





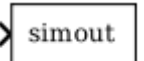




Laboratorium Metod Obliczeniowych
Wydział Elektrotechniki Automatyki i Informatyki
Politechnika Świętokrzyska

Studia: Stacjonarne I stopnia	Kierunek: Informatyka
Data wykonania: 08.01.2018	Grupa: 3ID13B
Imię i nazwisko: Bartłomiej Osak	
Numer ćwiczenia: 11	Temat ćwiczenia: Równania różniczkowe - Simulink

1. Wstęp teoretyczny.

Simulink jest rozszerzeniem pakietu MATLAB – przy pomocy graficznego środowiska możemy konstruować diagramy czasowe, które reprezentują określone procesy dynamiczne. W języku symulacyjnym Simulink wykorzystuje się najczęściej pliki Matlaba typu .m, służące do rozwiązywania układów równań różniczkowych, liniowych oraz nieliniowych. Zamiast pisać kod Matlaba opisujemy procesy w postaci połączonych bloków, wykorzystując odpowiednie komponenty, reprezentujące dane wejściowe, części układu oraz dane wyjściowe.

Stosowane komponenty w wykonanych układach:

 Constant	Wartość stała
 Integrator	Integrator
 Scope	Obserwacja sygnału (rezultat)
 Sine Wave	Funkcja sinusoidalna (sinus)
 To Workspace	Zapisanie wyniku do zmiennej (wykorzystanie w Matlab)
 Gain	Bramka
 Product	Produkt - zmienna
 Mux	Multiplekser
 Sum	Sumator

Równaniem różniczkowym nazywamy równanie, w którym występuje związek funkcji niewiadomej i jej pochodnych. Rząd równania różniczkowego jest równy największemu rzędowi występujących w nim pochodnych. Równaniem różniczkowym zwyczajnym rzędu n nazywamy równanie postaci:

$$F(x, y, y', y'', \dots, y^{(n)}) = 0$$

w którym niewiadomą jest funkcja:

$$y = y(x)$$

i w którym występuje pochodna rzędu n tej funkcji wraz z pochodnymi niższych rzędów, tzn:

$$y' = \frac{dy}{dx}, y'' = \frac{d^2y}{dx^2}, y''' = \frac{d^3y}{dx^3}, \dots, y^{(n)} = \frac{d^{(n)}y}{dx^{(n)}}$$

Rozwiązaniem lub całką równania różniczkowego w przedziale $[a, b]$ nazywamy każdą funkcję zmiennej x wyrażoną w postaci jawnej:

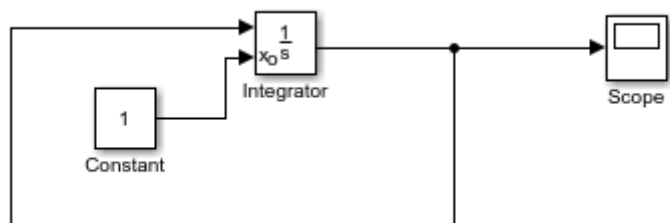
$$y = y(x)$$

lub w postaci uwikłanej:

$$h(x, y) = 0$$

2. Integrator z wartością inicjującą.

Model symulacyjny:



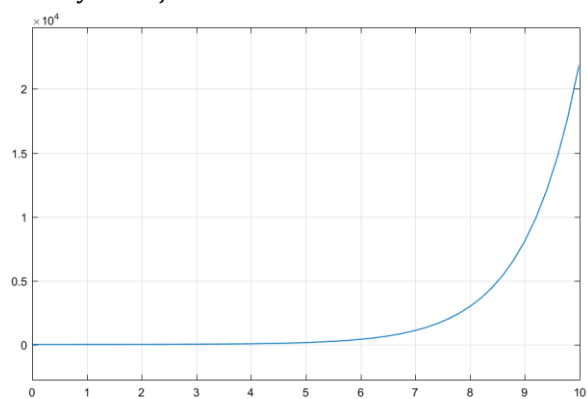
Konfiguracja:

- Constant → wartość inicjująca: 1
- Integrator → wartość inicjująca: z zewnątrz z bloku Constant

Rezultat:

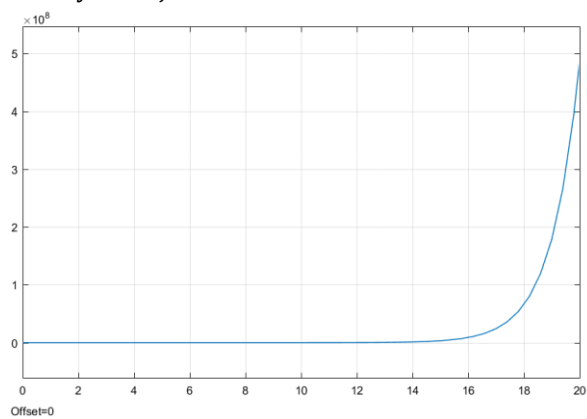
Integrator → wartość inicjująca: 1 (constant)

Czas symulacji: 10.0

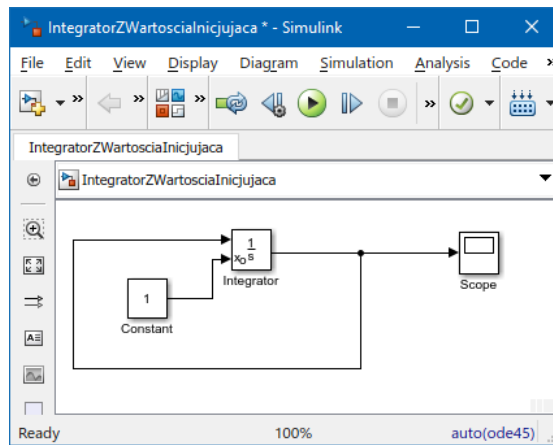
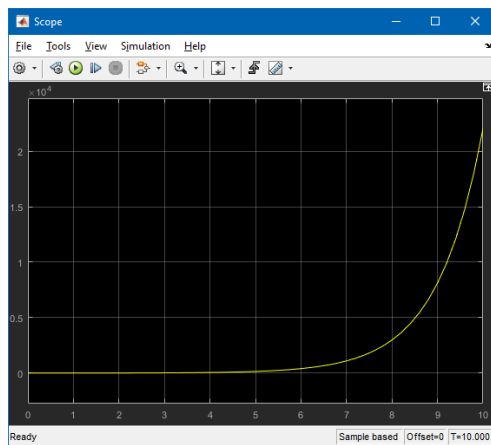


Integrator → wartość inicjująca: 1 (constant)

Czas symulacji: 20.0

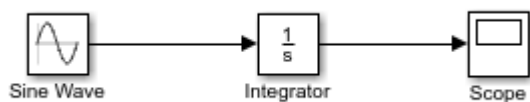


Przykładowe zrzuty ekranu:



3. Sinusoida.

Model symulacyjny:



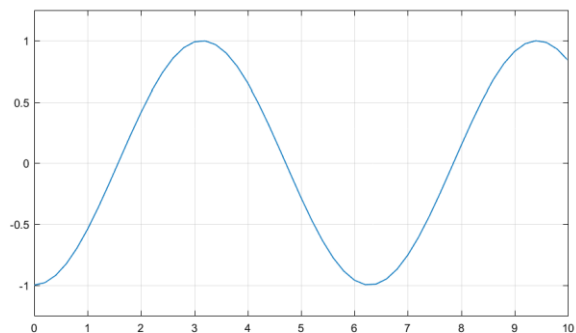
Konfiguracja:

- Integrator → wartość inicjująca wewnętrzna: -1

Rezultat:

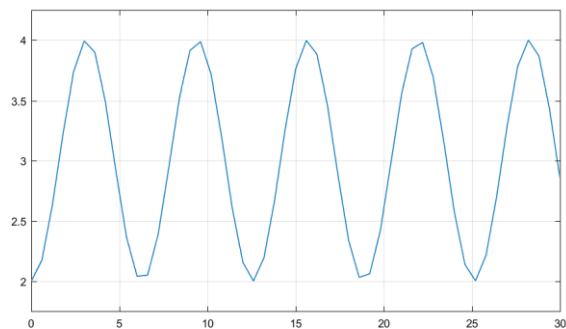
Integrator → wartość inicjująca wewnętrzna: -1

Czas symulacji: 10.0

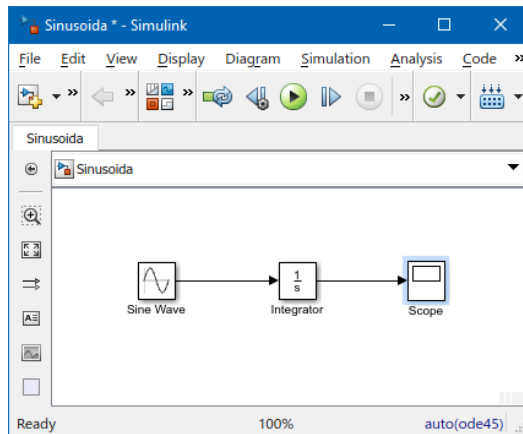
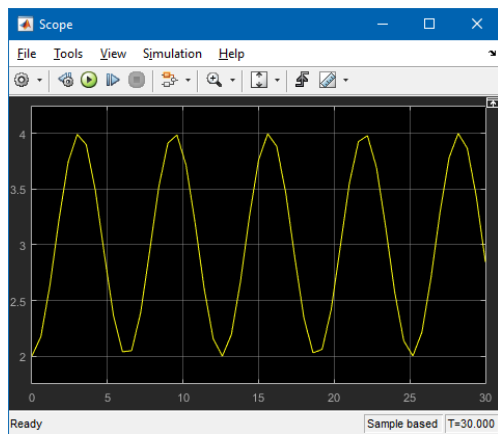


Integrator → wartość inicjująca wewnętrzna: 2

Czas symulacji: 30.0



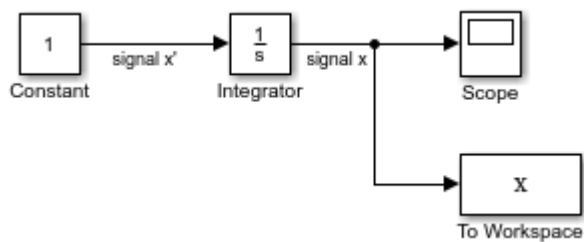
Przykładowe zrzuty ekranu:



4. Proste równanie różniczkowe.

Polecenie: Mamy równanie różniczkowe postaci: $\dot{x} = 1$. Wiedząc, że rozwiązanie takiego równania to prosta $x = t + C$ (C – stała) stworzyć model symulacyjny.

Model symulacyjny:



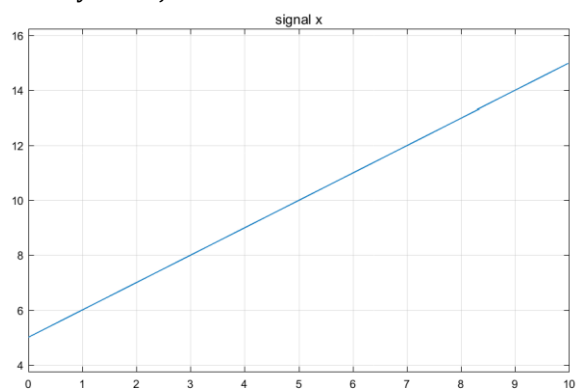
Konfiguracja:

- Integrator → wartość inicjująca wewnętrzna: 5

Rezultat – w Simulink:

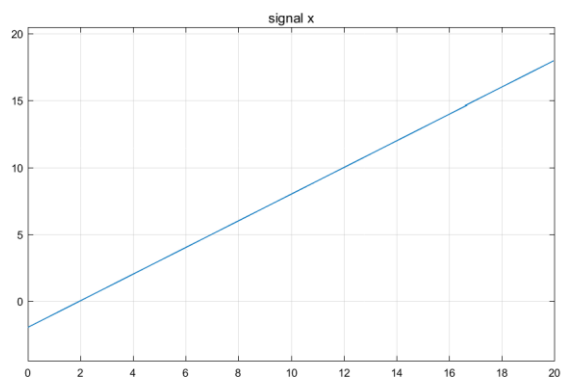
Integrator → wartość inicjująca wewnętrzna: 5

Czas symulacji: 10.0



Integrator → wartość inicjująca wewnętrzna: -2

Czas symulacji: 20.0



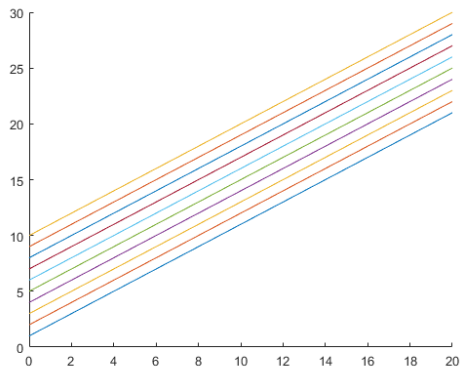
Rezultat – w Matlab:

Warunkiem początkowym dla integratora jest zapis „war_pocz”, który jest wykorzystywany w skrypcie matlaba.

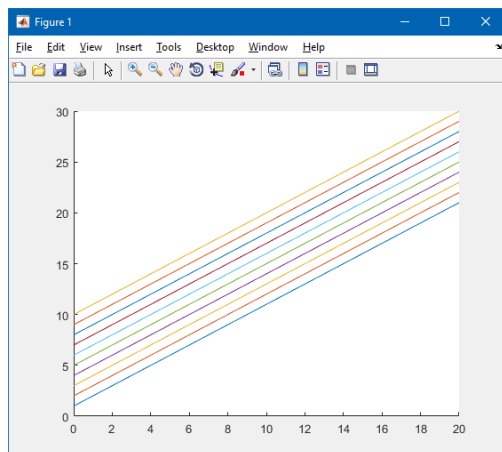
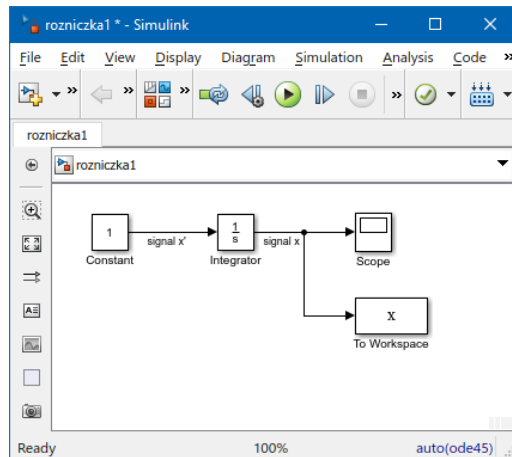
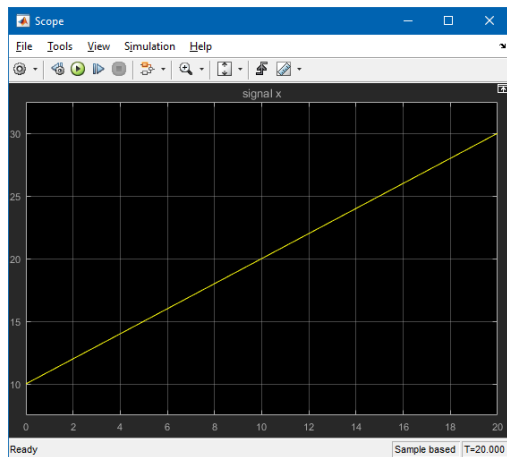
Kod skryptu:

```
figure;  
hold on;  
for i=1:10  
    war_pocz = i;  
    sim('rozniczka1');  
    plot(tout,x);  
end
```

Rezultat:



Przykładowe zrzuty ekranu:



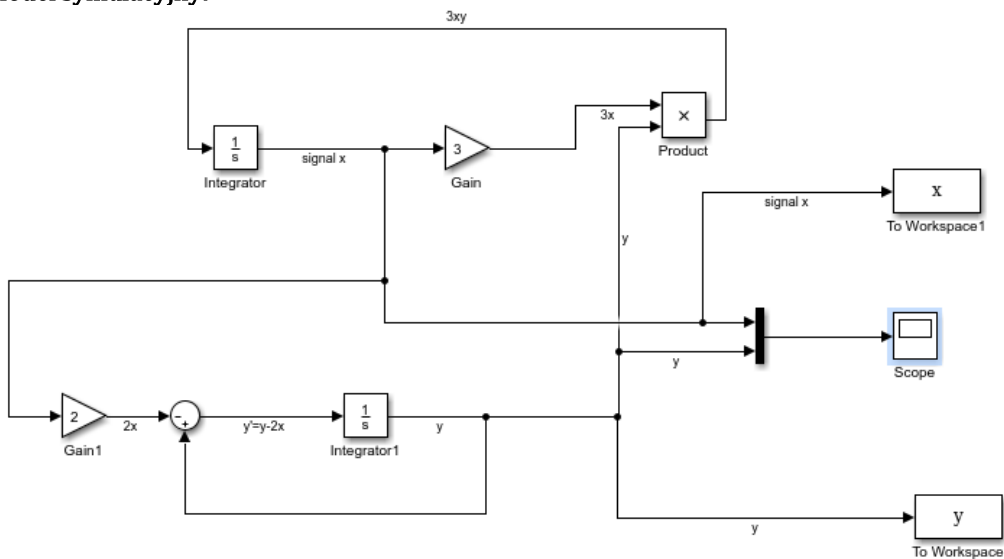
5. Układ prostych równań różniczkowych.

Polecenie: Dany jest układ dwóch równań różniczkowych:

$$\begin{cases} \dot{x} = 3xy \\ \dot{y} = y - 2x \end{cases}$$

Wykonać odpowiedni model symulacyjny.

Model symulacyjny:



Konfiguracja:

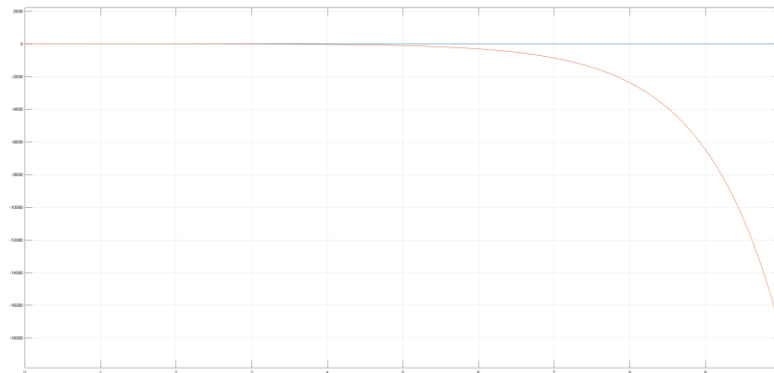
- Integrator → wartość inicjująca wewnętrzna: 1
- Integrator1 → wartość inicjująca wewnętrzna: 1

Rezultat – w Simulink:

Integrator → wartość inicjująca wewnętrzna: 1

Integrator1 → wartość inicjująca wewnętrzna: 1

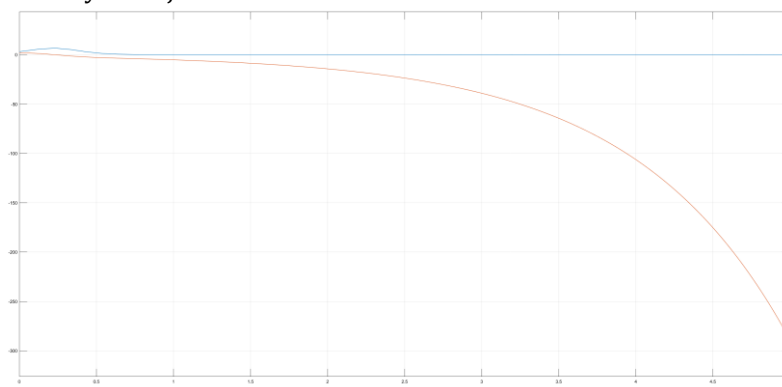
Czas symulacji: 10.0



Integrator → wartość inicjująca wewnętrzna: 3

Integrator1 → wartość inicjująca wewnętrzna: 2

Czas symulacji: 5.0



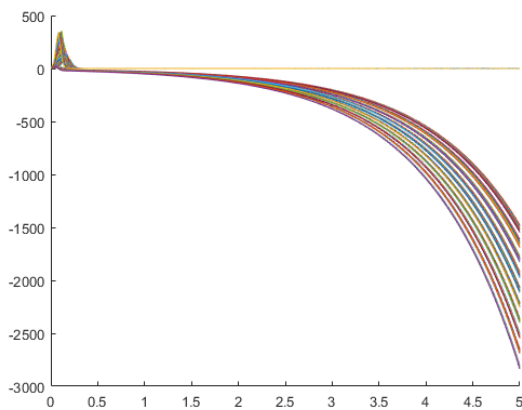
Rezultat – w Matlab:

Warunkiem początkowym dla integratora jest zapis „war_pocz”, a dla integratora1 „war_pocz1”.

Kod skryptu:

```
figure;  
hold on;  
for i=1:10  
    for j=11:20  
        war_pocz = i;  
        war_pocz1 = j;  
        sim('rozniczka2');  
        plot(tout,x,tout,y);  
    end  
end
```

Rezultat:



Przykładowe zrzuty ekranu:

