



# **POLITECHNIKA ŚLĄSKA**

## **WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI**

Przedmiot – Programowanie Systemów Sterowania

Temat: Model ARX w pętli z regulatorem PID

Autorzy: Szymon Badura, Przemysław Gröbner

Rok akademicki: 2022/2023,

Stopień: II,

Semestr: III,

Kierunek: Automatyka i robotyka,

Specjalizacje: Robotyka oraz SPil.

1. Link do repozytorium GitHub  
<https://github.com/Sbadura84/PSSLaboratorium.git>
2. Link do dokumentacji Doxygen  
<https://github.com/Sbadura84/PSSLaboratorium.git>
3. Lista wykorzystywanych klas:
  - Siso.h - klasa nadrzędna dla obiektów symulacji
  - ModelARX.h - klasa odpowiadająca za model ARX
  - Regulator.h - klasa odpowiadająca za regulator
  - Skok.h - klasa tworząca sygnał skoku o określonych parametrach
  - Iostream - klasa input output w celu wyświetlania na konsoli (debuggingu)
  - vector - klasa ułatwiająca przechowywanie danych i alokację pamięci
  - fstream - klasa odpowiadająca za obsługę plików
  - string - klasa odpowiadająca za elementy typu string
  - typeid - klasa służąca do debugowania kodu
  - Impuls.h - klasa generuje impuls jednostkowy
  - Generator.h - klasa łącząca sygnały i podaje odpowiednią wartość wyjściową
  - Sin.h - klasa generuje sygnał sinusoidalny o określonych parametrach
  - Trojkat.h - klasa tworząca sygnał trójkątny
  - Sygnał.h - klasa nadrzędna każdego sygnału, definiuje podstawowe parametry wszystkich sygnałów
4. Funkcje programu:
  - Program posiada klasy odpowiadające za generowanie sygnałów, wykorzystywanych w ramach zakłóceń lub jako wymuszenie. Sygnały dziedziczą z tej samej klasy - Sygnał.h - natomiast poszczególne sygnały z uwagi na różne parametry tworzone są w oddzielnych klasach. Program można łatwo rozszerzyć o dodatkowe typy sygnałów, poprzez dodanie kolejnych klas.
  - Modele ARX oraz regulatory dodawane są przez pliki tekstowe lub stworzenie stworzenie obiektów odpowiednich klas.

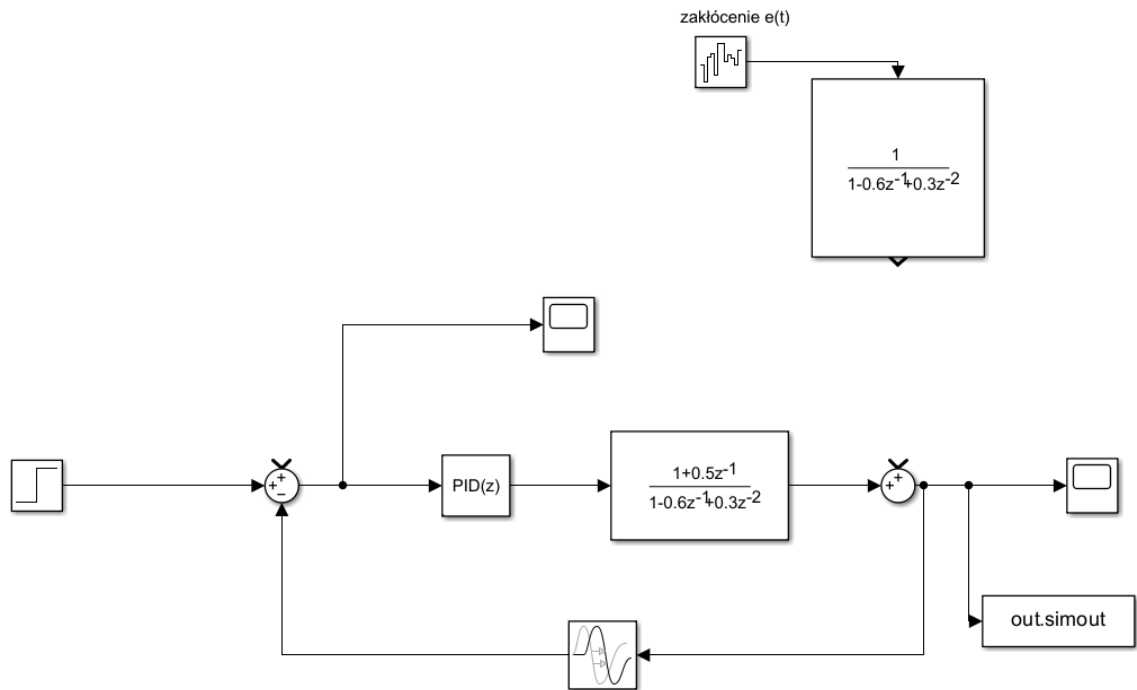
- Pętla symulacyjna jest wykona poprzez połączenie wyjść i wejść kolejnych elementów symulacji, natomiast wyniki zapisywane są w pliku tekstowym.

```
cout << "symuluj" << endl;

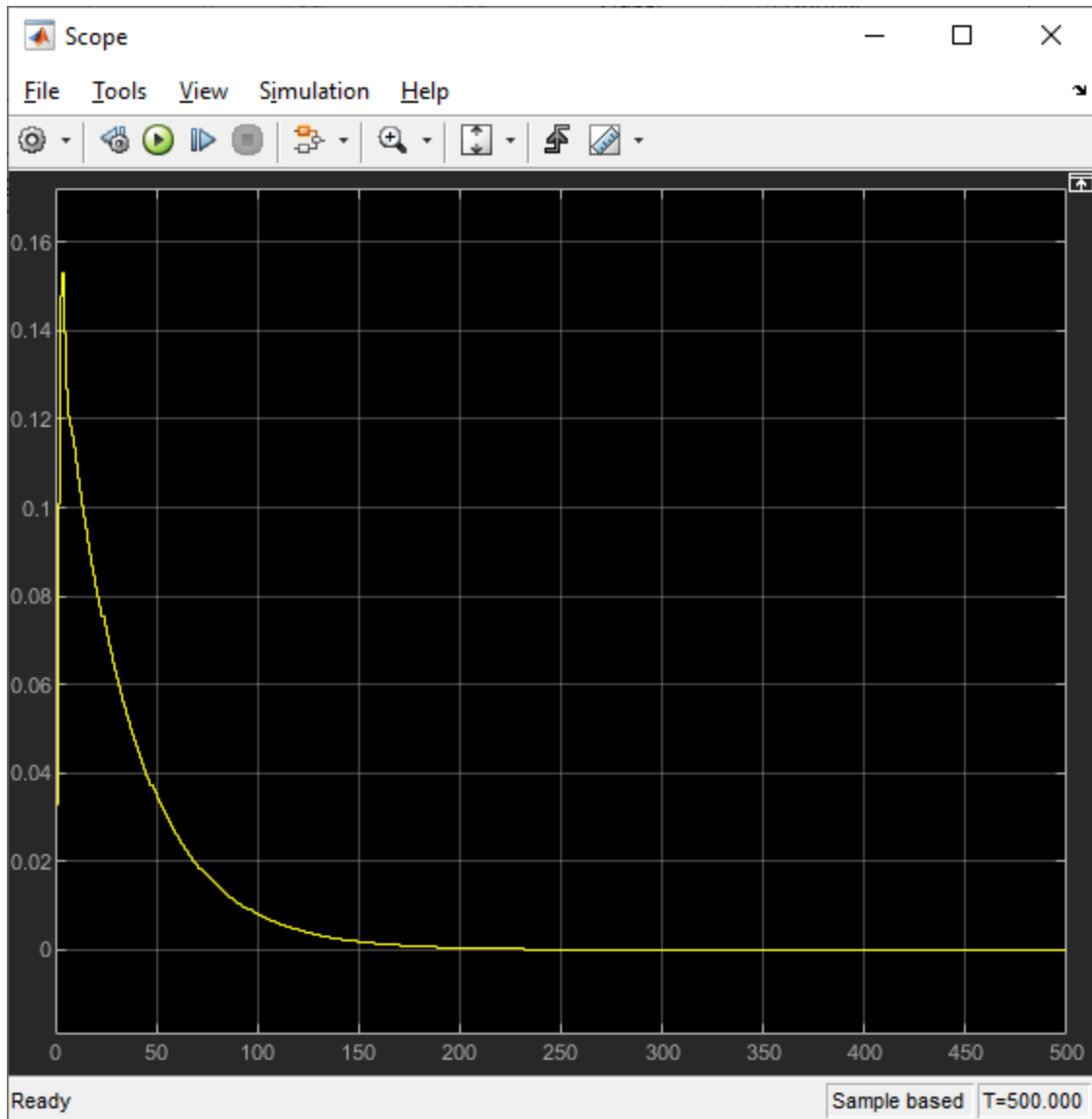
ofstream results;
results.open("resultsFile.txt");
for (int i = 0; i < 100; i++) {
    inputSignal = generator1.Generuj() - plantOutput;
    regulatorOutput = regTestowy.Symuluj(inputSignal, 0);
    plantOutput = modelTestowy.Symuluj(regulatorOutput, 0);
    results << plantOutput;
    results << "\n ";
}
results << "End of simulation.\n";
results.close();
```

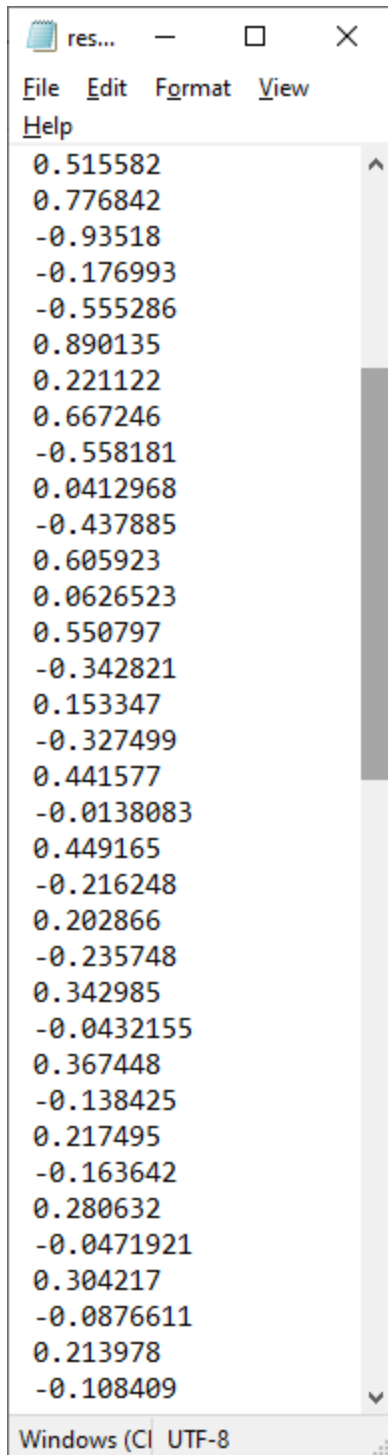
##### 5. Wyniki testów:

W celu testowania aplikacji, stworzyliśmy model w środowisku simulink



Następnie przeprowadziliśmy symulację obydwu układów. Otrzymane wyniki były zbliżone, tj. Nastawy regulatora były odpowiednie aby utrzymać obiekt w obydwu przypadkach w stanie stabilnym, natomiast wyjścia obydwu obiektów się różniły, co mogło być spowodowane różną jakością próbkowania.





```
res...
File Edit Format View
Help
0.515582
0.776842
-0.93518
-0.176993
-0.555286
0.890135
0.221122
0.667246
-0.558181
0.0412968
-0.437885
0.605923
0.0626523
0.550797
-0.342821
0.153347
-0.327499
0.441577
-0.0138083
0.449165
-0.216248
0.202866
-0.235748
0.342985
-0.0432155
0.367448
-0.138425
0.217495
-0.163642
0.280632
-0.0471921
0.304217
-0.0876611
0.213978
-0.108409
Windows (C| UTF-8
```

6. Wnioski i propozycje rozwoju

- Program wykonany w ramach laboratorium nie wymagał szaty graficznej oraz interfejsu użytkownika, lecz jest on przygotowany w taki sposób, że możliwa jest rozbudowa programu w tym kierunku.
- Pliki, z których odczytywane są dane, nie posiadają żadnych udogodnień dla użytkownika, tj. Nie posiadają instrukcji wprowadzania danych, przebudowanie

sekcji odpowiadającej za czytanie plików mogłaby usprawnić korzystanie dla osób zewnętrznych.

- Regulator PID został wykonany w prosty sposób, możliwym udoskonaleniem byłoby dodanie możliwości modyfikacji czasu całkowania.
- W celu ułatwienia komunikacji pomiędzy klasami, zastosowanie szablonu w klasie generator, mogłoby znacznie ułatwić dołączanie nowych typów sygnałów do generatora.
- Wylizanie rzędu wektorów A i B w modelach ARX mogłoby zostać zautomatyzowane w celu zmniejszenia szansy na pojawienie się błędu użytkownika.
- Dodanie możliwości modyfikacji nazwy pliku, do którego zapisywane są wyniki, takie rozwiązanie pozwoli na zapisywanie różnych wyników w różnych plikach, zamiast nadpisywać wyniki.