



Akademia Górniczo-Hutnicza  
im. Stanisława Staszica

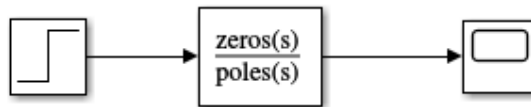
**Sprawozdanie**  
Sterowanie układów liniowych  
*Laboratorium 6*

**Borsuk Piotr**  
**Nr albumu 416947**  
**Technologie Przemysłu 4.0**  
**Grupa nr 1**  
**Rok akademicki 2023/2024**

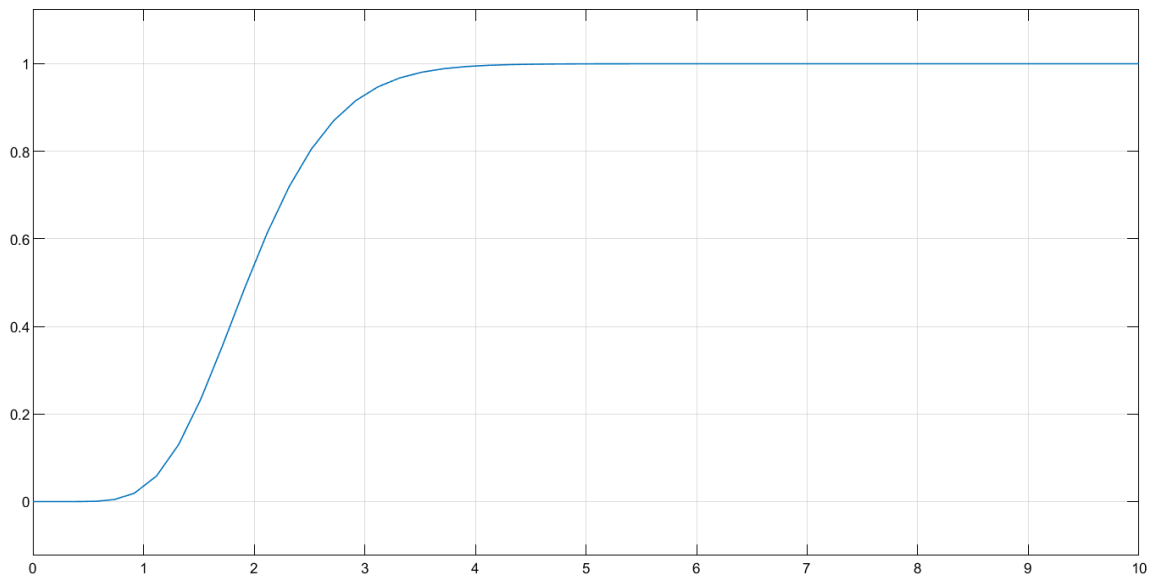
## Zadanie 1.1

Zrobiono transmitancje sygnału skokowego przy użyciu:

- Step c blok ten generuje skokowy sygnał wejściowy do modelowanego systemu. Ma następujące parametry:
  1. Step time: 0
  2. Initial value: 0
  3. Final value: 1
  4. Sample time: 0
- Zero-Pole – umożliwia modelowanie funkcji transferowej, która opisuje relację pomiędzy sygnałem wejściowym a wyjściowym w dziedzinie częstotliwości. Posiada poniższe cechy:
  1. Zeros: []
  2. Poles: [-5 -5 -5 -5 -5 -5 -5 -5 -5 -5]
  3. Gain:  $5^{10}$
  4. Absolute tolerance: auto
  5. Parameter tunability: auto
- Scope – jest używany do wizualizacji i monitorowania sygnałów w trakcie symulacji.



Schemat 1. Transmitancja sygnału.



Wykres 1. Odpowiedź scope

Możemy zaobserwować dwa punkty przejścia.

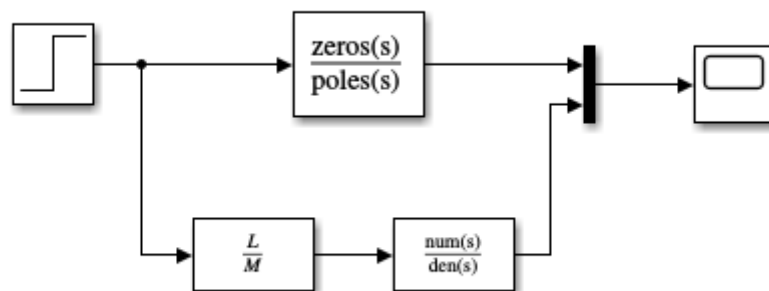
## Zadanie 1.2

Zaimplementowano kod mający na celu przygotowanie zmiennych oraz używa funkcji 'pade', która jest używana do przybliżania funkcji transmitancji w postaci rozwinięcia w szereg Taylora. Funkcja ta przyjmuje dwa argumenty: stałą czasową T oraz stopień rozwinięcia w szereg Taylora (w tym przypadku 5).

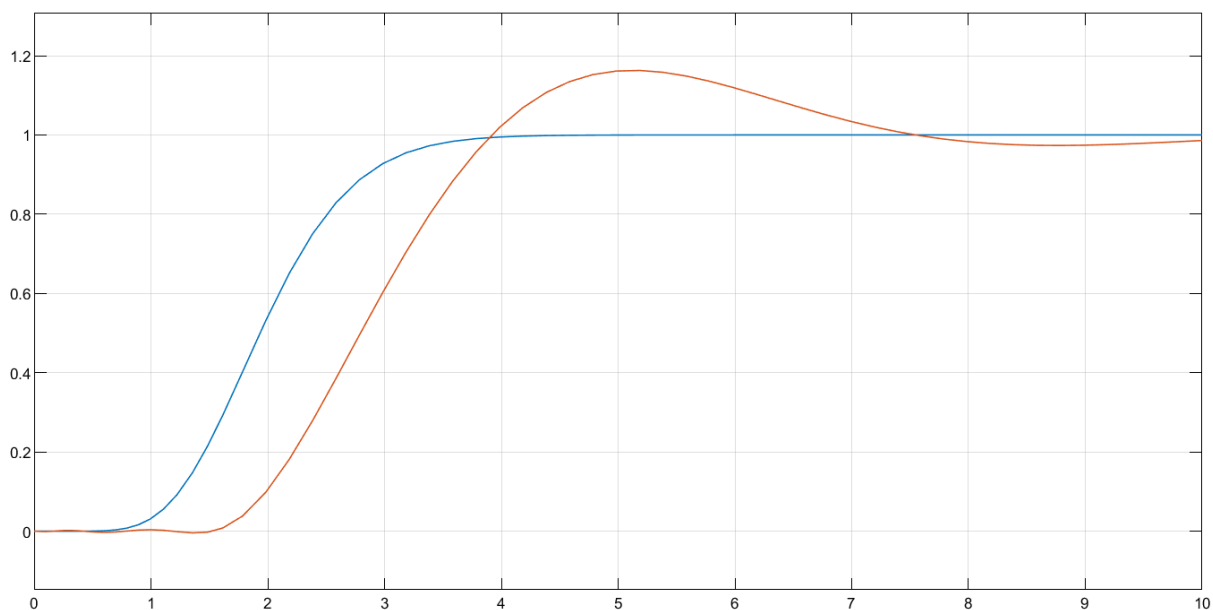
```
A = 1;  
B = 1;  
T = 1.5;  
[L, M] = pade(T, 5);
```

Dodaliśmy do poprzedniego modelu:

- Transfer Fcn – reprezentuje funkcję transmitancji, która opisuje relację między sygnałem wejściowym a wyjściowym w dziedzinie częstotliwości. O parametrach
  1. Numerator coefficients: [L]; [1]
  2. Denominator coefficients: [M]; [A B 1]
  3. Parametr tunability: auto
  4. Absolute tolerance: auto
- Mux – służy do łączenia wielu sygnałów wejściowych w jedno wyjście.



Schemat 2. Model transmitancyjny.

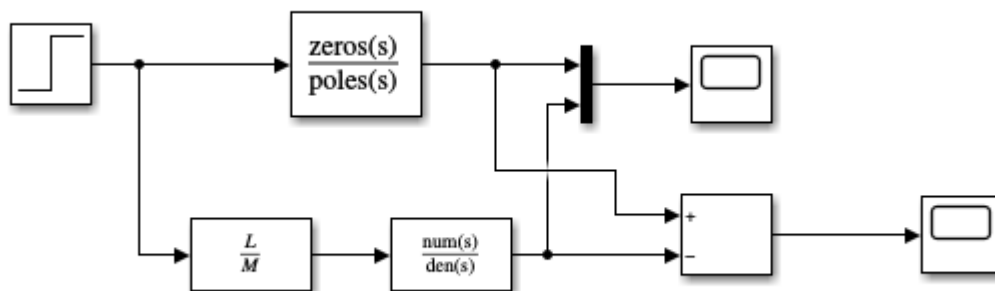


Wykres 2. Odpowiedź skokowa na bloku Scope

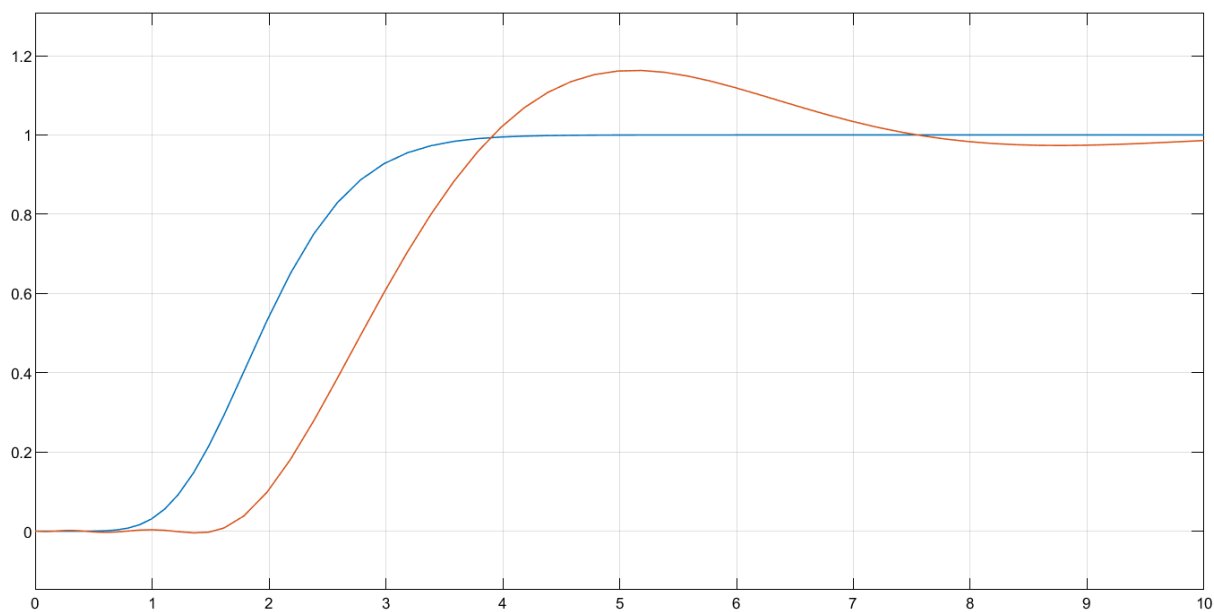
Transmitancja symuluje opóźnienie w sygnale około 1.5 sekundy

### Zadanie 1.3

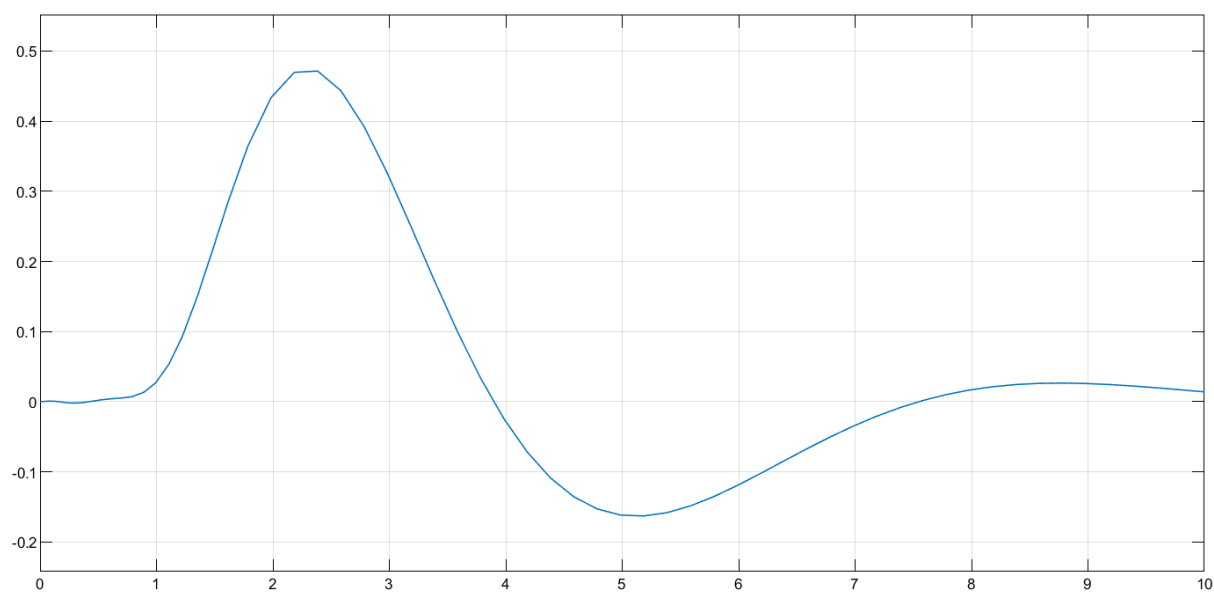
Do naszego modelu dodaliśmy blok „sum”, który używamy do sumowania sygnałów wejściowych i generowania sygnału wyjściowego, który jest sumą lub różnicą tych sygnałów, w zależności od konfiguracji bloku.



Schemat 3. Model transmitancyjny z dodanym blokiem „sum”.



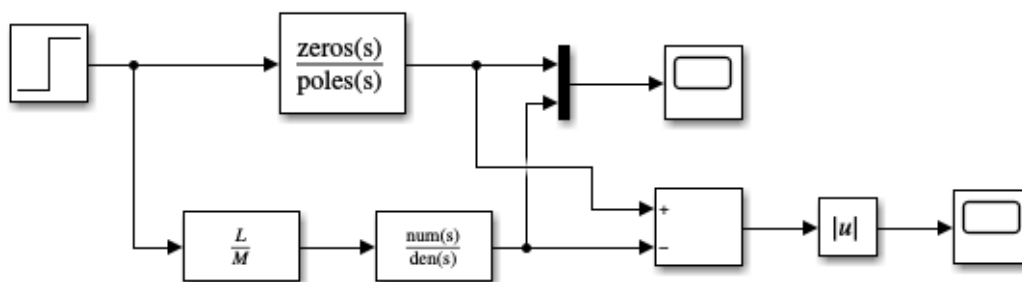
Wykres 3. Odpowiedź skokowa na scope 1.



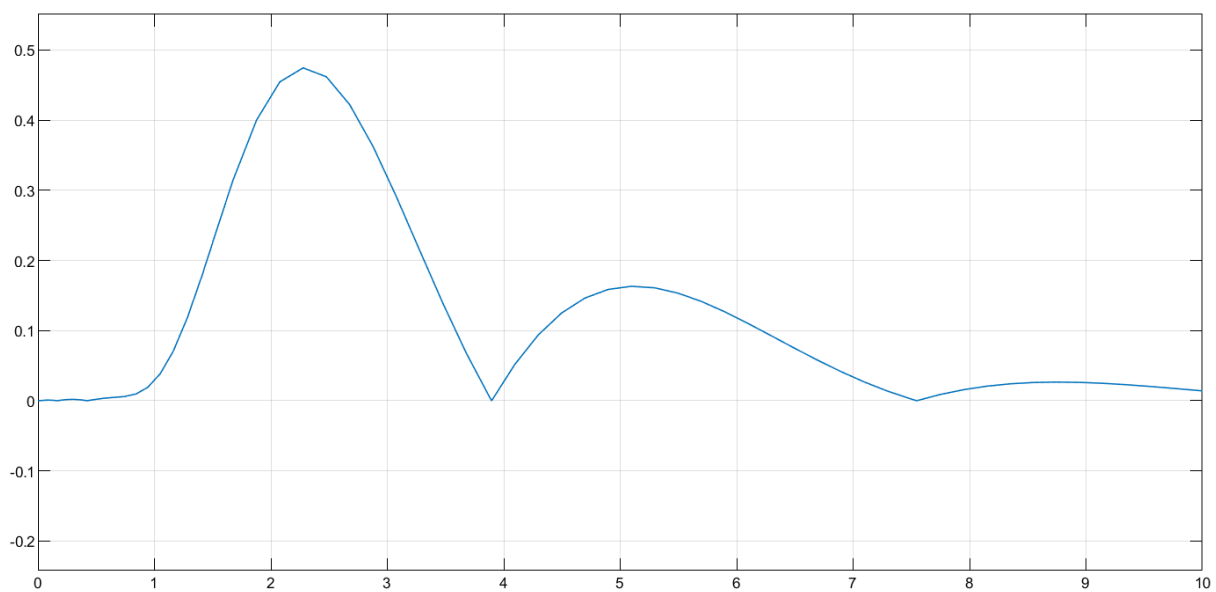
Wykres 4. Odpowiedź skokowa na scope 2.

#### Zadanie 1.4

Zaimplementowano blok „**abs**”, żeby zniwelować błąd wykresu 4. Chcemy, żeby cała oś y była nieujemna. Blok ten zwraca wartość bezwzględną sygnału.



Schemat 4. Model transmitancyjny z dodanym blokiem „abs”.

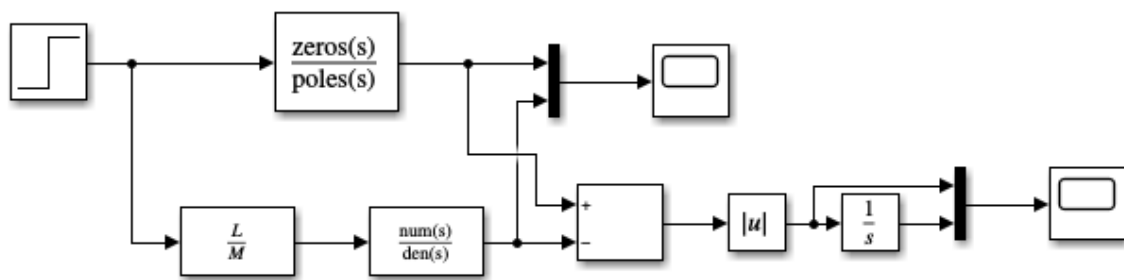


Wykres 5. Odpowiedź na scope 2.

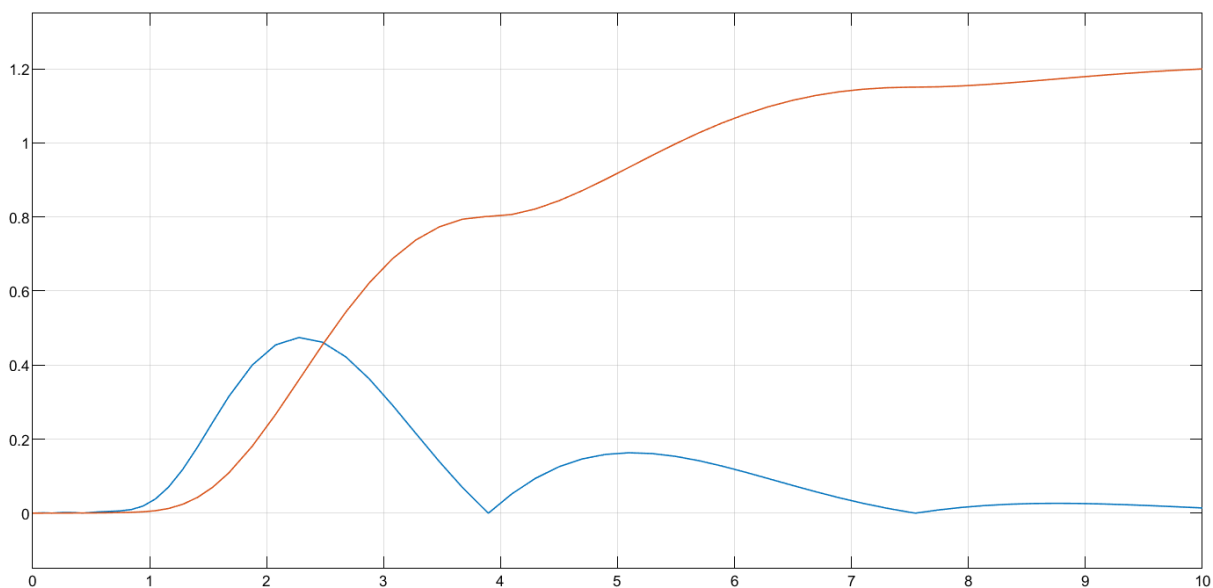
Wykres na scope 1 pozostał taki sam. W rezultacie dodania bloku „abs” odpowiedź scope 2 jest nieujemna.

### Zadanie 1.5

Dodajemy blok „integrator” w celu obliczenia pola pod wykresem oraz dodatkowy blok „Mux”.



Schemat 5. Model transmitancyjny z dodanym blokiem „integrator” oraz „mux”.

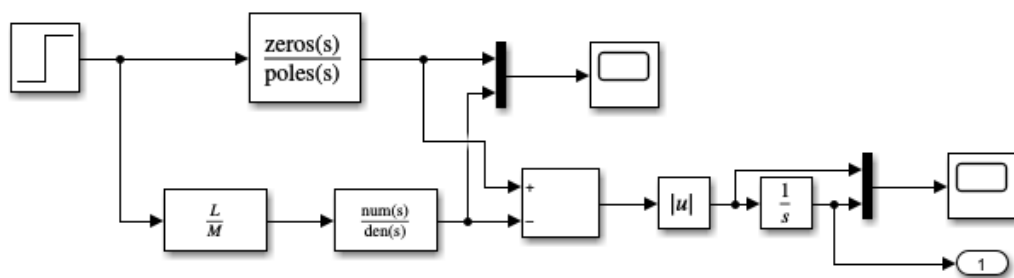


Wykres 6. Odpowiedź na scope 2

Pomarańczowa linia obrazuje nam wartości całkowania sygnału wejściowego ze względu na czas.

### Zadanie 1.6

Do naszego modelu włączamy blok „Out1”, który służy do przenoszenia sygnałów z modelu do środowiska zewnętrznego.



Schemat 6. Model transmitancyjny z dodanym blokiem „Out1”.

Blok ten nic nie zmienia w naszych wykresach, jedynie ma na celu przekazywania sygnałów na zewnątrz modelu do innych narzędzi MATLAB i do skryptów.

### Zadanie 1.7

Definiujemy dwa pliki celu i funkcji, które posłużą optymalizacji modelu dynamicznego w środowisku Simulink.

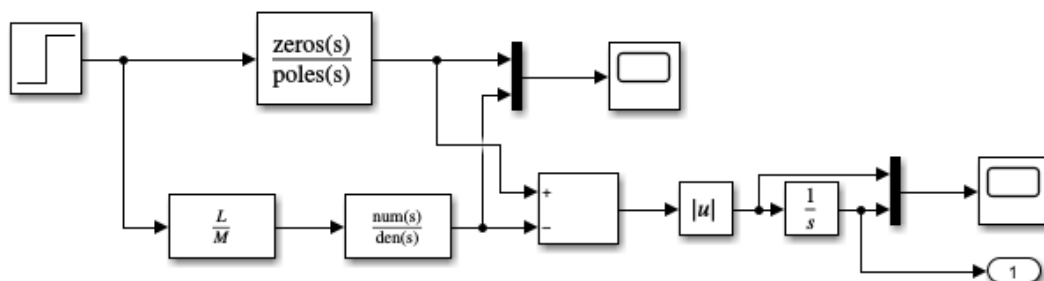
#### Plik „**cel.m**”

```
function y = cel(x)
global A B T L M
A = x(1);
B = x(2);
T = x(3);
[L, M] = pade(T,5);
sim('symulacja')
y = yout;
drawnow
```

#### Plik „**Skrypt.m**”

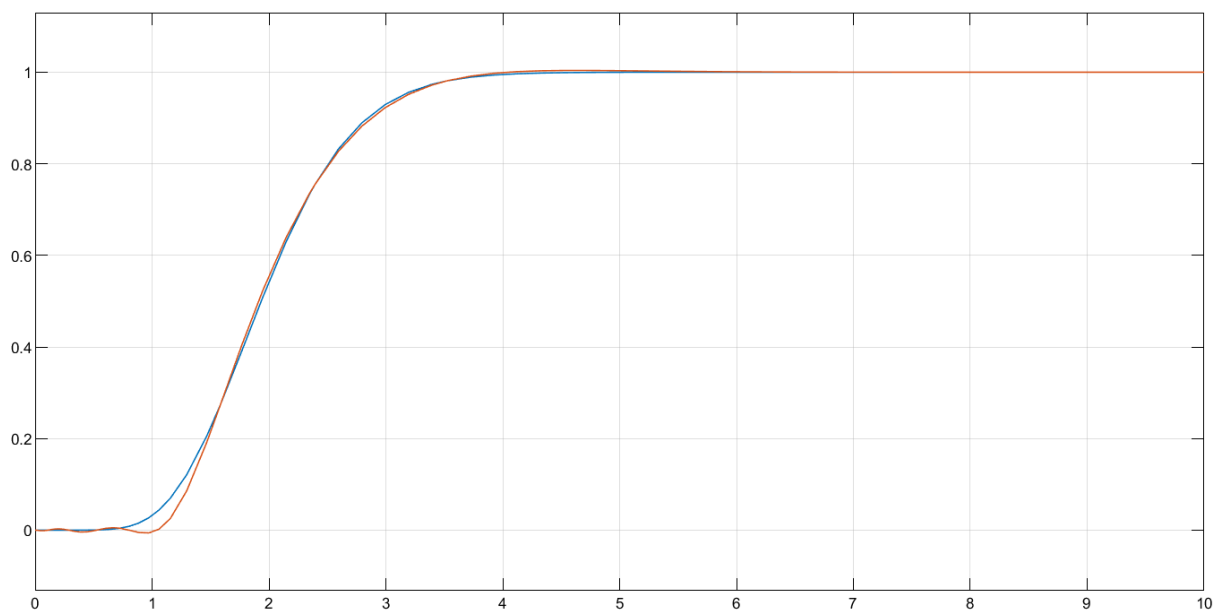
```
clc
clear
global A B T L M
x = fminsearch('cel',[1 1 1.5]);
A = x(1);
B = x(2);
T = x(3);
[L, M] = pade(T,5);
sim('symulacja.slx')
```

Skrypt próbuje znaleźć optymalne wartości parametrów A, B i T dla modelu w Simulinku, minimalizując wartość funkcji celu cel(x). Wartości te są optymalizowane tak, aby uzyskać pożądane wyniki symulacji.

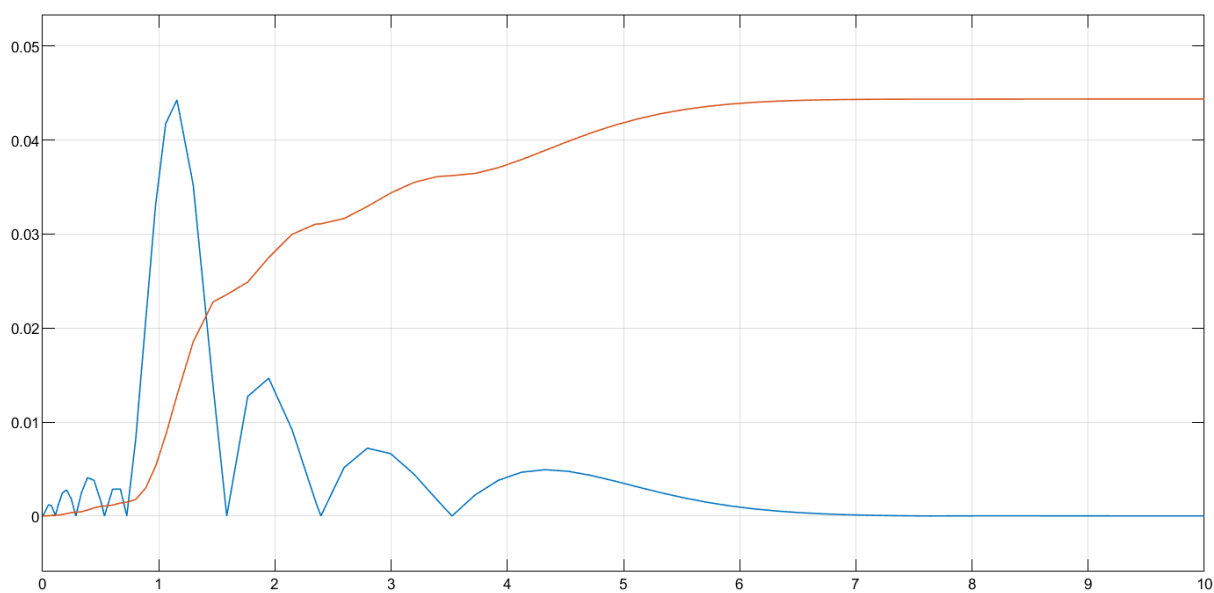


Schemat 7. Model transmitancyjny





Wykres 7. Odpowiedź na scope 1.



Wykres 8. Odpowiedź na scope 2.

Optymalizacja powiodła się. Na wykresie 7 możemy zauważyć, że linie praktycznie się pokrywają. Osiągnelismy najlepsze parametry dla:

- A: 0.3173
- B: 0.9813
- T: 1.0303

### Wnioski

Udało się osiągnąć zoptymalizowane parametry dla naszej funkcji. Kluczowym elementem udanej optymalizacji w Simulinku była precyzyjna definicja funkcji celu, zrozumienie modelu oraz właściwości funkcji matematycznych reprezentujących badany system. Dobre zrozumienie procesu optymalizacji i efektów wprowadzanych zmian pozwala na skuteczne dostosowanie modelu do założonych celów.