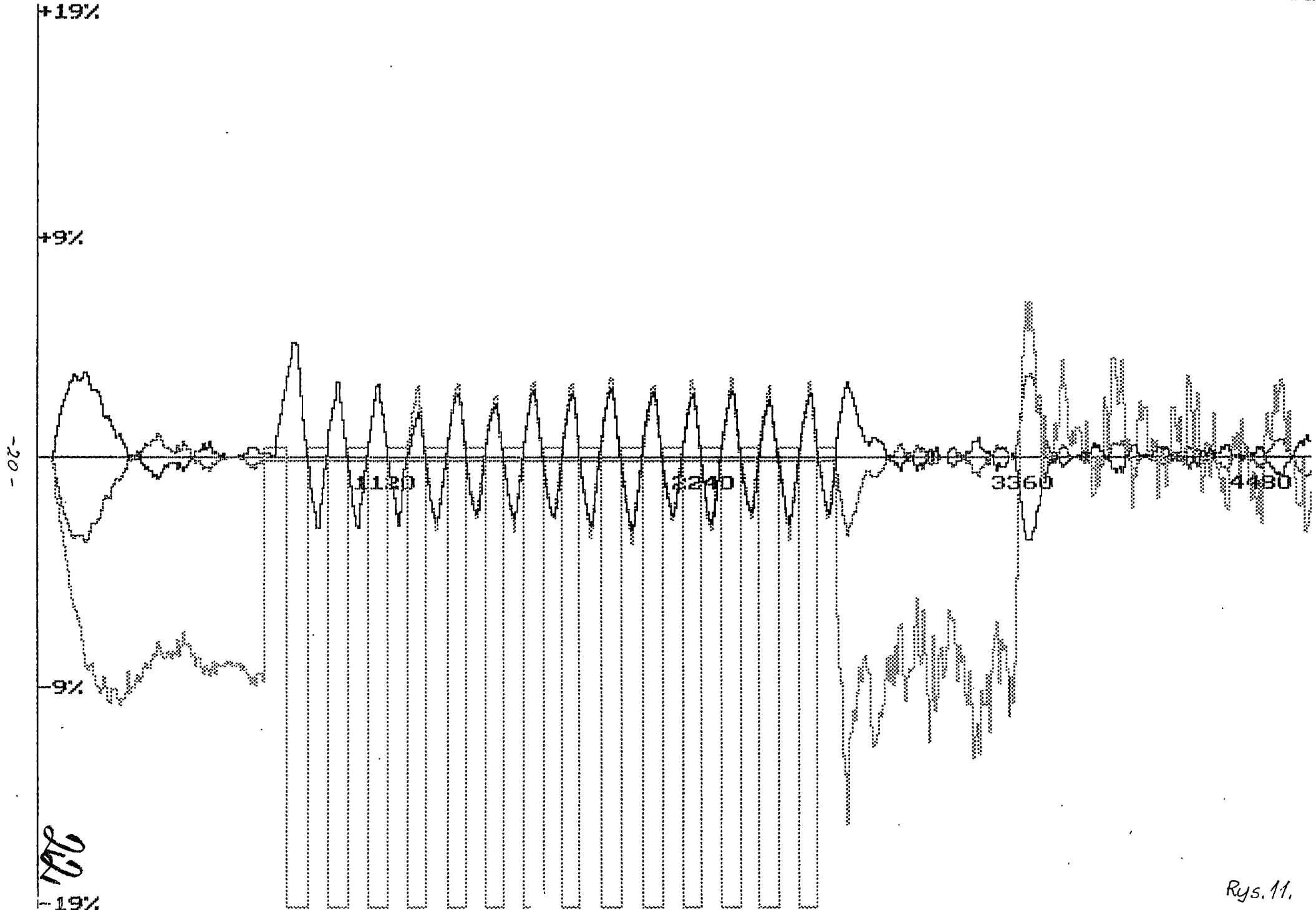
Regulator: $K_p=1.00/2.68$, $T_i=50.00/68.00$, $T_d=10.00/17.00$; Obiekt: $T_0=40.00$, $T_f=100.00$
 Sygnaly: $S_p=0.00\%$; Zakłócenia= 10.00%; 0.00%; Amplituda sygn. ident=10.00%



Rys.10.

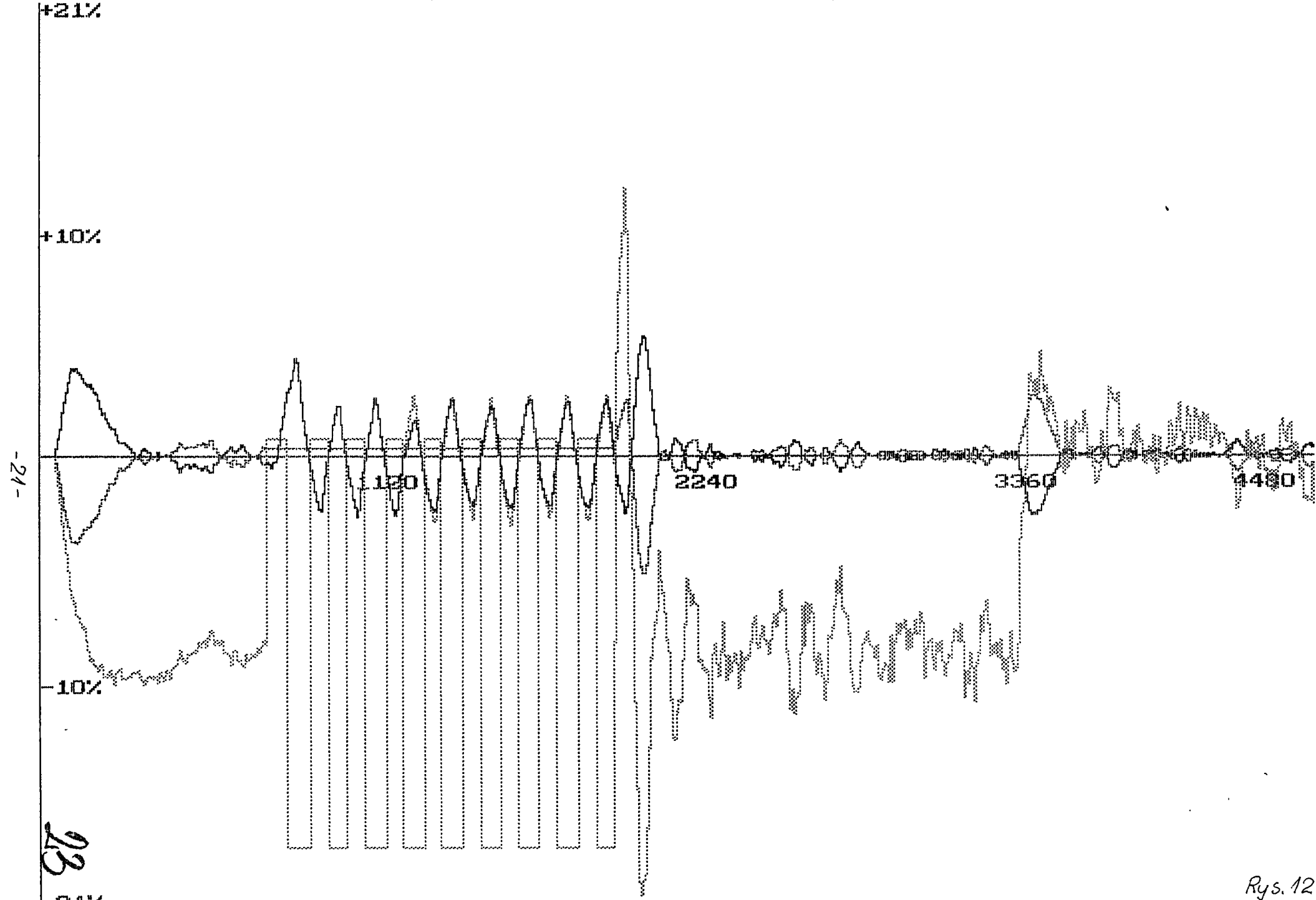
Regulator: $K_p=1.00/2.73$, $T_i=60.00/67.00$, $T_d=10.00/16.75$, $Cb_{akt.}=40.00$, $T_0=40.00$, $T_f=100.00$
 Signały: $S_p=0.00\%$; $Zakłocenia=10.00\%;0.00\%$; $Amplituda_{sygn.ideal}=10.00\%$

Dokł. powód. amp. $\Delta_{id} = 0.1\%$ · Amp. zakł. stoch = 20%



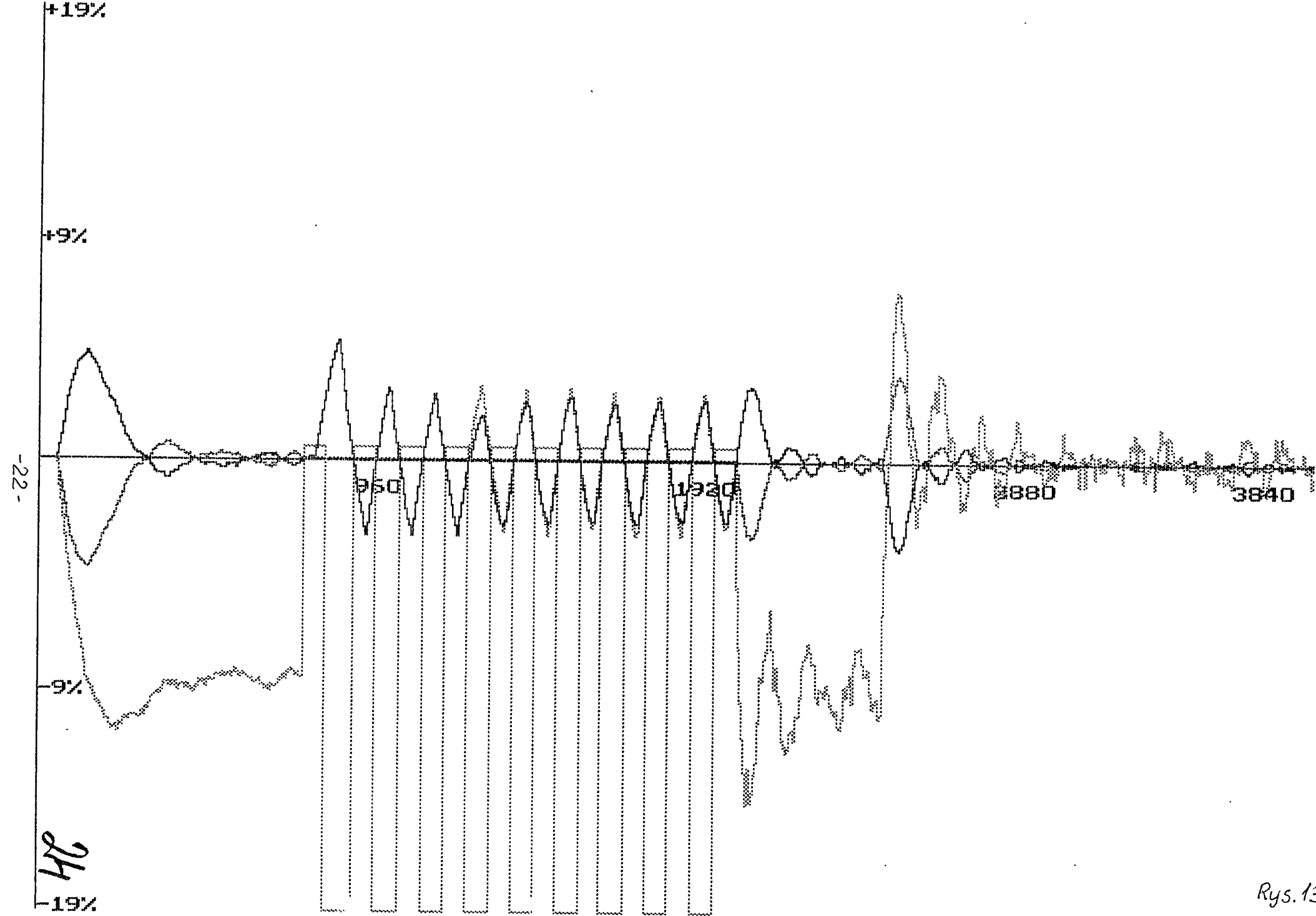
Regulator: $K_p=1.00/2.80$, $T_i=50.00/71.50$, $T_d=10.00/17$, $\beta=0.00$, $\text{cent: } T_0=40.00, T_f=100.00$
 Sygnaly: $S_p=0.00\%$; Zakłócenia = $10.00\%; 0.00\%$; Amplituda sygn. ident = 10.00%
 Dokł. porów. amd. $\Delta id = 0.1\%$ · Amo zakł. stoch. = 20%

Rys. 11.



Rys. 12.

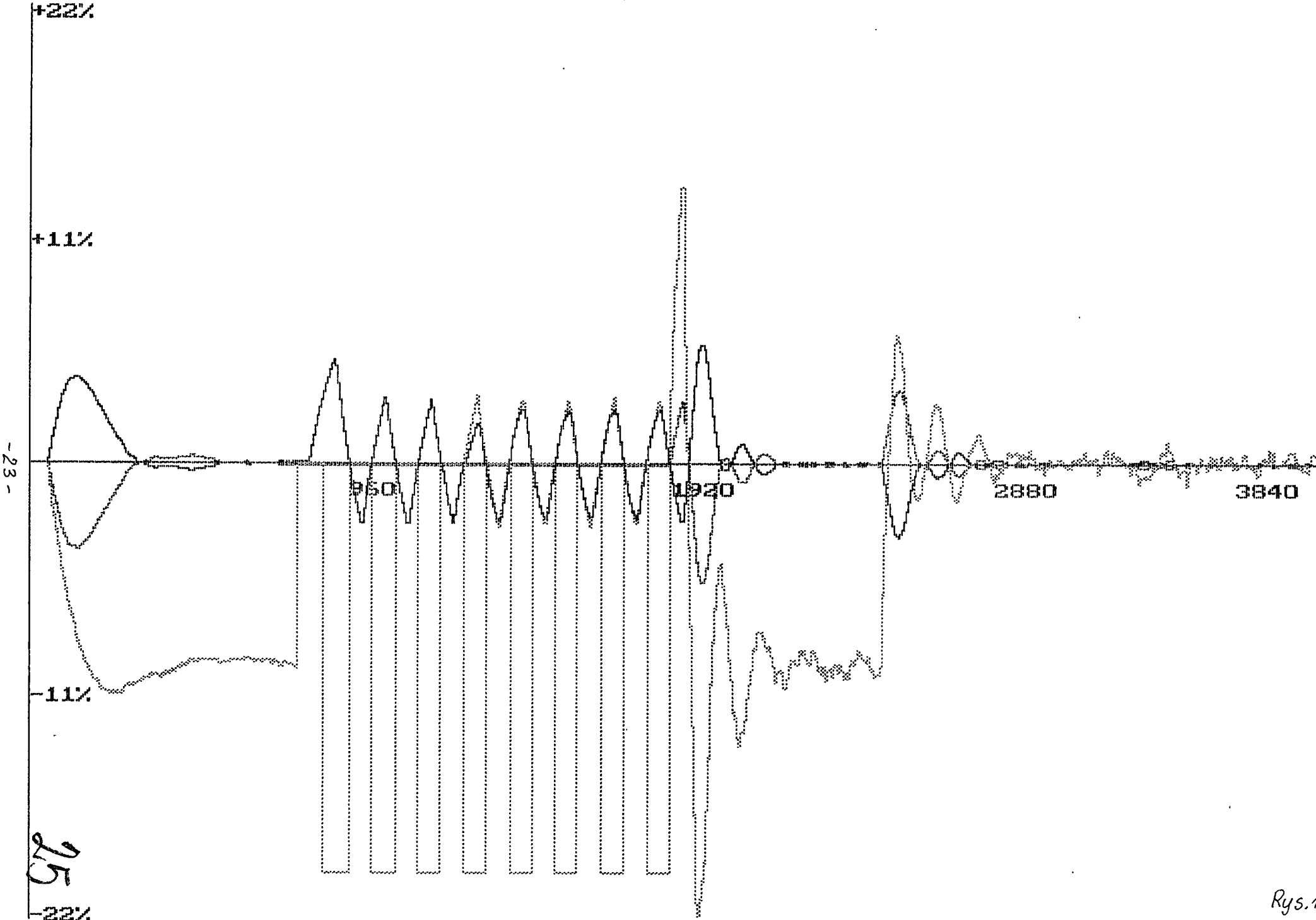
Regulator: $K_p=1.00/2.85$; $T_i=60.00/38.00$; $T_d=10.00/17.00$; Obiekt: $T_0=40.00$; $T_f=100.00$
 Sygnaly: $S_p=0.00\%$; Zakłocenia= 10.00% ; 0.00% ; Amplituda.sygn.ident= 10.00%
 Dokł. porów. amp. $A_{id} = 0,1\%$; $A_{mp.zakł.stoch.} = 15\%$



HC

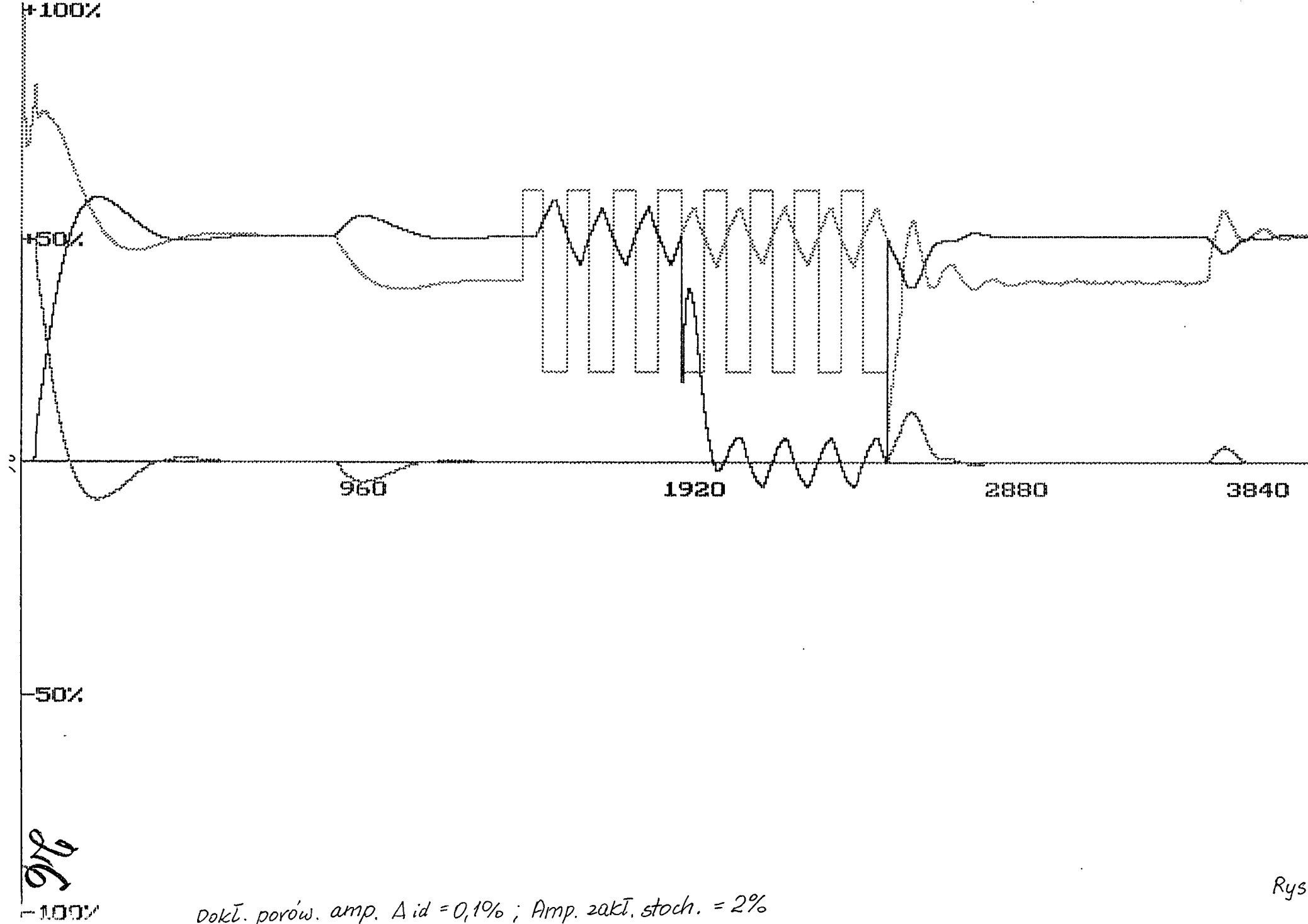
Rys. 13.

Regulator: $K_p=1.00/2.87$; $T_i=60.00/68.50$; $T_d=10.00/17.12$; Obiekt: $T_o=40.00$; $T_f=100.00$
 Sygnaly: $S_p=0.00\%$; Zakłócenia= 10.00% ; 0.00% ; Amplituda.sygn.ident= 10.00%
 Dokł. porów. amp. $\Delta id = 0.01\%$; Amp. zakł. stoch = 10%



Regulator: $K_p=1.00/2.81$; $T_i=60.00/67.50$; $T_d=10.00/16.88$; Obiekt: $T_o=40.00$; $T_f=100.00$
 Sygnaly: $S_p=0.00\%$; Zakłócenia= 10.00% ; 0.00% ; Amplituda.sygn.ident= 10.00%
 Dokł. porów. amp. $\Delta id = 0.01\%$; Amp. zakł. stoch. = 5%

Rys. 14.

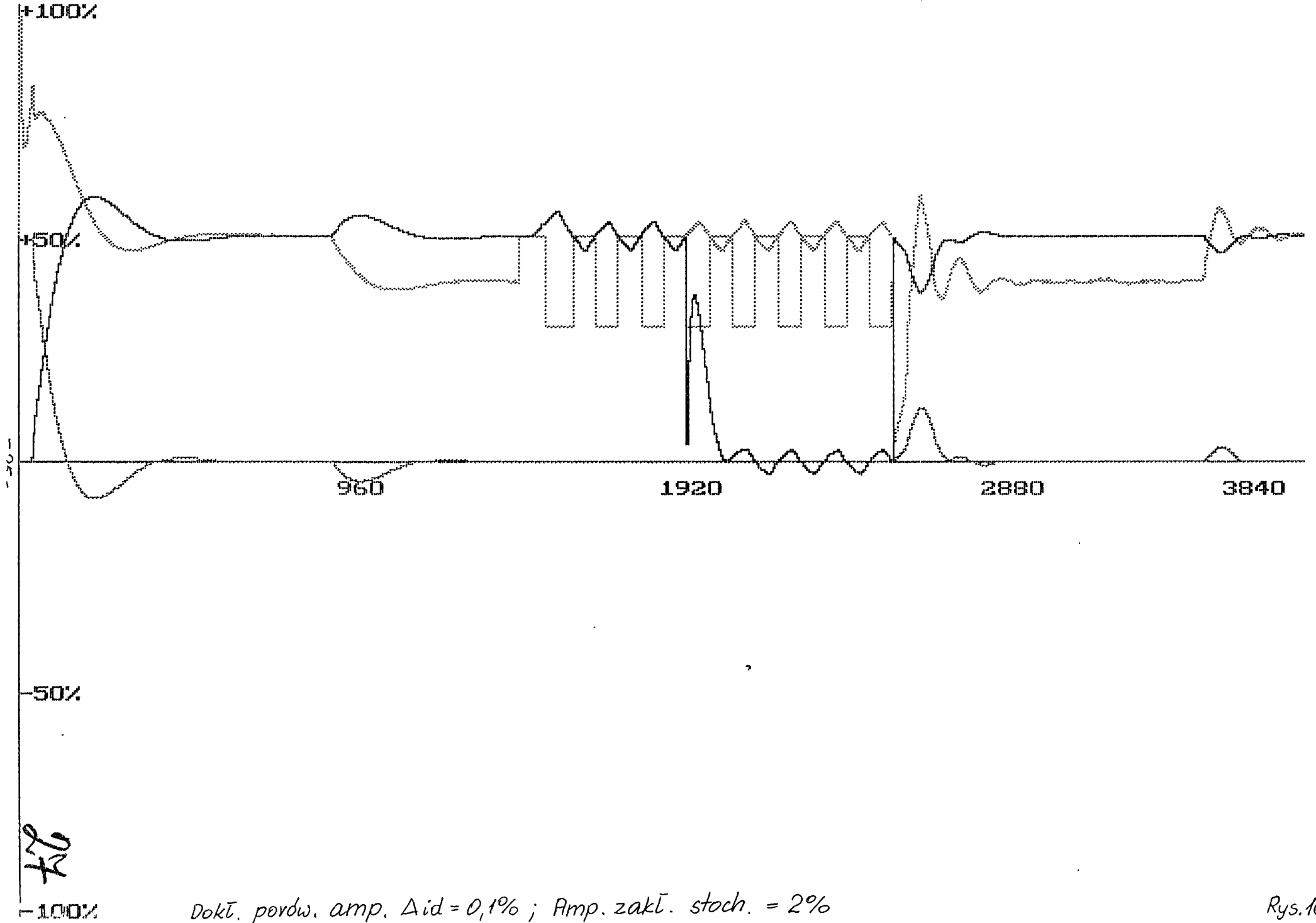


9%

Dokł. porów. amp. $A_{id} = 0,1\%$; Amp. zakł. stoch. = 2%

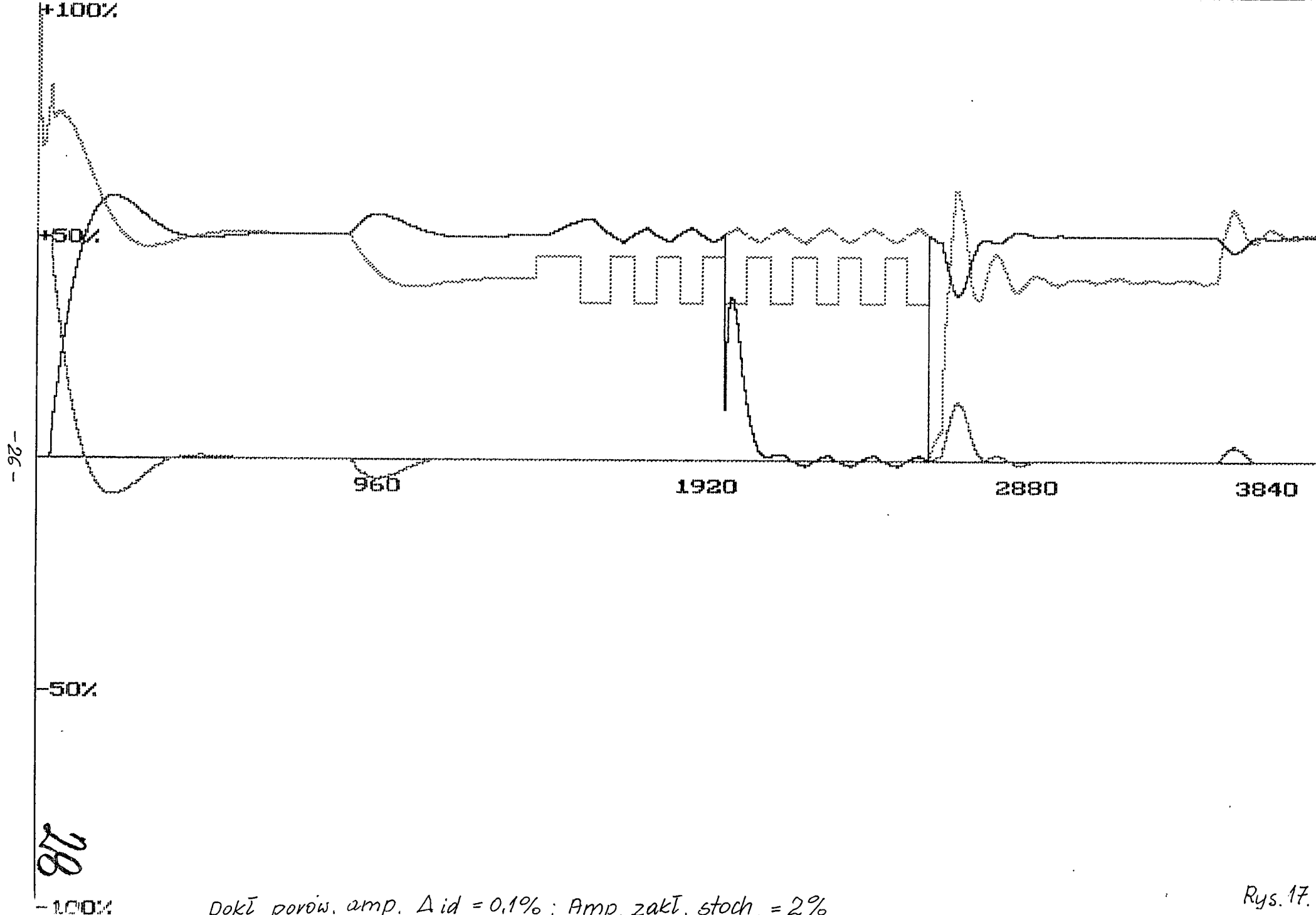
Regulator: $K_p = 15,00$; $T_i = 10,00$; $T_d = 0,50$; $T_c = 10,00$; $T_f = 0,00$; $T_r = 40,00$; $T_s = 100,00$
 Sygnały: $S_p = 50,00\%$; Zakłócenia = 10,00%; 0,00%; Amplituda.sygń.ident = 20,00%

Rys.15.



h
h

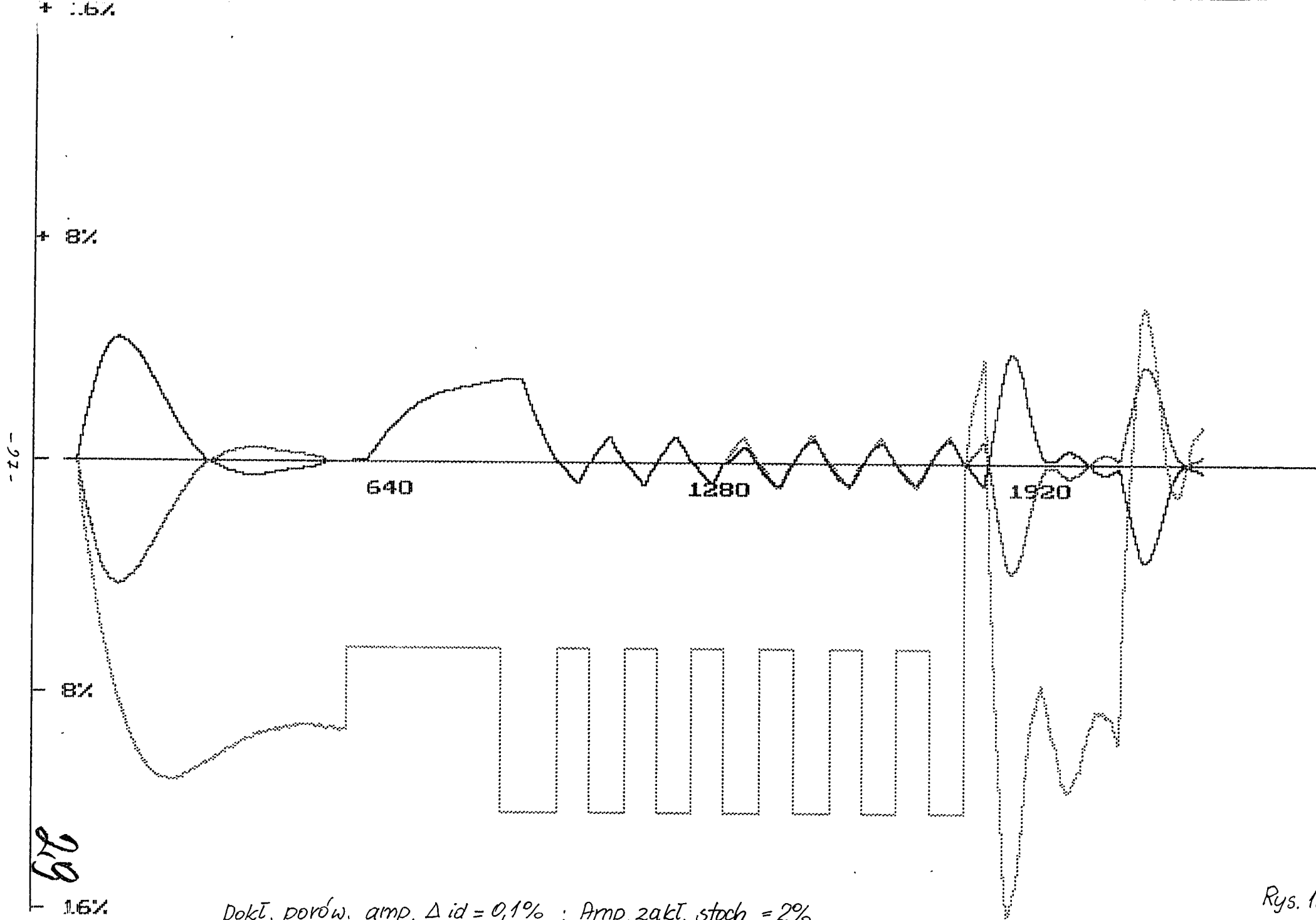
-100% Dokł. porów. amp. $\Delta id = 0,1\%$; Amp. zakł. stoch. = 2% Rys.16.
 Regulator: $K_p = 1,00/2,00$, $T_i = 20,00/12,00$, $T_d = 10,00/17,00$, $C_{ext} = 10=40,00$, $T_f = 100,00$
 Sygnaly: $S_p = 50,00\%$, $Zakłocenia = 10,00\%; 0,00\%$; $Ampituda.sygn.ident = 10,00\%$



Dokł. porów. amp. $\Delta id = 0,1\%$; Amp. zakt. stoch. = 2%

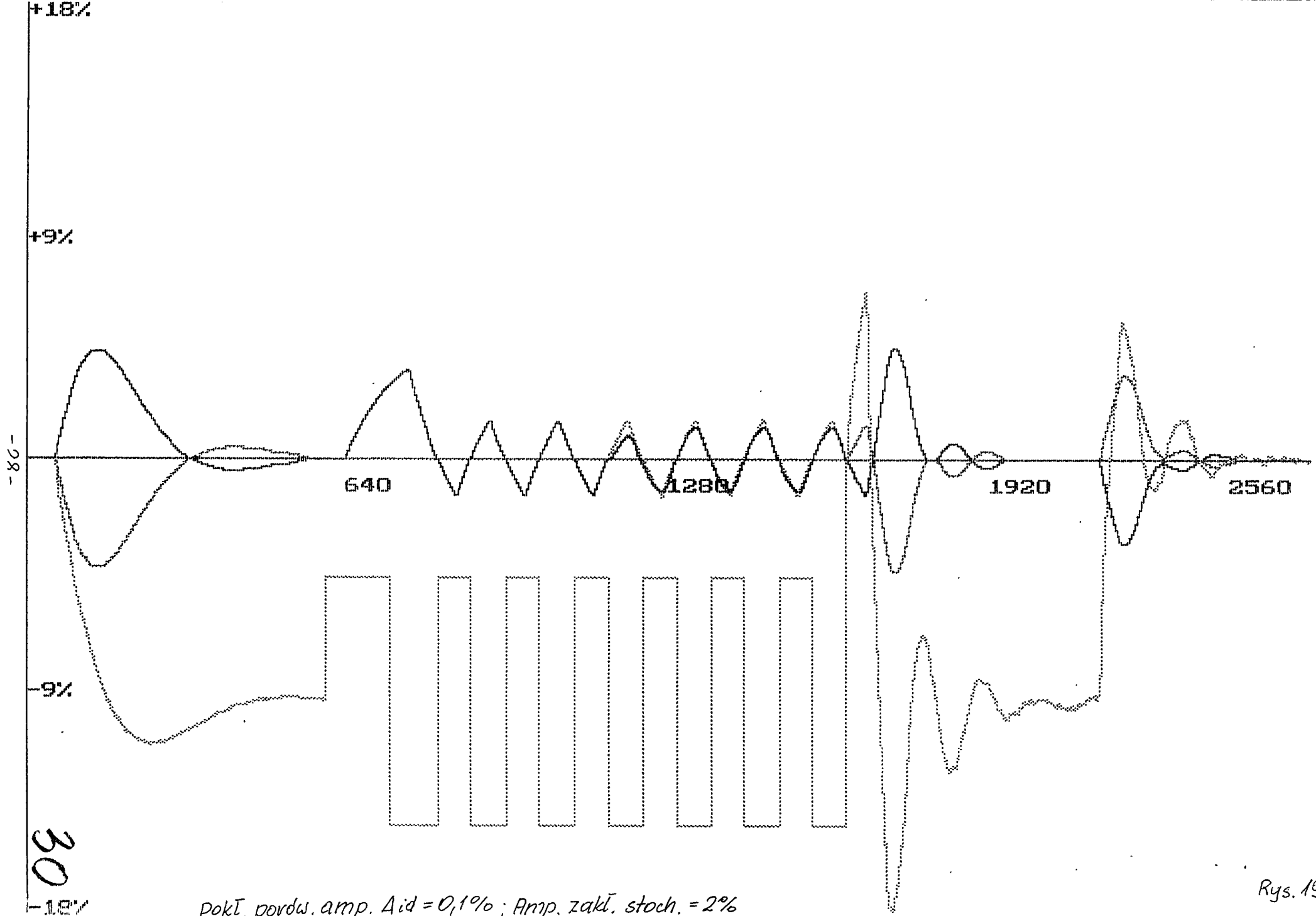
Rys. 17.

Regulator: $Kp=1,00/2,00$; $Ti=0,00/05,00$; $Td=10,00/17,00$; $C_{max}=1,00$; $Tc=40,00$; $Tf=100,00$
 Sygnaly: $\% = 50,00\%$; Zakłócenia = 10,00%; 0,00%; Amplituda sygn. ident = 5,00%



670
 16%
 Dokł. porów. amp. $\Delta id = 0,1\%$; Amp. zakt. stoch = 2%
 Regulator: $\lambda = 1,50/2,73$; $T_i = 50,00/67,50$; $T_d = 10,00/16,00$; $\lambda = 0,00$; $T_o = 40,00$; $T_f = 100,00$
 Sygnaly: $S_p = 0,00\%$; Zakłócenia = 10,00%; 0,00%; Amplituda sygn. ident = 3,00%

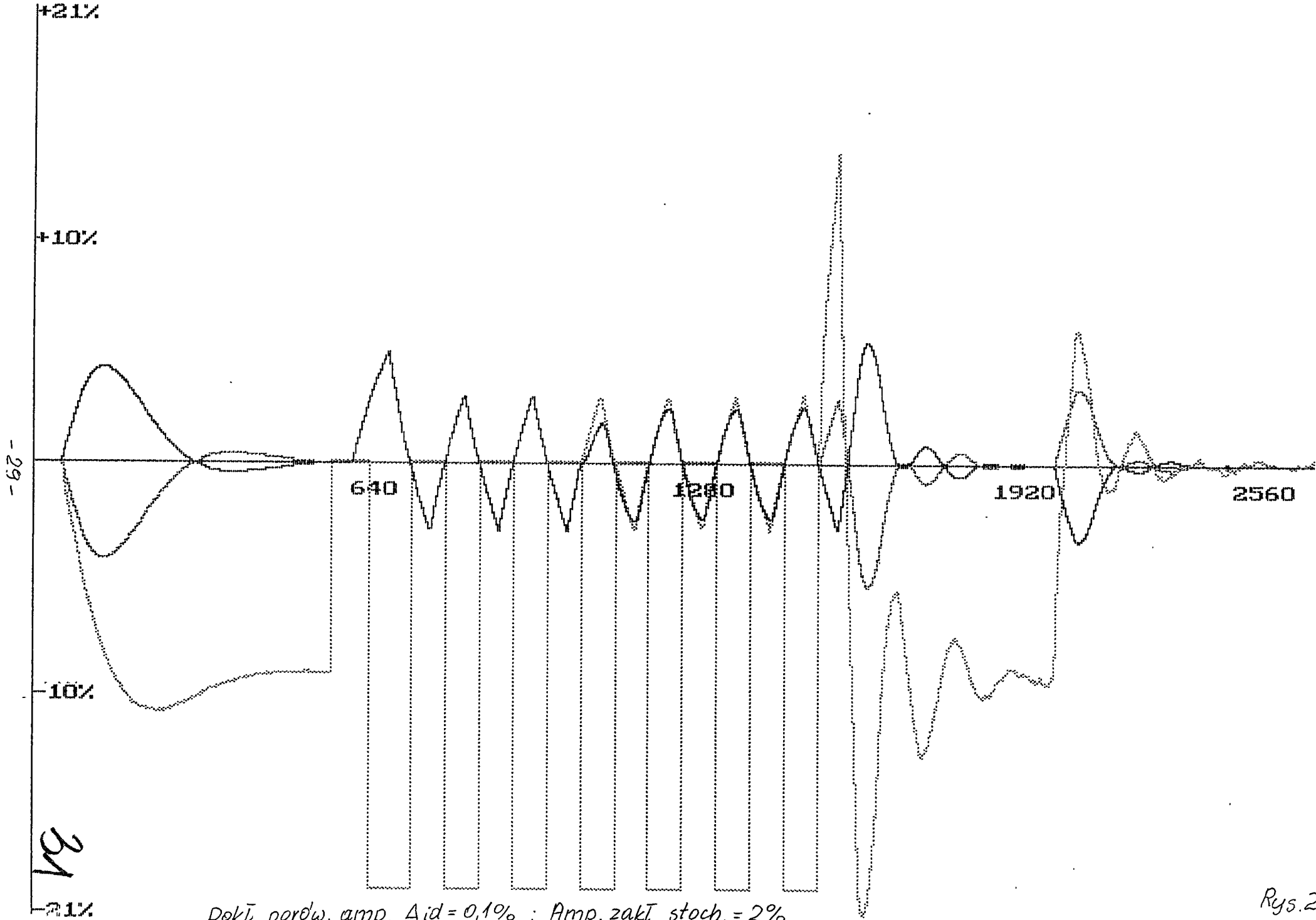
Rys. 18.



Dokl. porow. amp. $\Delta id = 0,1\%$; Amp. zakł. stoch. = 2%

Rys. 19.

Regulator: $\omega = 1,50/2 \text{ } \omega$; $T_i = 60,00/58,00$; $T_d = 13,00/17,00$; Obiekt: $T_0 = 40,00$; $T_f = 100,00$
 Sygnały: $\% = 0,00\%$; Zakłócenia = 10,00%; 0,00%; Amplituda.sygń.ident = 5,00%

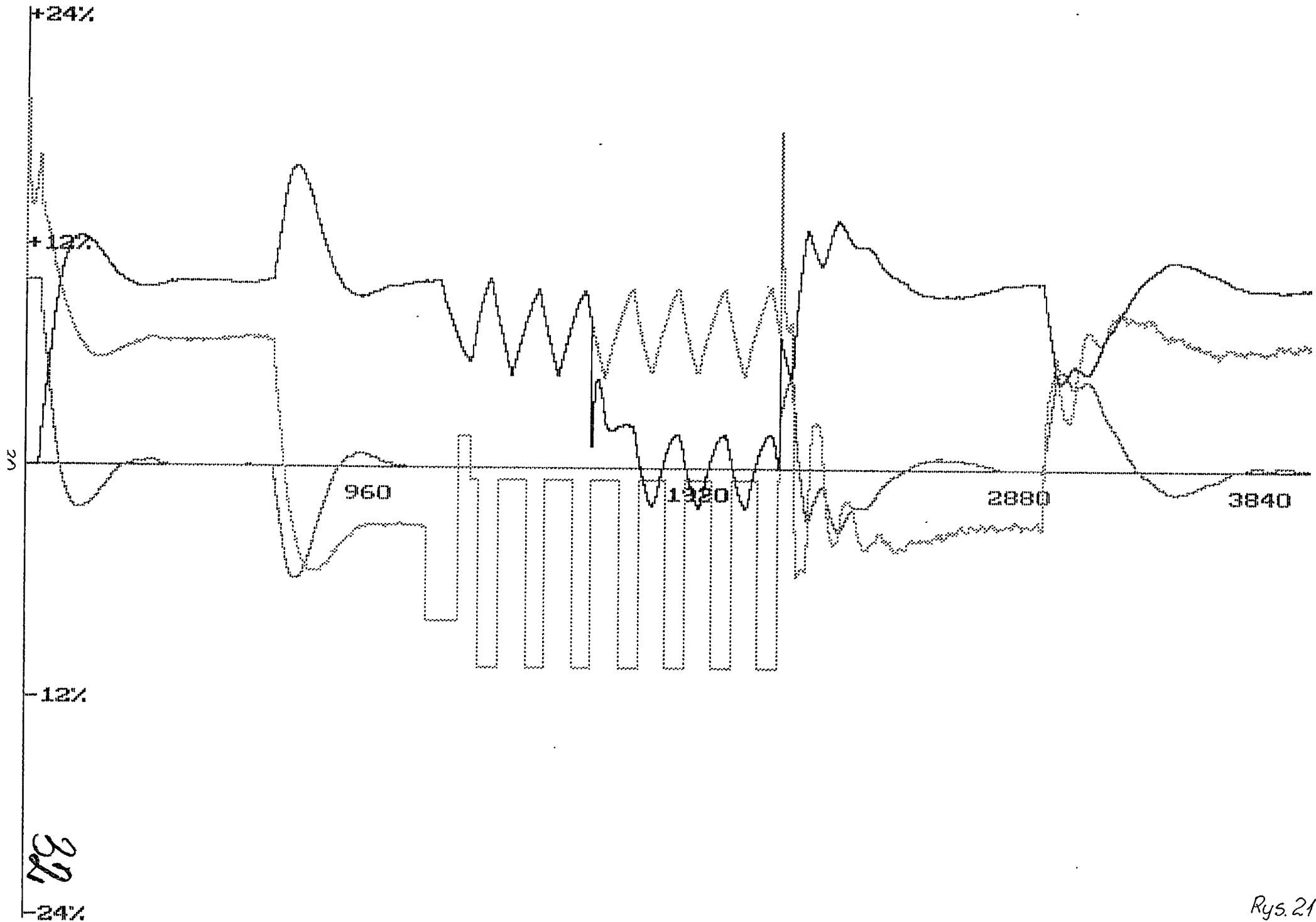


Dokł. porów. amp. $\Delta id = 0,1\%$; Amp. zakł. stoch. = 2%

Regulator: $K=1,00/2$; $T_i=50,00/57,50$; $T_d=10,00/16,00$; Obł. $T=40,00$; $T_f=100,00$
 Sygnal: $S_p=0,00\%$; Zakł. $1=10,00\%$; $2=0,00\%$; Amplituda sygn. ident = 10,00%

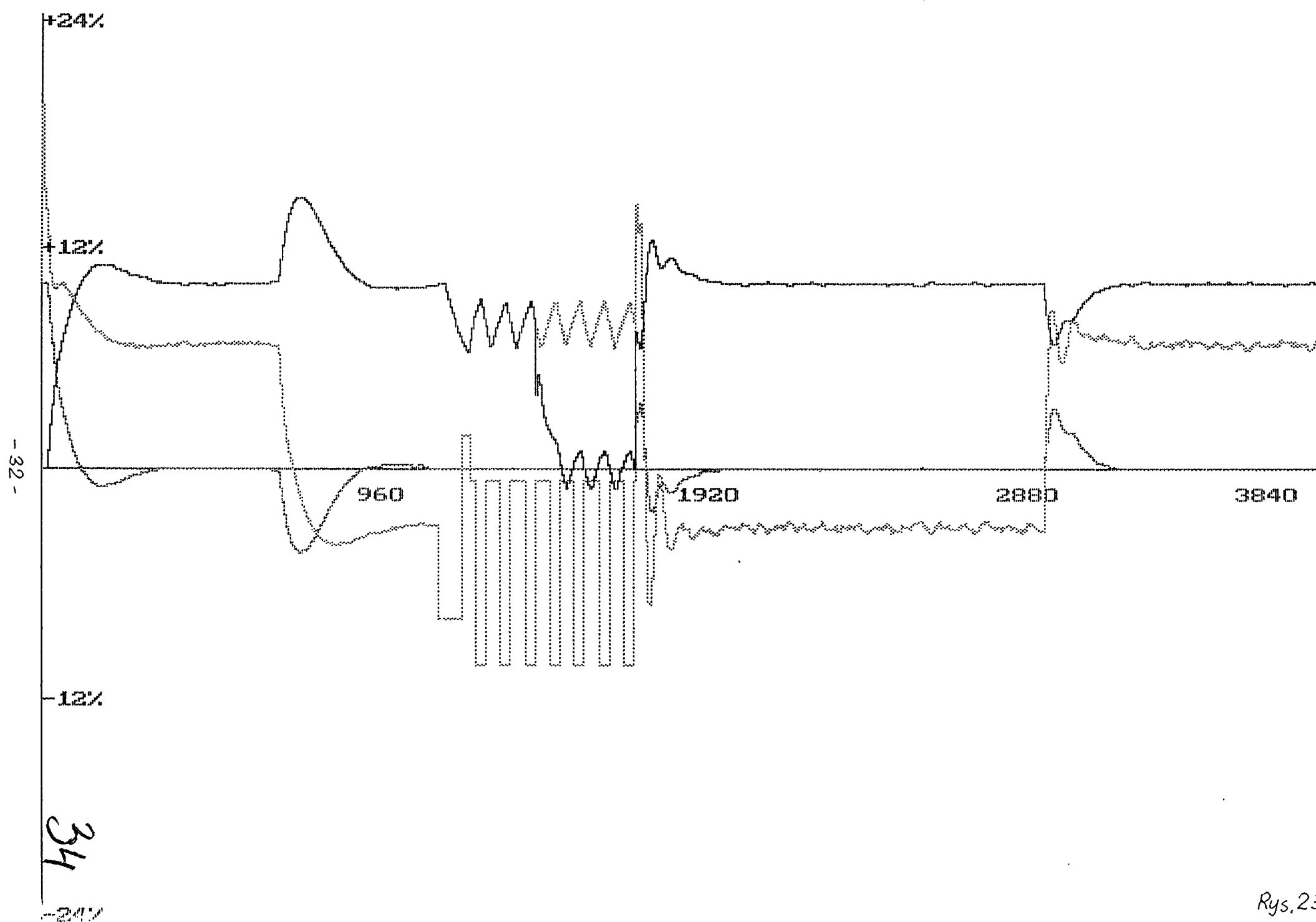
Rys.20.

31



Regulator: $K_p=1.00/0.62$; $T_i=60.00/46.20$; $T_d=10.00/70.00$; Objekt: $T_0=40.00$; $T_f=100.00$
 Sygnaly: $S_p=10.00\%$, Zakłócenia= $10.00\%; 0.00\%$; Amplituda sygn. ident= 5.00%

Rys. 21.

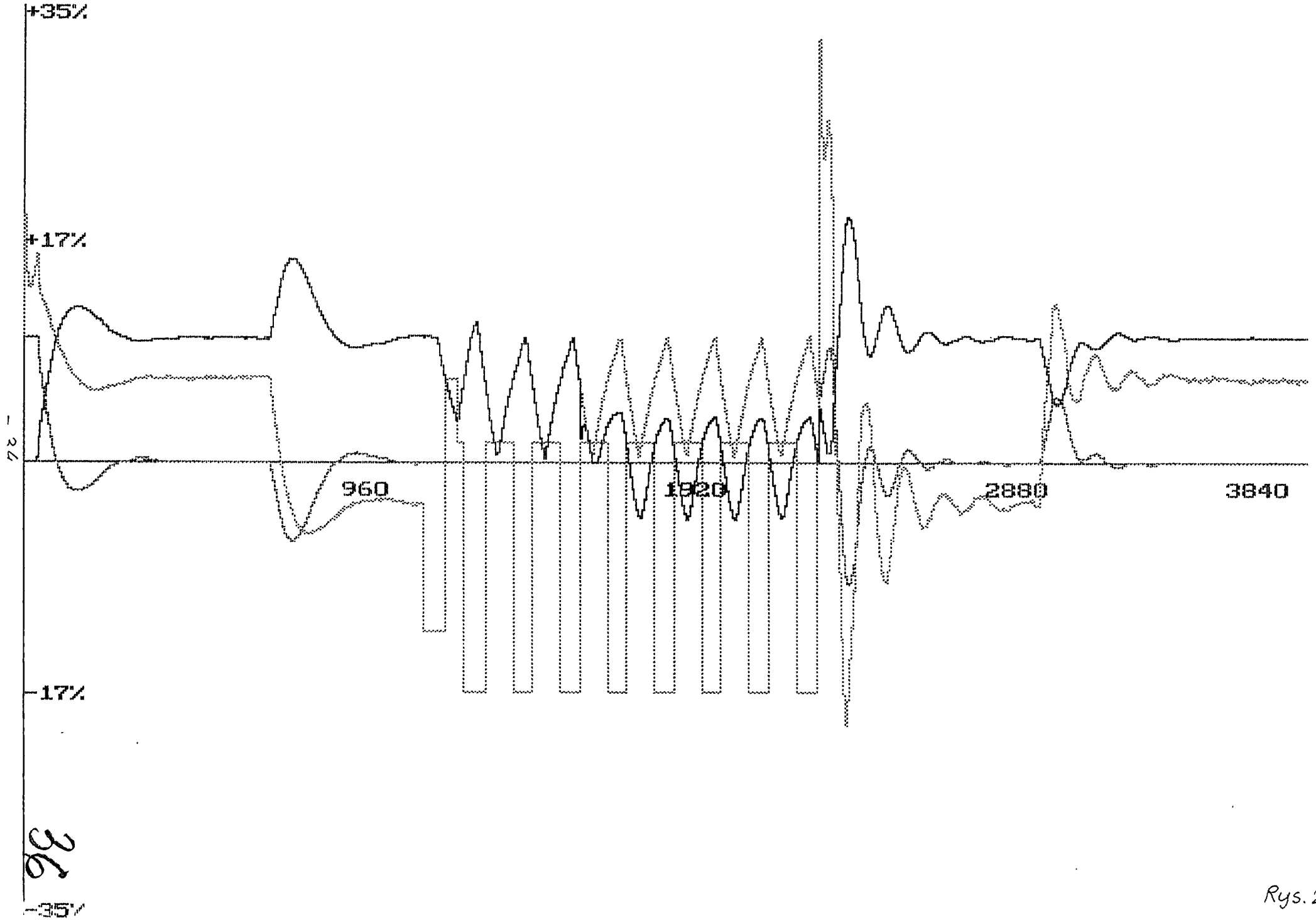


Parametry: $S_p=10.00\%$, $Zaklonen=10.00\%$, 0.00% ; $T_s=10.00$, $T_f=100.00$, $T_c=20.00$, $T_r=100.00$
 Sygnaly: $S_p=10.00\%$, $Zaklonen=10.00\%$, 0.00% ; $Amplituda\ sygn.\ idenc=5.00\%$

Rys.23.

34

-32-



Rys. 25.

Resultator: $\alpha_{\text{sp}} = 10.00\%$; $\gamma_{\text{Zakl}} = 10.00\%$; $\gamma_{\text{C}} = 15.00\%$; $\gamma_{\text{D}} = 17.70\%$; $\gamma_{\text{C}} = 40.00\%$; $\gamma_{\text{F}} = 100.00\%$
 Sygnaly: $\text{Sp} = 10.00\%$; $\text{Zakl} = 10.00\%$; 0.00% ; $\text{Amplituda} = 10.00\%$

4. OMÓWIENIE BADAŃ AUTOSTROJENIA CIĄGŁEGO (DOKŁADNEGO)

Strojenie dokładne polega na analizie charakterystyki częstotliwościowej sygnału uchybu regulacji. W dobrze dostrojonym do charakterystyki obiektu i aktualnie występujących zakłóceń regulatorze, charakterystyka ta powinna być możliwie płaska dla pasma częstotliwości przenoszonych przez obiekt regulacji.

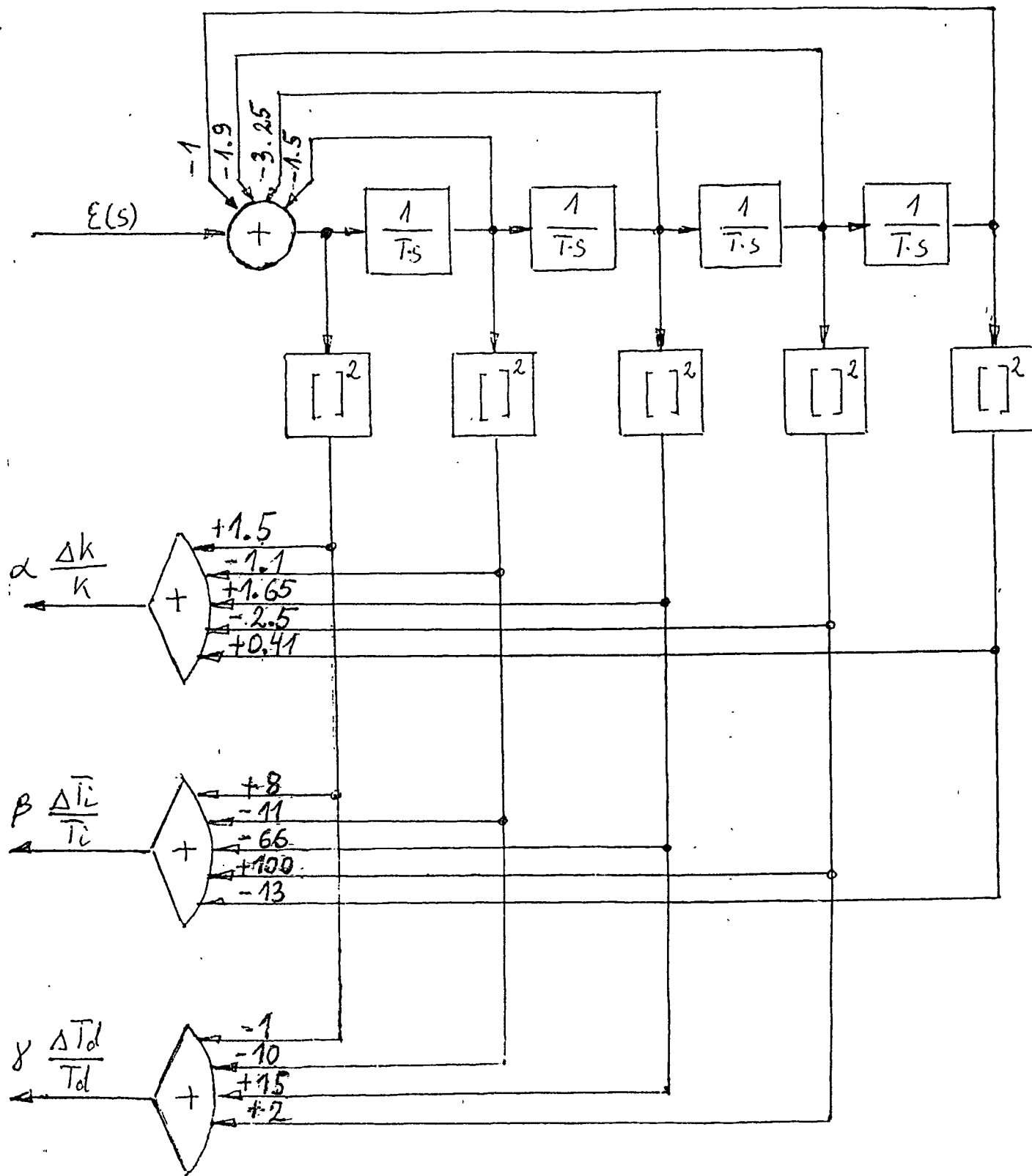
Zmiana każdego z parametrów regulacyjnych wpływa na polepszenie tłumienia zakłóceń w pewnym paśmie częstotliwości i pogorszenie tłumienia w innym zakresie częstotliwości. Przy znajomości charakterystyk oddziaływania zmiany parametrów regulacyjnych na kształt charakterystyki zamkniętego układu regulacji automatycznej, każdy z parametrów regulacyjnych może być strojony niezależnie i w stosunkowo prosty sposób. Daje to możliwość realizacji algorytmu odznaczającego się dość szybkim nadażaniem za zmianami charakterystyki obiektu i zmianami zakłóceń, przy małych wymaganiach co do mocy obliczeniowej mikroprocesora.

Przestrzajanie (autostrojenie) nastaw regulatora odbywa się z nastawianym współczynnikiem wagi oddzielnie dla każdego parametru regulacyjnego.

Algorytm autostrojzenia dokładnego został zrealizowany w oparciu o szeregowo połączenie czterech członów całkujących, o tej samej stałej czasowej całkowania - tak jak to pokazano na rys.26.

Układ ten pozwala na dobre przybliżenie pożądaných przebiegów charakterystyk przestrzajania poszczególnych parametrów regulatora : $f_K(w)$, $f_I(w)$, $f_D(w)$. Charakterystyki te w układzie z rys.26 mogą być w dużym zakresie modelowane (zmieniane) drogą odpowiedniego doboru współczynników w_i dla odpowiednich sygnałów, przed ich wprowadzeniem na węzły sumacyjne.

Dane wyjściowe wartości tych współczynników, przyjęte zgodnie z wstępnymi założeniami [1] i [2], okazały się nie całkowicie trafne. W czasie badań współczynniki zostały skorygowane tak, że na rys.26 podano już ich aktualne skorygowane wartości. Nie ma jednak pewności, że są to już wielkości optymalne, gdyż wymagałoby to znacznie obszerniejszych badań wykraczających poza ramy niniejszej pracy.



Rys. 2.6 Schemat strukturalny układu autostrojzenia dokładnego.

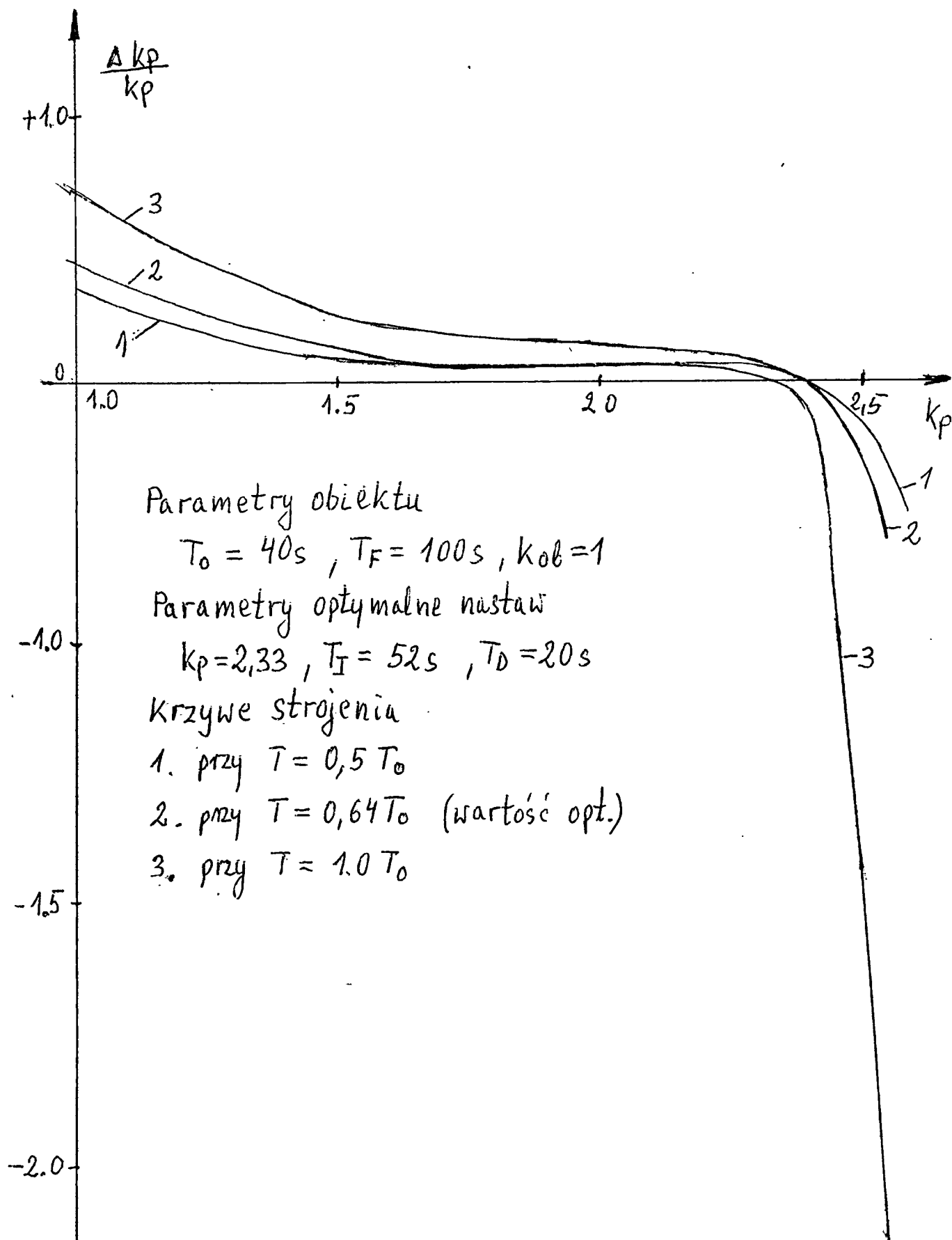
Głównym zadaniem badań algorytmów autostrojzenia dokładnego było sprawdzenie charakterystyk strojenia poszczególnych parametrów regulatora : K_p , T_I i T_D .

Badania wykonano podając zakłócenie skokowe na wyjście obiektu regulacji oraz mierząc wartość wyliczonych przez algorytm wskaźników pożądanych zmian parametrów dla tego typu zakłócenia. Pożądane dostrojenie regulatora do parametrów obiektu i charakterystyki zakłóceń powinno odpowiadać wskaźnikowi minimalizacji całki z kwadratu odchyłki regulacji.

Wyniki badań podane są na rys.27 oraz w tabelach 2 do 5.

Umożliwiają one wyciągnięcie następujących wniosków :

- Zaproponowana koncepcja realizacji autostrojzenia dokładnego jest poprawna. Wskaźniki pożądanych zmian parametrów regulatora oddziałują w kierunku dojścia do stanu ustalonego odpowiadającemu w przybliżeniu nastawom optymalnym dla danego typu zakłócenia i danych parametrów obiektu regulacji. Przebieg wskaźników jest monotoniczny w funkcji wartości parametru strojonego a przejście przez wartość zerową odpowiada w przybliżeniu nastawom optymalnym.
- Praca algorytmu jest poprawna przy różnych wartościach parametrów obiektu regulacji.
- Układ dostrajania nie jest specjalnie wrażliwy (patrz rys.27) na wartość stałej T w szeregowym połączeniu członów całkujących, stanowiących podstawę układu przestrajania tak jak to pokazano na rys.26. Koncepcja doboru wartości stałej T w oparciu o nastawę stałej różniczkowania T_D ($T = 1,28 T_D$) wydaje się być uzasadniona.
- W trakcie badań zaobserwowano pewne oddziaływania skrośne pomiędzy wskaźnikami strojenia poszczególnych parametrów. Dotyczy to głównie parametrów T_D i T_I . Współczynnik strojenia wzmocnienia regulatora K_p jest natomiast stosunkowo mało wrażliwy na zmiany nastaw T_D i T_I . Oddziaływania skrośne wskazują, że osiągnięcie stanu równowagi wymagać będzie wykonanie co najmniej kilku kroków strojenia. Istnieje jednak pewne niebezpieczeństwo wypadnięcia układu z zakresu strojenia przy dużych, szybkich zmianach parametrów obiektu lub specjalnie dobranych



Rys.27 Pomierzone przebiegi funkcji autostrojenia $\frac{\Delta k_p}{k_p} = f(k_p)$ dla różnych wartości T , przy zakłóceniu skokowym na wejściu obiektu.

40

Tabela 2

Warunki pomiarów	$K_{ob} = 1,5; T_O = 60s; T_F = 100s; K_P = 1,56; T_D = 30,0s$ Zakł. skokowe na wejście obiektu				
Nastawa T_I [s]	55	78	98	120	150
$\frac{\Delta K_P}{K_P}$	0,106	0,145	0,163	0,206	-0,256
$\frac{\Delta T_I}{T_I}$	1,54	0,38	-0,03	-0,84	-0,256
$\frac{\Delta T_D}{T_D}$	-0,03	-0,35	-0,46	-0,68	-0,80

Tabela 3

Warunki pomiarów	$K_{ob} = 1,5; T_O = 60; T_F = 100; K_P = 1,56; T_I = 78$ Zakł. skokowe na wejściu obiektu				
Nastawa T_D [s]	10	20	30	33	35
$\frac{\Delta K_P}{K_P}$	0,28	0,22	0,145	0,074	0,008
$\frac{\Delta T_I}{T_I}$	3,73	1,34	0,38	0,13	-0,02
$\frac{\Delta T_D}{T_D}$	1,03	0,26	-0,35	-0,86	-1,48

Tabela 4

Warunki pomiarów	$K_{ob} = 1,5; T_o = 40s; T_F = 100s; K_p = 2,33; T_D = 20s$ Zakł. skokowe na wejściu obiektu			
Nastawa T_I [s]	30,0	52,0	70,0	100
$\frac{\Delta K_p}{K_p}$	-0,025	0,002	-0,03	0,08
$\frac{\Delta T_I}{T_I}$	1,29	-0,01	-0,48	-0,64
$\frac{\Delta T_D}{T_D}$	0,28	-0,24	-0,54	-0,25

Tabela 5

Warunki pomiarów	$K_{ob} = 1,5; T_o = 40s; T_F = 100s; K_p = 2,33; T_I = 52s$ Zakł. skokowe na wejściu obiektu			
Nastawa T_D [s]	10	15	20	21
$\frac{\Delta K_p}{K_p}$	0,08	-0,052	0,0016	-0,05
$\frac{\Delta T_I}{T_I}$	-0,08	0,00	0,09	0,15
$\frac{\Delta T_D}{T_D}$	0,39	+0,09	-0,24	-0,50

złóśliwych zakłóceniach.

- Wskaźniki strojenia nie powinny być bezpośrednio użyte do przestrajania nastaw, gdyż w czasie przebiegu przejściowego w układzie regulacji ulegają one stosunkowo dużym wahaniom. Charakterystyki wskaźników zmian parametru regulowanego są nieliniowe (patrz rys.27), a prócz tego wartość ich silnie zależy od amplitudy zakłóceń. Nasuwa się wniosek, że przestrajanie parametrów powinno odbywać się skokowo co pewien określony czas, jeśli w tym czasie wskaźniki strojenia parametrów przekrocza dopuszczalne wartości. Charakterystyka przestrajania parametrów powinna być nieliniowa w funkcji wartości wskaźników przestrajania.

Powinna ona obejmować jedną lub kilka dopuszczalnych wartości skokowych procentowych zmian parametru. Algorytm powinien posiadać zabezpieczenie w postaci układu szybkiej reakcji przy pracy z nastawami bliskimi strefie stabilności układu regulacji.

5. POPRAWIONE PROGRAMY ALGORYTMÓW AUTOSTROJENIA

Programy zostały napisane w języku C z uwagi na umożliwienie stosunkowo prostej implementacji w regulatorach zrealizowanych w oparciu o różne typy mikroprocesorów. Są one opatrzone obszernymi komentarzami zapewniającymi wystarczającą przejrzystość i ułatwiającymi wprowadzanie ewentualnych poprawek.

Ze względu na konieczność zapewnienia odpowiedniej dokładności większość obliczeń wykonywana jest w arytmetyce z podwójną precyzją.

W stosunku do wersji opracowanej w ramach etapu 1 pracy programy zostały znacznie rozbudowane i zmienione.

Wprowadzane korekty miały na celu głównie korektę algorytmów w oparciu o wyniki badań i uściślenie metod pomiarowych.

```

/*****
*****/
/**** Wersja 1.21 programu symulacyjnego obiektu regulacji typu
****/
/**** opoznienie z inercja z regulatorem PID. Wyniki symulacji za-
****/
/**** pisywane sa do zbioru o rozszerzeniu .TAB. Obliczenia wykony-
****/
/**** wane sa na zmiennych typu double float. Po zakonczeniu ekspe-
****/
/**** rymentu wywoływany jest program prezentowania symulacji na wy-
****/
/**** kresie graficznym o zmienianych parametrach.
****/
/**** Dobor nastaw metoda Ziglera-Nicholsa polegajaca na wlaczeniu
****/
/**** w miejsce rewgulatora przekaźnika wykonujacego 10%-owe "klap-
****/
/**** niecia". Po wykonaniu 3-ch "klapniec" wlaczany jest podprogram
****/
/**** filtracji filtr_2() realizujacy transmitancje filtra pasmowo-
****/
/**** przepustowego o czestotliwosci charakterystycznej rownej cze-
****/
/**** stotliwosci "klapotania".Wowczas wywoływany jest program
****/
/**** zigler() wyliczajacy optymalne nastawy regulatora
****/
/**** Przy pomocy wersji 1.11 stwierdzono, ze niemożliwe jest wpro-
****/
/**** wadzenia ukladu w drgania, jesli uchyb nie jest bliski zeru.
****/
/**** w ver. 1.15 zmieniono wyglad wykresu na ekranie przystosowuja-
****/
/**** go do wymogow sprawozdania.
****/
/**** Proby wprowadzenia zaklocen stochastycznych. Zmieniony zostal
****/
/**** program zigler(), gdzie zaostrowane zostaly kryteria zakoncze-
****/
/**** nia doboru nastaw.Dodano wzmacnienie obiektu KO.
****/
/**** Optymalizacja czasowa programu.Proby auto-tuningu wg Z.P.
****/
/**** 19 grudnia 1992 *****/
*****/
/*****
*****/
#include<stdio.h>
#include<graphics.h>
#include<math.h>
#include<io.h>
#include<string.h>

```

```

#include<stdlib.h>
/*****/

/* parametry eksperymentu */
#define GLOBAL_TIME      3000          /* globalny czas eksperymentu */
#define CZEKAJ      2          /* opoznienie rozpozecia eksperymentu*/
#define START_ZAKLOCENIA_1      700    /*50*/ /* pojawienie sie zaklocenia */
#define START_ZAKLOCENIA_2      1500   /*2500/3600*/
#define START_DRUGIE_WZ      5001     /* zmiana wartosci zadanej */
#define START_IDENTYFIKACJI      4000  /* ilosc probkowan po jakiej nastapi */

/* start programu identyfikacji */
#define DOKLADNOSC_IDENT      0.001
#define IL_KLAPNIEC      141          /* ilosc zalaczen przekaznika ???*/
#define TAU      1.0                /* okres probkowania */
/*****/

/* STALE POMOCNICZE */
#define ZAKRES_MAX      10000.0
#define ZAKRES_MIN      -10000.0
#define PI      3.14
#define PLUS      1
#define MINUS      0
/*****/

/* wartosci sygnalow wejscowych */
#define PIERWSZE_WZ      0.1*ZAKRES_MAX /* wartosc poczatkowa */
#define DRUGIE_WZ      0.0
#define SKOK_1      0.1*ZAKRES_MAX      /* wartosci skokow zaklocajacych */
#define SKOK_2      0.0*ZAKRES_MAX
/*****/

/* parametry identyfikacji obiektu metoda Zieglera-Nicholsa */
#define PRZEKAZNIK      0.05*ZAKRES_MAX /* skoki wartosci podawane na obiekt regulacji */
#define WLK_ZAKL      0.02          /* amplituda zaklocen */
#define LEVEL      0.03            /* poziom przelaczania przekaznika 3% */
/*****/

/* nastawy regulatora PID */
#define KP      1.0
#define TI      60.0
#define TD      10.0
#define TFP      6.0
/*****/

```

```

*****/
/* parametry obiektu opoznienie z inercja */
#define TF      100.0      /* stala czasowa inercji */
#define DELAY   40        /* czas opoznienia */
#define KO      1.5       /* wzmacnienie obiektu
*/
/*****/
/* parametry czlonow calkujacych */
#define TII      2.0*((double)DELAY*TAU)
#define STALA    10000.0
/*****/
/* tablice wynikowe */
double ypop,a[DELAY];      /* dane chronione obiektul */
double sum,wejpop,delta_pop; /* dane chronione regulatora */
double kp,ti,td;
double ypp,yppp,xpp;      /* filtru pasmowo-przepustoweg
o */
/* pod
programu adaptacyjnego */
double ampax1,ampax2,ampax3;
double ampin1,ampin2,ampin3;
double pop_out,pop_delta;
double okres_oscl,okres_osc2,licznik_osc;
int ident_done;          /* odczekanie na bezza
kloceniowe wyjscie*/
/* z e
ksperymentu identyfikacyjnego */
/* chr
onione dane blokow calkujacych */
/* pro
gramu auto-tuningu wg Z.P. */
double se1,se2,se3,se4,sum_e1,sum_e2,sum_e3,sum_e4;
double in_pl,in_p2;
double wyl_kp;

double obiekt_1();      /* Symulacja obiektu typu poznienie z
inercja */
double pid();          /* Symulacja regulatora PID */
double filtr_2();      /* Filtr pasmowo-przepustowy wykorzystywany pr
zez */
/* algorytm auto-tunin
gu wg Ziglera-Nicholsa */
double calka();        /* Auto-tuning wg Z.P. */
double ograniczenie(); /* Podprogram ograniczania sygnalu do zakresu
*/
int zigler();          /* Auto-tuning wg Ziglera_Nicholsa */

#define ON      1        /* zbior otwarty do zapisu */
#define OFF 0        /* zbior nie otworzony */
#define NAZWA  20       /* Max. dlugosc nazwy zbioru */
int arg_c;
char *arg_v[NAZWA];
char name[NAZWA];

```

```

int dp,dd; /* dysk poczatkowy, dysk archiwalny */
FILE *out; /* nazwa otwieranego zbioru do zapisu */
*/
/*****/
*****/
/*****/
*****/
/* POCZATEK PROGRAMU GLOWNEGO */
/*****/
*****/
/*****/
*****/
void main(argc,argv)
    int argc;
    char *argv[];
{
    char c; /* pomocniczy klawisz */
*/
    int opened = OFF; /* zbior nie otworzony */
/*****/
*****/
/* zmienne dla autostrojzenia */
/* stan identyfikacji */
    int id = 0;
*/
    int lid = 0; /* licznik skokow identyfikacy
jnych i licznik */
    int ip = 0; /* probkowan identyfik
acyjnych, */
    int id_osc = 0; /* wstepne obliczanie TOSC */
    int tosc = 0; /* zgrubna stala czasowa filtr
u pasmowo-przep. */
    int znak = 0; /* znak wyjscia obiektu */
    int znak_pop = 0;
    double yo_as = 0.0; /* przechowywane wartosci star
towe auto-tuningu */
    double y_as = 0.0; /* odtwarzane po zakonczeniu i
dentyfikacji */
    double wz_as = 0.0;
    double offset_in,offset_out; /* przesuniecie wejścia obiekt
u, aby */
/* wyj
scie nie przekraczalo nastawianej */
/* war
tosci zadanej */
/*****/
*****/
    int i,t,tp,j,jj;
    int cg,cy,cb,cr,cs; /* wlaczenie/wylaczenie
e kolorowych ch-k */
/* Gre
en, Yellow, Blue, Red. Szary */
    double wz,e,yo,y,ypom; /* charakterystyczne zmienne s
ymulacji */
    double wzp,ep,yop,yp,ypomp; /* zmienne z poprzedniego krok
u */

```



```

        double przechowane_zakl;          /* przechowywana wartosc zaklo
cen, aby w */                               /* kaz
dym probkowaniu bylo nowe zaklocenie */   /* sto
chastyczne, a nie scalkowywane z po- */   /* prz
ednich krokow */
        long int max,may;                  /* maksymalna wartosc
jakiegokolwiek sygnalu */                 /* pot
rzebna do ustalenia wygladu wykresu (cho-*/ /* dzi
o wykorzystanie maksymalne ekranu */
        long int mg,my,mb,mr,ms;          /*wzmocnienia poszczegolnych k
olorow*/
        int g_driver,g_mode,max_x,max_y;
        double zakl = 0;                  /* wartosc sygnalu zak
locajacego */
        char tekst[70];                   /* string tworzonych napisow na wy
kresie w */                               /* try
bie graficznym */
        long int kursor = 0L;              /* polozenie znacznika czytani
a zbioru */
        kp = KP; ti = TI; td = TD;        /* Inicjalizacja nastaw */
        cg = cy = cb = cr = cs = 0;
        max = 0;
/*****/
/*****/
        wz = 0.0;
        y = e = yo = yp = ep = yop = ypom = ypomp = 0.0;
                                           /* inicjalizacja zmiennych pro
gramu auto-tuningu */
                                           /* metody Ziglera-Nicholsa */
        ampax2 = ampax3 = ampin2 = ampin3 = 1.0;
        ampax1 = ampin1 = 0.0;
        pop_out = pop_delta = 0.0;
        okres_oscl = okres_osc2 = licznik_osc = 0.0;
        ident_done = MINUS;
        offset_in = offset_out = 0.0;
        lid = 0;
                                           /* inicjalizacja zmiennych fil
tru*/
        ypp = yppp = xpp = 0.0;
        ypop = 0.0;
                                           /* inicjalizacja zmiennych reg
ulatora PID */
        wejpop = 0.0;
        sum = 0.0;
        delta_pop = 0.0;
                                           /* inicjalizacja zmiennych aut
o-tuningu metoda Z.P.*/

```

```

    se1=se2=se3=se4=sum_e1=sum_e2=sum_e3=sum_e4=0.0;
    in_p1 = in_p2 = 0.0;
    wyl_kp = 0.0;
    t = tp = 0;

/* zerowanie tablicy symulacji
opoznienia */
    for(i=0; i<=DELAY-1; i++)
        a[i] = 0.0;
/*****
*****/
    /* zbior juz otwarty, ustawianie kursora na koniec */
    if(opened == ON)
        fseek(out,kursor,SEEK_END);
    /* pierwsze otwieranie zbioru do zapisu */
    if(opened == OFF)
    {
        arg_c = argc;
        if(argc > 1 )
            strcpy(arg_v[1],argv[1]);
        if(otworz_zapis() == 1)
            printf("Nie mozna otworzyc zbioru do zapisu !!
!");
        opened = ON;
    }
/*****
*****/
    randomize(); /* uruchomienie generatora liczb losowych */
/*****
*****/
/*****
*****/
    /* petla oczekiwania na start eksperymentu */
    for(i=0; i<=CZEKAJ; i++,t++)
    {
        zapisz(t,wz,e,y,yo,ypom);
    }
/*****
*****/
    /* rozpozecie eksperymentu, ustawienie wartosci poczatkowych
*/

    wz = ograniczenie(PIERWSZE_WZ);
/*****
*****/
/*****
*****/
    /* glowna petla obliczeniowa */
    for(i=0; t<=GLOBAL_TIME; i++,t++)
    {
/*****
*****/
        /* wlaczanie poszczegolnych funkcji po: */
        if(t == START_ZAKLOCENIA_1)
            zakl = SKOK_1;
        if(t == START_IDENTYFIKACJI)

```

```

        id = 1;
    if(t == START_ZAKLOCENIA_2)
        zakl = SKOK_2;
    if(t == START_DRUGIE_WZ)
        wz = DRUGIE_WZ;
    przechowane_zakl = zakl;
/*****/
    /* wprowadzane zaklocenie stochastyczne !!!!!!!!!!!!!!!!!!!!!!! */
    zakl = zakl + WLK_ZAKL*((double)random((int)ZAKRES_MAX) - (0.5
*(ZAKRES_MAX)));
/*****/
    switch(id)
    {
    case 0: /* id = 0 - normalna regulacja
PID */
/*****/
        {
            /* petla zamknieta regulacji z obiektem opozni
enie z inercja */
            e = ograniczenie(wz - y);
            y = obiekt_1(ograniczenie((yo=pid(e))+zakl));
            yo_as = yo; /* + z
            y_as = y;
            wz_as = wz;
            ypom = y;
            calka(e);
            break;
        }
    case 1: /* id = 1 - start identyfikacj
i */
/*****/
        {
            y = obiekt_1(zakl + (yo=(yo_as - PRZEKAZNIK)))
;
            ip++;
            ypom = y;
            if(y > y_as)
                znak_pop = znak = PLUS;
            else
                znak_pop = znak = MINUS;
            if( fabs(y-y_as) < LEVEL*ZAKRES_MAX)
                break;
            else
            {
                id = 2;
                lid = 1;
                break;
            }
        }
    case 2: /* realizacja generatora "w go

```

```

re" */
*****/
/* W pierwszym "klapnieciu" przekaznika ustalana jest
maksymalna */
ziom prze- */
/* lacznania przekaznika */
if(lid == 1)
{
    /* wyszukanie minimum */
    if((y-pop_out) >= 0.0 && (pop_delta <
0.0))
    {
        offset_out = fabs((y_as-pop_out
t)*0.5);
        offset_in = PRZEKAZNIK*0.5;
    }
    pop_delta = y - pop_out;
    pop_out = y;
}

if(id_osc == 0) /* wyjscie y filtrowan
e dla id_osc = 1 */
{
    ypom = y;
    if(lid == 3)
        tosc++;
}
else
    ypom = filtr_2(y, (double)tosc);
ip++;
if(y > (y_as - offset_out))
    znak = PLUS;
else
    znak = MINUS;
if (znak == znak_pop)
    break;
else
{
    znak_pop = znak;
    id = 3;
    break;
}
}
case 3: /* realizacja generatora "w do
1" */
*****/
{
    y = obiekt_1(zakl + (yo=(yo_as - PRZEKAZNIK -
offset_in)));
    if(id_osc == 0)

```

```

        {
            ypom = y;
            if(lid == 3)
                tosc++;
        }
        else
            ypom = filtr_2(y,(double)tosc);
        ip++;
        if(y > (y_as - offset_out))
            znak = PLUS;
        else
            znak = MINUS;
        if (znak == znak_pop)
            break;
        else
        {
            znak_pop = znak;
            lid++;
            if( lid >= IL_KLAPNIEC)
            {
                id = 4;
                break;
            }
            else
            {
                id = 2;
                if(lid == 4)
                    id_osc = 1;
            }
        }
        break;
    }
}
case 4: /* przerwanie identyfikacji po
IL_KLAPNIEC */
/******
*****/
{
    y = y_as;
    yo = yo_as; /* - zakl;*/
    wz = wz_as;
    id = 0;
    wejpop = ograniczenie(wz-y);
    delta_pop = 0.0;
    sum = 0.0;
    break;
}
}
/******
*****/
/* wywoływanie programu auto-tuningu w czasie realizacji gener
atora, */
/* lecz juz po włączeniu filtru pasmowo-przepustowego, tzn. po
3 klap*/
if(id_osc != 0 && (id == 2 || id == 3))

```

```

        if(zigler(ypom) == PLUS)                                /* bylo zigler
(y) */
        {
            y = y_as;
            yo = yo_as;                                        /* - z
akl;*/
            wz = wz_as;
            sum = 0.0;
            id = 0;
            wejpop = ograniczenie(wz-y);
            delta_pop = 0.0;
        }
/*****
*****/
        zakl = przechowane_zakl;
/*****
*****/
        /* wyszukanie maksimum dla wykresow */
        if(max <= (int)fabs(wz))
            max = (int)fabs(wz);
        if(max <= (int)fabs(e))
            max = (int)fabs(e);
        if(max <= (int)fabs(y))
            max = (int)fabs(y);
        if(max <= (int)fabs(yo))
            max = (int)fabs(yo);
        if(max <= (int)fabs(ypom))
            max = (int)fabs(ypom);
/*****
*****/
        zapisz(t,wz,e,y,yo,ypom);                                /* zapis zmiennych do zbioru */
/
    }
/*****
*****/
/*****
*****/
        /* zapis koncowy parametrow eksperymentu do otwartego zbioru */
/
        fprintf(out,"Nastawy regulatora:\n\tKp = %3.2f,\tTi = %3.2f,\t
Td = %3.2f (Tf= %3.2f)\n",KP,TI,TD,TFP);
        fprintf(out,"Parametry obiektu:\n\tTopz = %d,\tTf = %3.2f\n",D
ELAY,TF);
        fprintf(out,"Parametry sygnalow:\n\tWZ = %5.2f,\tXz = %5.2f\n"
,PIERWSZE_WZ,SKOK_1);
/*****
*****/
        /* zamknij_zapis();*/
        /* otworz_odczyt();*/
/*****
*****/
        /* zakonczenie zapisu eksperymentu i wizualizacja */
        /* ustawienie dysku ze zbiorami .bgi */
        setdisk(dp);
        detectgraph(&g_driver,&g_mode);

```

```

if(g_mode == EGAHI)
    g_mode = EGALO;
if(g_driver == CGA)
    g_mode = CGAHI;
initgraph(&g_driver,&g_mode,"");
setfillstyle(0,0);
settextstyle(SMALL_FONT,HORIZ_DIR,USER_CHAR_SIZE);

setusercharsize(2,1,3,2);
bar(0,0,max_x=getmaxx(),max_y=(getmaxy()-25));
/* ustawienie 25 pixli na komentarze na wykres
ie */
fseek(out,kursor,SEEK_SET);
/* ustalenie początkowych wzmocnień */
max = max*2;
mg = mb = my = mr = may = ms = max;
j = 1;
fseek(out,kursor,SEEK_SET);
c = NULL;
/*****
*****/
/* REALIZACJA WYKRESU GRAFICZNEGO */
/*****
*****/
do
{
    clearviewport();
    setcolor(WHITE);
/* rysowanie osi x,y */
    setlinestyle(SOLID_LINE,OXFFFF,NORM_WIDTH);
    line(0,0,0,max_y);
    line(0,max_y/2,max_x,max_y/2);
/*****
*****/
    if(g_driver == EGA || g_driver == VGA) /* ustawianie
rodzaju linii */
        /* Opcja THICK_WIDTH daje szerokie,grube linie
na ekranie */
        /* beznadziejnie wychodzące na wydrukach */
        /*
        setlinestyle(SOLID_LINE,OXFFFF,THICK_WIDTH);
*/
        setlinestyle(SOLID_LINE,OXFFFF,NORM_WIDTH);
/*****
*****/
/* rysowanie wykresu */
    if(c == 'q' || c == 'Q')
        break;
    if(c == '+') /* Rozszerzani
e i skracanie wykresu */
        j--;
    if(c == '-')
        j++;
    if(c == 'y' || c == 'Y') /* Wylaczanie poszczegolny
ch kolorow */
        cy = !cy;

```

```

    if(c == 'g' || c == 'G')
        cg = !cg;
    if(c == 'b' || c == 'B')
        cb = !cb;
    if(c == 'r' || c == 'R')
        cr = !cr;
    if(c == 's' || c == 'S')
        cs = !cs;
    if(c == 'w' || c == 'W')
    {
/* zmiana wzmocnien w gore */
        if(cy == 0 && my > may/10)
            my -= may/10;
        if(cg == 0 && mg > may/10)
            mg -= may/10;
        if(cb == 0 && mb > may/10)
            mb -= may/10;
        if(cr == 0 && mr > may/10)
            mr -= may/10;
        if(cs == 0 && ms > may/10)
            ms -= may/10;
    }
    if(c == 'z' || c == 'Z')
    {
/* zmiana wzmocnien w dol */
        if(cy == 0)
            my += may/10;
        if(cg == 0)
            mg += may/10;
        if(cb == 0)
            mb += may/10;
        if(cr == 0)
            mr += may/10;
        if(cs == 0)
            ms += may/10;
    }
    if(c == 'n' || c == 'N')
        ;
    else
    {
        kursor = 0L;
    }
    fseek(out, kursor, SEEK_SET);
    fscanf(out, "%d%lf%lf%lf%lf%lf", &tp, &wzp, &ep, &yp, &yop, &
ypom);
    /******
    /* wypisanie danych pod wykresem */
    sprintf(tekst, "Regulator: Kp=%3.2f/%3.2f;Ti=%3.2f/%3.2
f;Td=%3.2f/%3.2f;Obiekt: To=%3.2f;Tf=%3.2f", KP, kp, TI, ti, TD, td, DELAY*TA
U, TF);

    outtextxy(0, max_y+8, tekst);
    sprintf(tekst, " Sygnaly: Sp=%3.2f%%; Zaklocenia= %3.2f
%%;%3.2f%%; Amplituda.sygn.ident=%3.2f%%", PIERWSZE_WZ*100.0/ZAKRES_MAX
, SKOK_1*100.0/ZAKRES_MAX, SKOK_2*100.0/ZAKRES_MAX, PRZEKAZNIK*100.0/ZAKR

```



```

ES_MAX);
    outtextxy(0,max_y+17,tekst);
/*****
*****/
    for(i = 0; i <= max_x; i++)
    {
        if(j >= 1)
        {
            for(jj=1; jj<=j; jj++)
                if(fscanf(out,"%d%lf%lf%lf%lf%lf",&t,&wz,&e,&y,&yo,&ypom) == EOF || t >= GLOBAL_TIME)
                    goto the_end;
        }
        else
        {
            jj = abs(j)+2;
            if(i%jj == 0)
                if(fscanf(out,"%d%lf%lf%lf%lf%lf",&t,&wz,&e,&y,&yo,&ypom) == EOF || t >= GLOBAL_TIME)
                    goto the_end;
        }
        setcolor(WHITE);
        sprintf(tekst,"%d",t);
        if(i == max_x/4)
            outtextxy(max_x/4,max_y/2+10,tekst);
        if(i == max_x/2)
            outtextxy(max_x/2,max_y/2+10,tekst);
        if(i == max_x*3/4)
            outtextxy(max_x*3/4,max_y/2+10,tekst);
        if(i == max_x)
            outtextxy(max_x-40,max_y/2+10,tekst);
        sprintf(tekst,"%d%",max/2*100/(int)ZAKRES_MA
X);
        outtextxy(0,0,tekst);
        sprintf(tekst,"%d%",max/4*100/(int)ZAKRES_MA
X);
        outtextxy(0,max_y/4,tekst);
        sprintf(tekst,"%d%",max/2*100/(int)ZAKRES_MA
X);
        outtextxy(0,max_y-5,tekst);
        sprintf(tekst,"%d%",max/4*100/(int)ZAKRES_MA
X);
        outtextxy(0,max_y*3/4,tekst);
        if(t >= GLOBAL_TIME-1)
            break;

        if(cr == 0)
        {
            if(g_driver == EGA || g_driver == VGA)
                setcolor(RED);
            else
                setlinestyle(DOTTED_LINE,0xFFFF
F,NORM_WIDTH);
            line(i,(max_y/2-((long int)ep*max_y)/m
r),i,(max_y/2-((long int)e*max_y)/mr));

```

```

    }
    if(cg == 0)
    {
        if(g_driver == EGA || g_driver == VGA)
            setcolor(GREEN);
        else
            setlinestyle(CENTER_LINE,0xFFFF
F,NORM_WIDTH);
        line(i,(max_y/2-((long int)yp*max_y)/m
g),i,(max_y/2-((long int)y*max_y)/mg));
    }
    if(cy == 0)
    {
        if(g_driver == EGA || g_driver == VGA)
            setcolor(YELLOW);
        else
            setlinestyle(DASHED_LINE,0xFFFF
F,NORM_WIDTH);
        line(i,(max_y/2-((long int)wzp*max_y)/
my),i,(max_y/2-((long int)wz*max_y)/my));
    }

    if(cb == 0)
    {
        if(g_driver == EGA || g_driver == VGA)
            setcolor(LIGHTBLUE);
        else
            setlinestyle(DOTTED_LINE,0xFFFF
F,NORM_WIDTH);
        line(i,(max_y/2-((long int)yop*max_y)/
mb),i,(max_y/2-((long int)yo*max_y)/mb));
    }
    if(cs == 0)
    {
        if(g_driver == EGA || g_driver == VGA)
            setcolor(LIGHTGRAY);
        else
            setlinestyle(DASHED_LINE,0xFFFF
F,NORM_WIDTH);
        line(i,(max_y/2-((long int)ypomp*max_y
)/ms),i,(max_y/2-((long int)ypom*max_y)/ms));
    }

the_end:
        tp = t;wzp = wz;ep = e;yp = y;yop = yo;ypomp = ypom;
    }
    kursor = ftell(out);
}
while((c=getch()) /*== '+' || c == '-' || c*/ != 'q' || c != '
Q');
    clearviewport();
    closegraph();
/*****
*****/
    setdisk(dd);

```

```

        zamknij_odczyt();
/*      printf("Pierwotne nastawy regulatora:\n\tKp = %3.2f,\tTi = %3.
2f,\tTd = %3.2f (Tf= %3.2f)\n",KP,TI,TD,TFP);*/
/*      printf("Wyliczone nastawy regulatora:\n\tKp = %3.2f,\tTi = %3.
2f,\tTd = %3.2f\n",kp,ti,td);          */
/*      printf("Parametry obiektu:\n\tTopz = %d,\tTf = %3.2f\n",DELAY,
TF);          */
/*      printf("Wcisnij jakis klawisz...");          */
/*          */
/*      getch();          */
/*          */
        exit(0);
}

```

```

/*****
*****/
/*****          P O D P R O G R A M Y   I   F U N K C J E          *****/
*****/
/*****
*****/

```

```

/*****
*****/
/* podprogram doboru optymalnych nastaw regulacji *****/
*****/
/*****
*****/

```

```

int zigler(akt_out)
    double akt_out;
{
    extern double kp,ti,td;
    extern double ampax1,ampax2,ampax3;
    extern double ampin1,ampin2,ampin3;
    extern double pop_out,pop_delta;
    extern double okres_osc,licznik_osc;
    extern int ident_done;
    extern double offset;

    double akt_delta,wart_sr;
        /* wyszukiwanie wartosci max amplitudy i czestotliwosc
i charak.*/
    akt_delta = akt_out - pop_out;
    wart_sr = ((ampax1-ampin1)*0.5)+ampin1;

    if((ident_done==PLUS)&&((akt_out-wart_sr)*(pop_out-wart_sr) <=
0.0))
        return PLUS;

    if((akt_delta > 0.0) && (pop_delta >= 0.0))
    {
        pop_delta = akt_delta;
        pop_out = akt_out;
        licznik_osc++;
        return MINUS;
    }
}

```

```

    }
    if((akt_delta < 0.0) && (pop_delta <= 0.0))
    {
        pop_delta = akt_delta;
        pop_out = akt_out;
        licznik_osc++;
        return MINUS;
    }
    if((akt_delta >= 0.0) && (pop_delta < 0.0))
    {
        /* znaleziony min */
        ampin3 = ampin2;
        ampin2 = ampin1;
        ampin1 = pop_out;
    }
    if((akt_delta <= 0.0) && (pop_delta > 0.0))
    {
        /* znaleziony max */
        ampax3 = ampax2;
        ampax2 = ampax1;
        ampax1 = pop_out;
        okres_osc2 = okres_osc1;
        okres_osc1 = licznik_osc;
        licznik_osc = 0.0;
    }
    pop_delta = akt_delta;
    pop_out = akt_out;
    licznik_osc++;

    if(fabs((ampax1-ampin1)-(ampax2-ampin2)) > DOKLADNOSC_IDENT*ZA
KRES_MAX)
        return MINUS;
    if(fabs((ampax1-ampin1)-(ampax3-ampin3)) > DOKLADNOSC_IDENT*ZA
KRES_MAX)
        return MINUS;
    if(fabs(okres_osc1-okres_osc2) > 2*TAU)
        return MINUS;
    if((ampax2==ampin2) || (ampax3==ampin3))
        return MINUS;
    else
        {
            /* Nastawy wg kryterium 0% prz
eregulowania */
            /*
            *0.95/1.4; */
            /*
            */
            /*
            */
            /* Nastawy wg Ziglera-Nicholsa
            */
            kp = 0.6*4*2*PRZEKAZNIK/(PI*(ampax1-ampin1));
            ti = 0.5*okres_osc1*TAU;
            td = 0.125*okres_osc1*TAU;
            ident_done = PLUS;
            return MINUS;
            /* UWAGA !!!!!!!!!!!!! */
        }

```

```

    }
}
/*****
*****/
/* algorytm obiektu typu opoznienie z inercja *****/
*****/
/*****
*****/
double obiekt_1(xf)
    double xf;
{
    extern double ypop;
    extern double a[];
    int p;
    double yf;
    double tf = TF;
    double tautf = TAU+TF;

    for(p=0; p<=DELAY-1; p++)
        a[p] = a[p+1];
    a[DELAY-1] = xf;

    yf = ograniczenie(((a[0]*TAU*KO)+ypop*tf)/(tautf));

    ypop = yf;
    return(yf);
}
/*****
*****/
/* algorytm filtru pasmowo-przepustowego *****/
*****/
/*****
*****/
double filtr_2(xf,tf)
    double xf,tf;
{
    extern double ypp,yppp,xpp;
    double yf,tff;
    tff = tf*TAU/(2*PI);

    yf = ((2*ypp*(1+(tff/TAU))-yppp*(tff/TAU)+2*(xf- xpp))*(tff/TAU))/
    (1+2*(tff/TAU)+(tff*tff)/(TAU*TAU));

    yf = ograniczenie(yf);

    yppp = ypp;
    ypp = yf;
    xpp = xf;
    return(yf);
}
/*****
*****/
/* Algorytm regulatora P I D *****/
*****/
/*****
*****/

```

```

*****/
double pid(wej)
    double wej;
{

double wyj;
double delta_wej=0.0;
extern double sum,wejpop,delta_pop;
extern double kp,ti,td;
double tautfp = (TFP+TAU);
double tfp = TFP;

/*      sum = sum + wejpop;*/
    delta_wej = ((wej-wejpop)*TAU + delta_pop*tfp)/tautfp;

    wyj = kp*wej + (kp*sum*TAU) /ti + (kp*td/TAU)*delta_wej;

    wyj = ograniczenie(wyj);

    sum = sum + wej;
    wejpop = wej;
    delta_pop = delta_wej;
return(wyj);
}
/*****
*****/
/* algorytm ukladu auto-tuningu wersja Z.P. *****/
*****/
/*****
*****/
double calka(in)
    double in;
{

    extern double se1,se2,se3,se4,sum_e1,sum_e2,sum_e3,sum_e4;
    extern double in_p1,in_p2;
    extern double kp,ti,td;
    extern double wyl_kp;
    double ek = 0.0,tii;
    double delta_kp,delta_ti,delta_td;
    double taut = 0.16*TII + TAU;
    double tli = 0.16*TII;

    tii = TII;

    in_p1 = (in*TAU + in_p1*tli)/taut;
    in_p2 = (in_p1*TAU + in_p2*tli)/taut;

    ek = (in_p2/STALA)-1.5*se1-3.250*se2-1.9*se3-se4;

    sum_e1 = sum_e1 + ek;
    se1 = (TAU*sum_e1)/tii;

    sum_e2 = sum_e2 + se1;
    se2 = (TAU*sum_e2)/tii;

```

```

        sum_e3 = sum_e3 + se2;
        se3 = (TAU*sum_e3 )/tii;

        sum_e4 = sum_e4 + se3;
        se4 = (TAU*sum_e4 )/tii;

        delta_kp = 1.5*se4*se4 - 1.1*se3*se3 + 1.65*se2*se2 - 2.5*se1*
        se1 + 0.41*ek*ek;
        wyl_kp = wyl_kp + delta_kp*10;

/*      printf("delta_kp = %lf\t\t wyl_kp = %lf\n",delta_kp,wyl_kp);*/
/*      getch();*/

        return(wyl_kp);
}
/******
*****/
/* Pomocnicza funkcja wydruku zmiennych na ekranie *****/
*****/
/******
*****/
print_wynik(time,in1,in2,out1,out2)
        int time;
        double in1,in2,out1,out2;
{
        printf("%4d   %5.2f   %5.2f   %5.2f   %5.2f\n",time,in1,in2,
out1,out2);
        if(time%20 == 0)
                getch();
}
/******
*****/
/* otwieranie zbioru do zapisu *****/
*****/
/******
*****/
otworz_zapis()
{
extern int arg_c;
extern char *arg_v[];
extern dp,dd;
extern char name[];
char *roz = ".TAB";
char *pkt;
extern FILE *out;
int i;

        /* WYBOR DYSKU */
        dp = getdisk();
        dd = setdisk();
        printf(" Wybierz dysk: ");
        for(i='A'; i<=('A'-1+dd); i++)
                printf(" %c,",i);

```

```

printf("...?\n");
if((dd=toupper(getch())) >= 'A' && dd <='F')
    setdisk(dd-='A');
else
    dd = dp;
    /* nazwa zbioru do zapisu */
    if(arg_c > 1)
        strcpy(name, arg_v[1]);
    else
    {
        printf(" Podaj nazwe zbioru ... ");
        scanf("%12s", &name);
    }
   strupr(name);
    strcpy(arg_v[1], name);
    if((pkt=strchr(name, '.')) == NULL)
        strcat(name, roz);
    else
        strcpy(pkt, roz);
    strcpy(arg_v[1], name);
    if((out=fopen(name, "wt+")) == NULL)
        return(1);
    printf(" Zapis do zbioru %c: \\%s\n", dd+'A', name);
}
/*****
*****/
/* ZAMKNIECIE ZAPISU DO ZBIORU *****/
/*****
*****/
zamknij_zapis()
{
extern int dp;
extern FILE *out;
    fclose(out);
    /* setdisk(dp); */
}
/*****
*****/
/* OTWORZENIE ZBIORU DO ODCZYTU *****/
/*****
*****/
otworz_odczyt()
{
extern char name[];
extern int dp;
extern FILE *out;
    if((out=fopen(name, "r")) == NULL)
        return(1);
    printf(" Odczyt ze zbioru %c: \\%s\n", dp, name);
}
/*****
*****/
/* ZAMYKANIE CALOSCI ZBIOROW *****/

```



```

*****/
/* *****/
*****/
zamknij_odczyt()
{
extern int dp;
extern FILE *out;
    fclose(out);
    setdisk(dp);
}
/* *****/
*****/
/* Zapisanie zmiennych do zbioru w postaci sformatowanej *****/
*****/
/* *****/
*****/
zapisz(time, lx1, lx2, lx3, lx4, lx5)
    int time;
    double lx1, lx2, lx3, lx4, lx5;
{
    extern FILE *out;
    fprintf(out, "%d\t%5.5f\t%5.5f\t%5.5f\t%5.5f\t%5.5f\n", time, lx1
, lx2, lx3, lx4, lx5);
}
/* *****/
*****/
/* Funkcja ograniczenia zmiennej typu float do zakresu *****/
*****/
/* *****/
*****/
double ograniczenie(wej)
    double wej;
{
    if(wej > ZAKRES_MAX)
        wej = ZAKRES_MAX;
    if(wej < ZAKRES_MIN)
        wej = ZAKRES_MIN;
    return(wej);
}
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/
/* *****/
*****/

```

6. WNIOSKI

Badania potwierdziły poprawność koncepcji realizacji algorytmów autostrojzenia. Stopień zaawansowania prac w dwu podstawowych algorytmach autostrojzenia tj. strojeniu zgrubnym i strojeniu dokładnym jest nierówny. Znacznie lepiej rozpoznane i przebadane jest działanie algorytmu autostrojzenia zgrubnego, którego praca nie będzie już specjalnych zastrzeżeń i wątpliwości tak, że wydaje się iż można w tym zakresie przystąpić do prac związanych z jego implementacją w zastosowaniach przemysłowych. Mógłby on być stosowany niezależnie od algorytmu autostrojzenia dokładnego. Byłby to algorytm jednorazowego dostrojzenia się regulatora na żądanie operatora, zgłaszane np. za pomocą przycisku na pulpicie operacyjnym. Algorytmy o podobnym działaniu są często spotykane w rozwiązaniach firm zagranicznych wyposażonych w algorytmy autostrojzenia.

Algorytm autostrojzenia dokładnego w świetle uzyskanych wyników wydaje się być bardzo obiecujący, stwarza jednak szereg nie rozwiązanych do tej pory problemów, dla przezwyciężenia których potrzebne są dalsze prace projektowe i badawcze. Są to takie problemy jak : znalezienie sposobu przeciwdziałania zjawisku interakcji strojenia parametrów regulatora, opracowanie skutecznej metody wprowadzania korekt nastaw dynamicznych, opracowanie metod autostrojzenia przy różnych wskaźnikach jakości regulacji, zapewnienie odpowiedniej dynamiki w procesie autostrojzenia itp. Celową byłaby kontynuacja pracy w tym zakresie w miarę posiadanych przez Instytut środków.