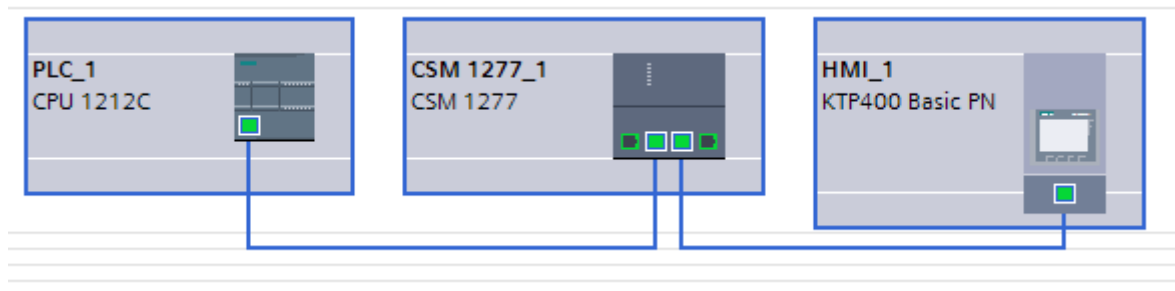


Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie Laboratorium Aparatury Automatykacji			
Numer i temat ćwiczenia: Ćwiczenie 3. Sterowanie układem lewitacji powietrznej.			
Grupa ćwiczeniowa: Wtorek 17:00-19:15, Zespół: 3			
Lp.	Imię i nazwisko	Ocena	Podpis
1.	Katarzyna Wątorska		
2.	Sonia Wittek		
3.	Karolina Świerczek		
Data wykonania ćwiczenia: 26.03.2019			

1. Schemat i opis konfiguracji systemu:

Celem ćwiczenia było skonfigurowanie i przetestowanie systemu sterowania lewitacji powietrznej, bazującego na sterowniku PLC SIEMENS S7 1200 z panelem operatorskim. Dodatkowo wykorzystaliśmy laserowy czujnik odległości, wentylator napędzany silnikiem elektrycznym DC oraz przezroczystą rurę, w której porusza się piłeczka ping-pongowa.

Konfigurację rozpoczęliśmy od dodania do utworzonego w TIA PORTAL projektu sprzętu znajdującego się na stanowisku laboratoryjnym, tj. jednostki centralnej CPU 1212C, switcha sieciowego CSM 1277 oraz panelu operatorskiego KTP-400 Basic PN. Następnie elementy te odpowiednio skonfigurowaliśmy i połączyliśmy zgodnie z instrukcją, czego efekt przedstawia poniższy schemat:


















Rysunek 1. Połączone elementy sterownika PLC.

2. Program sterowania logicznego:

Stworzyliśmy algorytm sterowania logicznego, zaczynając od zdefiniowania nazw zmiennych w tabeli z nazwami symbolicznymi PLC tags oraz ich zaadresowania, w wyniku czego otrzymaliśmy poniższy zestaw zmiennych:

PLC tags

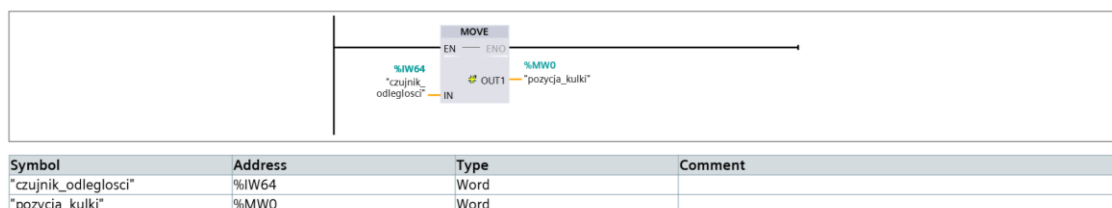
PLC tags						
	Name	Data type	Address	Retain	Visible in HMI	Accessible from HMI
	czujnik_odleglosci	Word	%IW64	False	True	True
	pozycja_kulki	Word	%MW0	False	True	True
	PWM_Start	Bool	%M128.0	False	True	True
	Zakres_ok	Bool	%M128.1	False	True	True
	Poza_zakresem	Bool	%I0.0	False	True	True
	Stan_PWM	Word	%MW2	False	True	True
	Zadany_PWM	Word	%MW4	False	True	True
	Wartosc_PWM	Word	%QW1000	False	True	True
	Manual_ON	Bool	%M128.2	False	True	True
	War_zadana	Real	%MD6	False	True	True
	Ster_manual	Real	%MD10	False	True	True
	Obroty	DWord	%ID1004	False	True	True
	RPM	DWord	%MD14	False	True	True
	blad_poz	Real	%MD20	False	True	True
	Tag_1	Bool	%Q0.0	False	True	True

Rysunek 2. Zdefiniowane i zaadresowane zmienne.

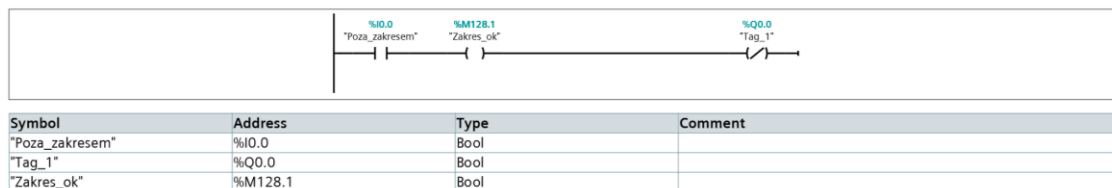
Utworzone zmienne wykorzystaliśmy następnie do realizacji algorytmu logicznego, szczytującego dane z laserowego czujnika odległości. Sygnał wyjściowy podpięliśmy do wejścia analogowego AI0.0 sterownika (o adresie %IW64). Wyjście dyskretne czujnika podpięliśmy do wejścia dyskretnego sterownika DI0.0 (o adresie %I0.0). Osiąga ono stan wysoki, jeśli odległość czujnika od mierzonego przedmiotu nie mieści się w zakresie pomiarowym (10 cm – 100 cm).

Używając języka drabinkowego, zaprogramowaliśmy algorytm w pliku źródłowym bloku organizacyjnego OB1:

Network 1:

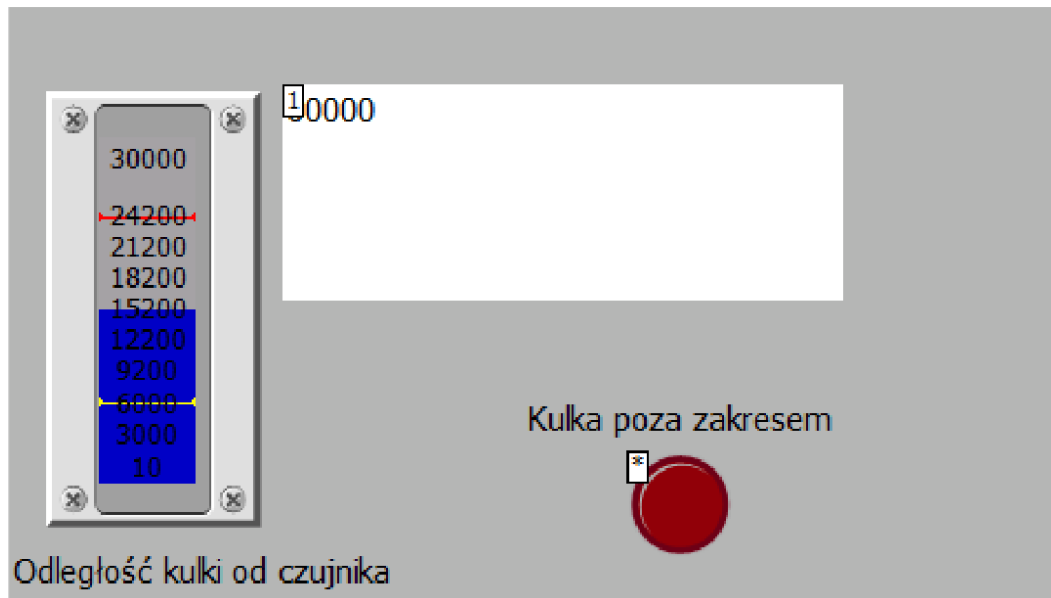


Network 2:



Rysunek 3. Algorytm sterowania zaprogramowany w języku drabinkowym.

Na panelu operatorskim umieściliśmy pole cyfrowe i bargraf, określające pozycję kulki oraz lampkę sygnalizacyjną, pokazującą, że piłeczka znajduje się poza zakresem pomiarowym czujnika.



Rysunek 4. Zrzut ekranu panelu operatorskiego.

Zmienną „pozycja_kulki” powiązałyśmy z bargrafem oraz polem cyfrowym, a zmiennej „Zakres_ok” przypisałyśmy lampkę sygnalizacyjną, zapalającą się na czerwono, gdy piłka znajdzie się poza zakresem. Następnie odpowiednio zdefiniowałyśmy podpisy do tych elementów.

3. Pomiary odległości piłeczki od czujnika

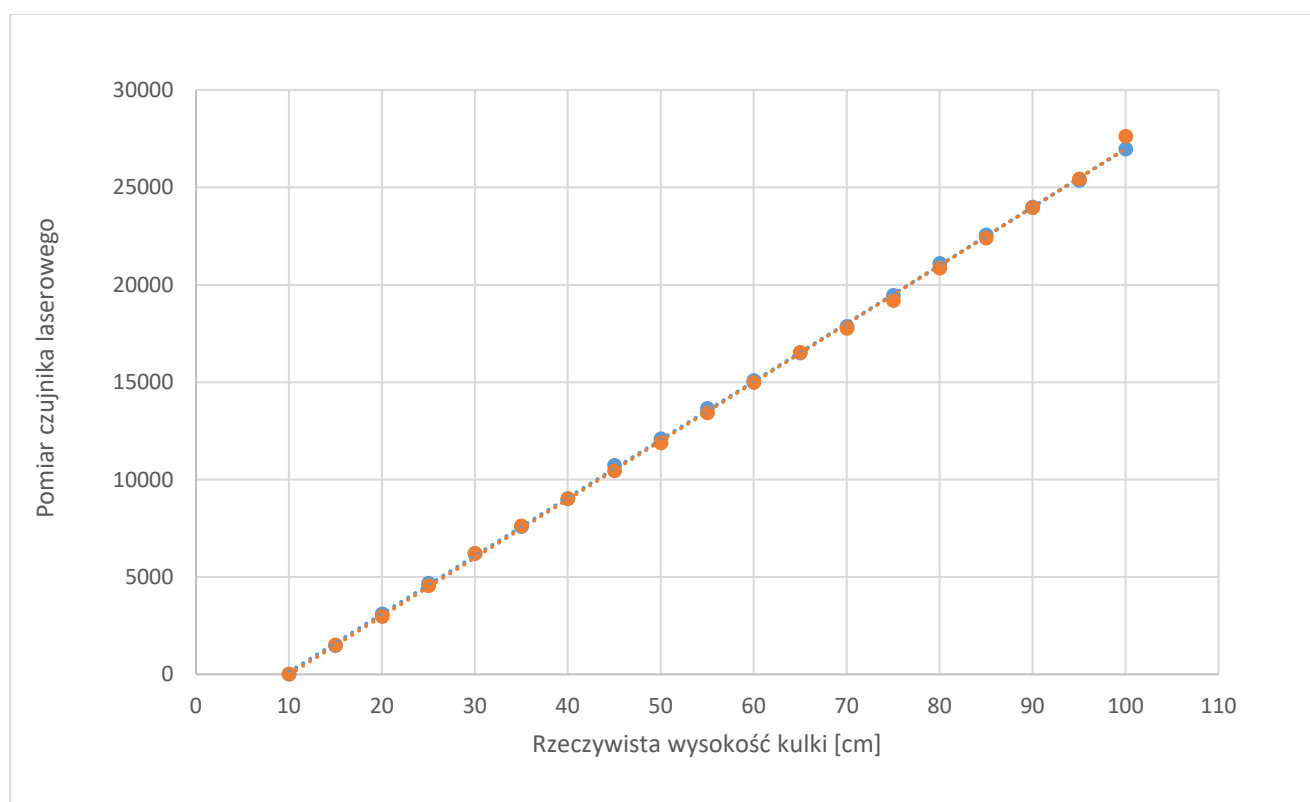
Następnym krokiem było wgranie algorytmu na sterownik i sprawdzenie jego działania za pomocą piłeczki przymocowanej do linki. Trzymając sznurek w ręce, zmienialiśmy pozycję piłki w całym zakresie ruchu z krokiem 5 cm. Pomiar zaczęłyśmy od sprawdzenia, czy lamka sygnalizacyjna na panelu operatorskim zaświeci się, jeśli piłeczka znajdzie się mniej niż 10 cm lub więcej niż 100 cm od czujnika. Potem zaczynając od 10 cm, zmienialiśmy pozycję piłeczki i notowałyśmy wyniki wyświetlane w polu cyfrowym jako pomiar 1. Analogiczny pomiar przeprowadziłyśmy z góry na dół zakresu pomiarowego notując wyniki jako pomiar 2. Otrzymane wyniki prezentuje tabela:

Tabela 1: Pomiar wartości wysokości kulki oraz pomiary wykonane przez laserowy czujnik odległości.

Wysokość Kulki [cm]	Pomiar 1	Pomiar 2
10	0	0
15	1467	1501
20	3105	2969
25	4675	4539
30	6211	6211
35	7576	7610
40	9010	9010

45	10716	10443
50	12082	11877
55	13652	13413
60	15085	14983
65	16484	16518
70	17849	17757
75	19454	19181
80	21092	20853
85	22559	22389
90	23993	23959
95	25358	25424
100	26962	27611

Na podstawie pomiarów stworzyliśmy wykres, na podstawie którego można stwierdzić, że czujnik odległości działa prawidłowo, odczytany pomiar jest liniowy.



Rysunek 5: Wykres pomiarów dokonanych przez czujnik w zależności od rzeczywistej wysokości kulki. Niebieskie punkty – pomiar P1, pomarańczowe punkty – pomiar P2

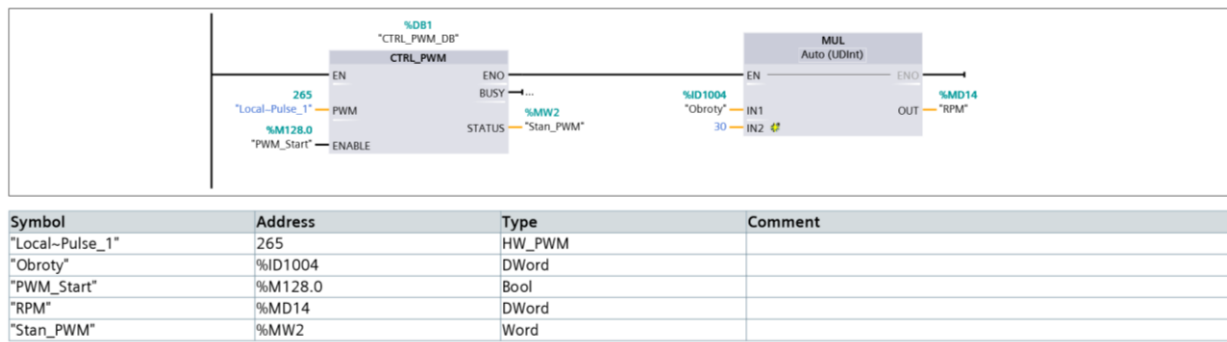
Wykorzystując regresję liniową w programie Excel dla uśrednionych wartości pomiarów czujnika laserowego obliczyliśmy współczynnik kierunkowy $a = 0,003342$ oraz wyraz wolny $b = 9,8301$, dzięki którym można powiązać wysokość kulki y z pomiarem czujnika x : $y = ax + b$.

4. Sterowanie prędkością obrotową wentylatora DC oraz pomiar jego prędkości obrotowej

Aby wykorzystać wentylator w układzie, skonfigurowaliśmy wyjście sterownika PTO1/PWM1, do którego jest podłączone sterowanie wentylatora. Zmieniliśmy ustawienia na tryb sygnału: PWM, podstawę czasu: milisekundy, format wartości pulsu: S7 analog format; okres cyklu 40ms.

W programie głównym OB1 dodałmy blok kontroli wyjścia PWM.

Network 3:



Rysunek 6. Blok kontroli wyjścia PWM oraz przelicznik częstotliwości na obroty na minutę.

Wejście „Enable” bloku „CTRL_PWM_DB” połączyliśmy ze zmienną wewnętrzną „PWM_Start”, której wartość jest zadawana przełącznikiem na panelu operatorskim.

Następnie przeprowadziliśmy konfigurację szybkiego licznika HSC, wprowadzając parametry:

- typ zliczania - częstotliwość,
- tryb pracy - jedna faza,
- kierunek zliczania - wybierany programowo,
- początkowy kierunek zliczania - do góry,
- okres zliczania - 1 sek.

Aby licznik mógł współpracować z sygnałem pochodzącym od wentylatora, zmieniliśmy domyślną filtrację sygnału na wejściu dyskretnym DI0.2 z wartości 6.4 ms na 1.6 ms.

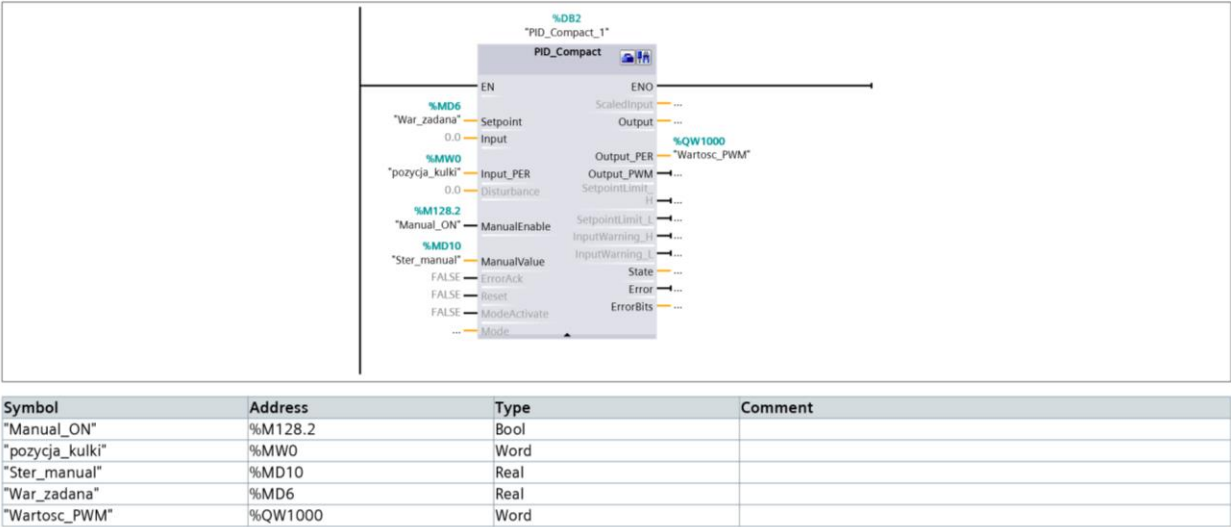
Na panelu operatorskim umieściliśmy pola cyfrowe do zadawania wartości sterowania PWM oraz odczytu prędkości obrotowej wentylatora. Przez blok MUL w programie głównym przeliczyliśmy prędkość obrotową na obroty na minutę (RPM).

Następnie przebadaliśmy zachowanie się układu sterowania. Wyznaczona przez nas wartość sterowania PWM, przy którym piłęczka zaczyna się podnosić do góry wynosiła około 18000, a sterowanie PWM, przy którym zaczynała spadać wynosiło około 17000. Ta różnica wartości spowodowana jest zachodzącym w układzie zjawiskiem histerezy – więcej siły należy przyłożyć do nieruchomej kulki, aby wzniesić ją w powietrze, niż potrzeba żeby ją w powietrzu utrzymać ze względu na zmiany ciśnień w rurze. Dodatkowo wyznaczyliśmy także minimalną wartość sterowania PWM, przy której zaczynał obracać się wentylator – 1400 oraz maksymalną prędkość obrotową wentylatora – 5130 obrotów/min.

5. Zastosowanie regulatora PID

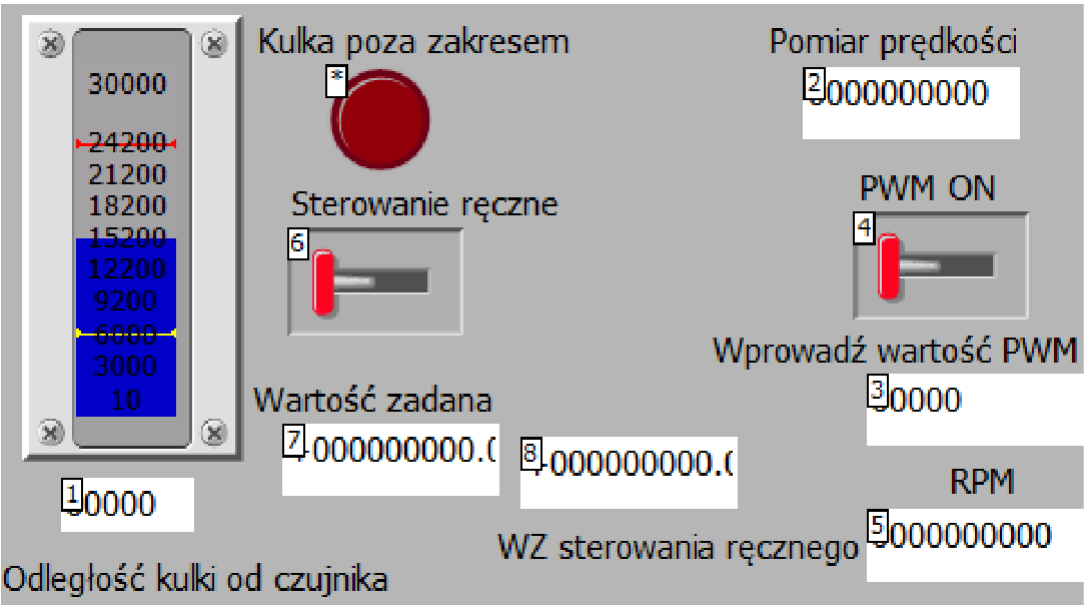
Na kolejnym etapie ćwiczenia umieściliśmy regulator PID z opcją sterowania ręcznego w bloku przerwán cyklicznych [OB30].

Network 1:



Rysunek 7. Podłączenie wejść i wyjść regulatora PID.

Wejścia i wyjścia regulatora zdefiniowaliśmy jako typu „PER”. Do wejścia ustawiającego wartość początkową podpięliśmy zmienną „War_zadana”. Na wejście podałyśmy zmienną „pozycja_kulki”, a na wyjście „Wartosc_PWM”. Opcję sterowania ręcznego zrealizowaliśmy, łącząc zmienną „Manual_ON” z wejściem odpowiadającym za tę opcję, a początkową wartość sterowania ręcznego wprowadzając za pomocą „Ster_manual”. Wszystkie te zmienne są ustawiane na panelu HMI.



Rysunek 8. Zrzut ekranu panelu operatorskiego.

Następnie ustawiliśmy wstępne nastawy regulatora PID zgodnie z poniższymi wartościami:

☒ Enable manual entry

Proportional gain: 6.101218E-1

Integral action time: 2.169376 s

Derivative action time: 0.549218 s

Derivative delay coefficient: 0.1

Proportional action weighting: 2.564467E-1

Derivative action weighting: 0.0

Sampling time of PID algorithm: 9.99985E-2 s

Tuning rule

Controller structure: PID

Rysunek 9. Wstępne nastawy regulatora PID

Zaobserwowaliśmy, że dla tych wartości układ jest w stanie przybliżyć wysokość piłeczki do wartości zadanej, ale z pewnym odchyleniem. Aby poprawić działanie układu postanowiliśmy przejść do automatycznego doboru nastaw.

6. Automatyczny dobór nastaw regulatora PID

Sterownik PLC, którego używaliśmy miał opcję uruchomienia procedury automatycznego doboru nastaw regulatora PID. Ponieważ nasza wartość zadana znajdowała się już blisko rzeczywistej wartości wybraliśmy opcję „Fine tuning” zamiast „Pretuning” dla wartości zadanej 50cm. Zaobserwowaliśmy w jaki sposób działa algorytm dobierający nastawy – początkowo pozycja piłeczki zmieniała się znacznie, potem wykonywała coraz mniejsze ruchy, aby układ mógł „nauczyć się” nią poruszać. Chociaż ten proces chwilę trwał, to z pewnością był szybszy niż ręczny dobór najlepszych wartości. Ostatecznie uzyskaliśmy wartości nastaw pokazane na poniższym rysunku:

☒ Enable manual entry

Proportional gain: 2.910965E-1

Integral action time: 2.812781 s

Derivative action time: 7.136555E-1 s

Derivative delay coefficient: 0.1

Proportional action weighting: 2.529754E-1

Derivative action weighting: 0.0

Sampling time of PID algorithm: 9.999911E-2 s

Tuning rule

Controller structure: PID

Rysunek 10. Automatyczne nastawy regulatora PID

Czasy całkowania i wzmocnienie lekko wzrosły, natomiast znacznie wzrósł czas różniczkowania w porównaniu z poprzednimi nastawami.

7. Wnioski

W wyniku powyższych działań udało nam się stworzyć układ, który był w stanie regulować samodzielnie wysokością piłeczki ping-pongowej tak, aby osiągnęła zadaną wartość. Nie było to idealne sterowanie, piłeczka nie osiągała dokładnej wysokości oraz ustalanie jej wysokości było dokładniejsze w okolicach 50cm. Było to spowodowane tym, że automatyczny dobór nastaw regulatora PID odbywał się właśnie dla 50cm. Dodatkowo niedokładność układu mogła być spowodowana zmiennym ciśnieniem w rurze oraz tym, że laser poprawnie mierzył odległość do spodu kulki jedynie w okolicach wysokości 50cm (w innych miejscach wiązka światła trafiała na boki piłeczki). Zmiana położenia piłeczki była także utrudniona dla regulatora ze względu na zachodzącą w układzie histerezę.

Problem automatycznego doboru nastaw regulatora PID dla konkretnej wysokości można byłoby rozwiązać poprzez stworzenie zestawu nastaw dla różnych zakresów wysokości i wybieranie właściwego na podstawie zadanej wartości.

Podczas laboratoriów nauczyliśmy się jak stosować regulator PID, sterować silnikiem za pomocą sygnału PWM, jak odczytywać sygnał z czujnika laserowego oraz przede wszystkim jak zaprogramować taki system przy użyciu sterownika PLC.