Akademia Górniczo-Hutnicza

im. Stanisława Staszica

Sprawozdanie

Sterowanie układów liniowych

Laboratorium 1

Borsuk Piotr

Nr albumu 416947

Technologie Przemysłu 4.0

Grupa nr 1

Rok akademicki 2023/2024

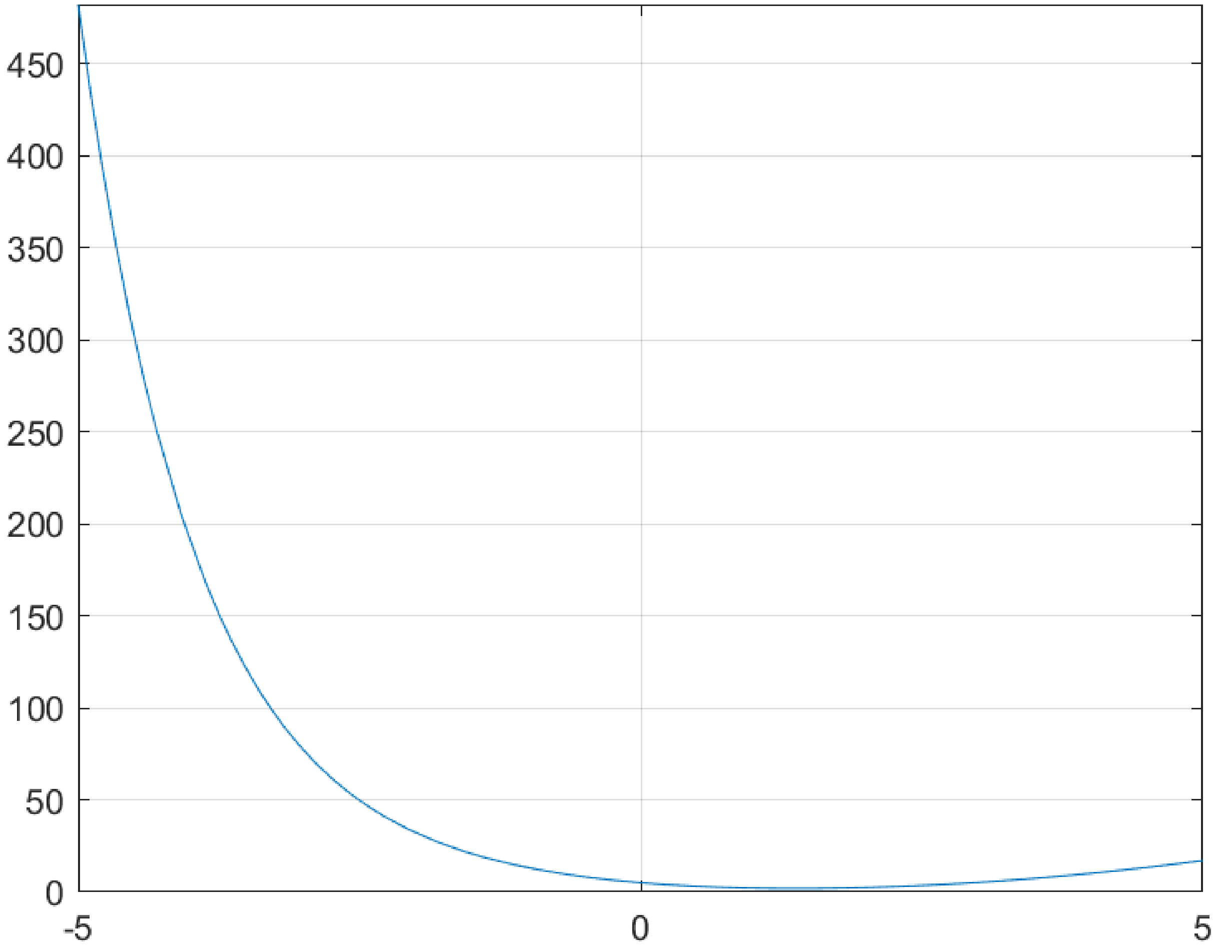
Zadanie 1

Poniższy kod rozwiązuje równanie różniczkowe liniowe pierwszego rzędu: $\frac{dy}{dt} + y = t^2$. Rysuje jego wykres co pozwala zobaczyć zachowanie rozwiązania tej funkcji.

```
clc %czyści okno konsoli
clear % czyści wszystkie zmienne z bieżącej przestrzeni roboczej
syms y(t) %deklaruje zmienną symboliczną y(t)
Dy = diff(y); % oblicza pochodną ‘y’ względem ‘t’ i przypisuje ją do zmiennej ‘Dy’
z = dsolve(Dy+y==t^2,y(0) == 5); %rozwiązuje równanie róż. z warunkiem początkowym
pretty(z) % wyświetla rozwiązanie ‘z’ w formie bardziej czytelnej
fplot(z) %generuje wykres rozwiązania ‘z’
grid on %włącza siatkę na wykresie
```

Wynik:

```
              2
3 exp(-t) - 2 t + t  + 2
```



Rys. 1. Wykres rozwiązania funkcji różniczkowej $\frac{dy}{dt} + y = t^2$

Następnie przekształcono powyższy kod, żeby rozwiązywał równanie różniczkowe liniowe pierwszego rzędu: $\frac{dy}{dt} + y = t^2$ z warunkami początkowymi: **y0 = [-5 -2 0 5]** i rysował wykresy tych rozwiązań na jednym wykresie.

Dodaliśmy w nim:

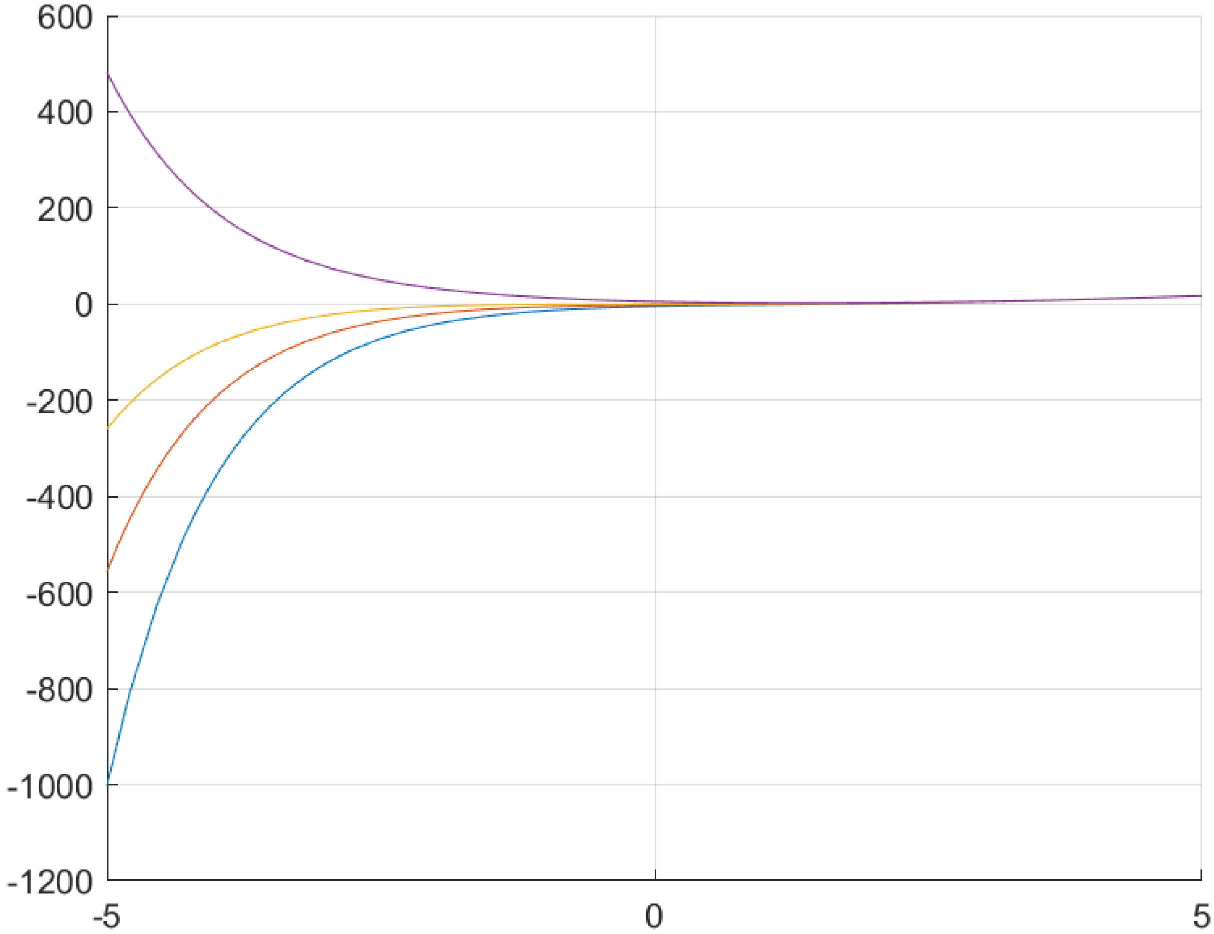
- ‘syms y(t) **a**’ – Zadeklarowano symboliczną zmienną **‘a’**, która będzie użyta jako stała w warunku początkowym
- ‘z = dsolve(Dy+y == t^2, **y(0) == a)**;’ – dodaliśmy warunek początkowy, gdzie **‘a’** jest zmienną
- ‘**y0 = [-5 -2 0 5];**’ - Tworzy wektor **‘y0’**, który zawiera różne wartości **‘a’** (warunków początkowych), dla których chcemy obliczyć rozwiązania.
- Pętla **‘for’** - Przechodzi przez każdą wartość **‘a’** z wektora **‘y0’** i oblicza odpowiadające rozwiązanie **‘z1’** poprzez podstawienie konkretnej wartości **‘a’** w miejsce **‘a’** w równaniu **‘z’**. Następnie rysuje wykres tego rozwiązania na aktualnym wykresie.

Ostatecznie, kod ten generuje jeden wykres, na którym rysuje rozwiązania równania różniczkowego dla różnych warunków początkowych zdefiniowanych w wektorze **‘y0’**. Każdy wykres jest rysowany na jednym wykresie, dzięki czemu można porównać różne rozwiązania.

```
clc
clear
syms y(t) a
Dy = diff(y);
z = dsolve(Dy+y == t^2, y(0) == a);
pretty(z)
y0 = [-5 -2 0 5];
figure %tworzy nowe okna rysunku, w którym będą generowane wykresy dla ‘a’
hold on
for i = 1:length(y0)
    z1 = subs(z, 'a', y0(i));
    fplot(z1)
end
grid on
```

Wynik:

```
           2
exp(-t) (a - 2) - 2 t + t  + 2
```



Rys. 3. Wykres rozwiązania dla funkcji $\frac{dy}{dt} + y = t^2$ z warunkami początkowymi

Równanie różniczkowe liniowe drugiego rzędu

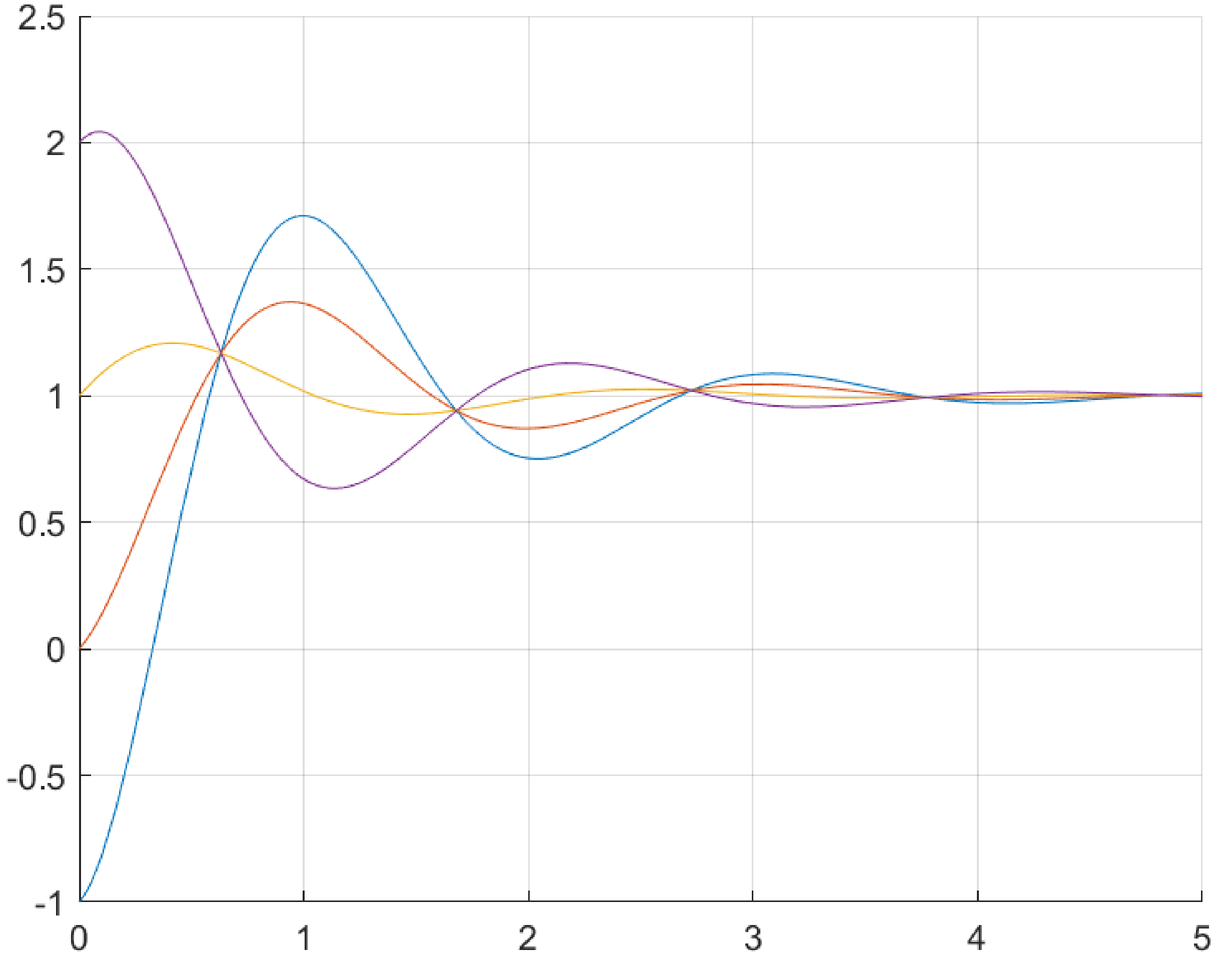
Kod wykonuje równanie różniczkowe liniowe drugiego rzędu: $\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 10y = 10$ z różnymi warunkami początkowymi oraz rysuje wykresy tych rozwiązań.

```
clc
clear
syms y(t) a
Dy = diff(y);
D2y = diff(y,2);
z = dsolve(D2y + 2*Dy + 10*y == 10, Dy(0) == 1, y(0) ==a);
pretty(z)
y0 = [-1 0 1 2];
figure
hold on
for i = 1:length(y0)
    z1 = subs(z, 'a', y0(i))
    fplot(z1,[0,5])
end
grid on
```

Wynik:

```
Command Window

a sin(3 t) exp(-t)
------------------ + cos(3 t) exp(-t) (a - 1) + 1
        3


z1 =

1 - (sin(3*t)*exp(-t))/3 - 2*cos(3*t)*exp(-t)


z1 =

1 - cos(3*t)*exp(-t)


z1 =

(sin(3*t)*exp(-t))/3 + 1


z1 =

cos(3*t)*exp(-t) + (2*sin(3*t)*exp(-t))/3 + 1

fx >>
```

Rys. 4. Wynik rozwiązania $\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 10y = 10$ z warunkami początkowymi



Rys. 5. Wykres rozwiązania $\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 10y = 10$ z warunkami początkowymi

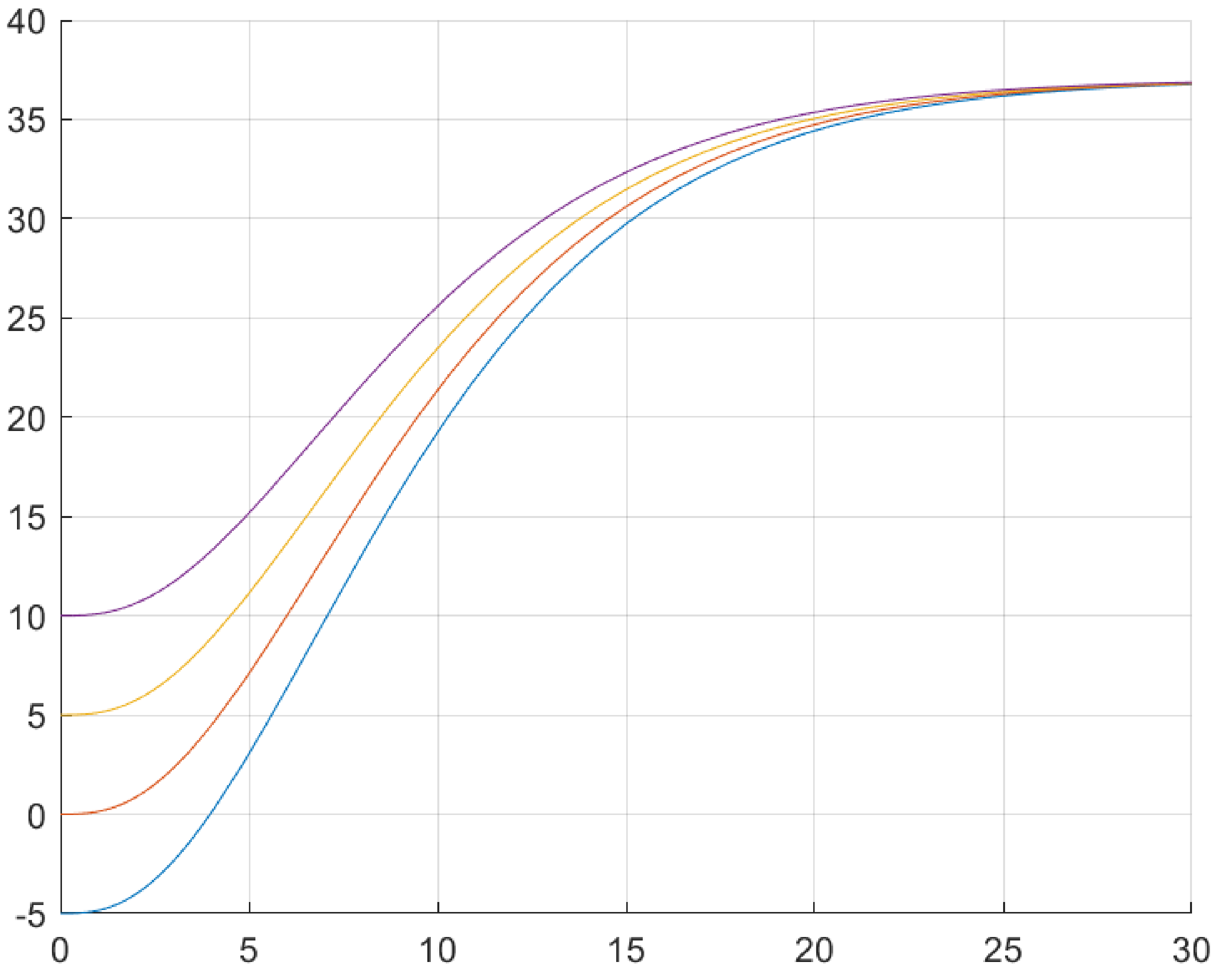
Równanie różniczkowe liniowe trzeciego rzędu

Oto przykładowe rozwiązanie równania różniczkowego liniowego trzeciego rzędu dla równania: $\frac{d^3y}{dt^3} + 0{,}9\frac{d^2y}{dt^2} + 0{,}27\frac{dy}{dt} + 0{,}027y = 1$. Z warunkami początkowymi **y0 = [-5, 0, 5, 10]** i wykresem z zakresem osi – [0 30].

```
clc
clear
syms y(t) a
Dy = diff(y);
D2y = diff(y, 2);
D3y = diff(y, 3);

eqn = D3y + 0.9 * D2y + 0.27 * Dy + 0.027 * y == 1;

y0 = [-5, 0, 5, 10];

figure
hold on
for i = 1:length(y0)
    z = dsolve(eqn, Dy(0) == 0, D2y(0) == 0, y(0) == y0(i));
    fplot(z, [0, 30])
end
pretty(z)
grid on
```

```
Command Window
      /    3 t \ /             / 3 t \                2          \
   exp| - --- | | 4380 t - exp| --- | 20000 + 657 t  + 14600 |
      \    10 / \             \  10 /                          /
 - -------------------------------------------------------------
                              540

fx >>
```

Rys. 6. Wynik rozwiązania równania $\frac{d^3y}{dt^3} + 0{,}9\frac{d^2y}{dt^2} + 0{,}27\frac{dy}{dt} + 0{,}027y = 1$ z warunkami początkowymi.



Rys. 7. Wykres rozwiązania równania $\frac{d^3y}{dt^3} + 0{,}9\frac{d^2y}{dt^2} + 0{,}27\frac{dy}{dt} + 0{,}027y = 1$ z warunkami początkowymi

Przekształcenia Laplace'a

Sprawdzono, jak wyglądają transformaty Laplace'a i wyświetlano je w czytelny sposób następujących wyrażeń:

$$y = t,$$

$$y = t^n,$$

$$y = k * t * e^{-\omega t},$$

$$y = k * \sin(\omega * t),$$

$$y = k * \cos(\omega * t),$$

$$y = k * \sin(\omega * t) * e^{-\omega t},$$

$$y = k * \cos(\omega * t) * e^{-\omega t}.$$

```
clc
clear
syms k a t n w
y = k*exp(-a*t);
y1 = t;
```

```
y2 = k*t^n;
y3 = k*t^n*exp(-a*t);
y4 = k*sin(w*t);
y5 = k*cos(w*t);
y6 = k*sin(w*t)*exp(-a*t);
y7 = k*cos(w*t)*exp(-a*t);
Y = laplace(y);
Y1 = laplace(y1);
Y2 = laplace(y2);
Y3 = laplace(y3);
Y4 = laplace(y4);
Y5 = laplace(y5);
Y6 = laplace(y6);
Y7 = laplace(y7);
pretty(Y)
pretty(Y1)
pretty(Y2)
pretty(Y3)
pretty(Y4)
pretty(Y5)
pretty(Y6)
pretty(Y7)
```

Wynik:

Command Window

```
  k
-----
a + s

 1
--
 2
s

{ k gamma(n + 1)
{ --------------  if  -1 < real(n) or (n in integer and 1 <= n)
{      n + 1
{     s

{ k gamma(n + 1)
{ --------------  if  -1 < real(n) or (n in integer and 1 <= n)
{         n + 1
{  (a + s)

  k w
-------
 2    2
s  + w
```

```
   k s
-------
 2    2
s  + w

      k w
-------------
        2    2
(a + s)  + w

   k (a + s)
-------------
        2    2
(a + s)  + w

fx >>
```

Rys. 7,8. Czytelne rozwiązania transformat Laplace'a

Podane powyżej wyniki przedstawiają transformaty Laplace'a dla tych funkcji. Wartości **'k'**, **'a'**, **'n'**, i **'w'** to parametry, które można dostosować w zależności od konkretnych wartości funkcji.

Podsumowanie

MATLAB oferuje narzędzia do rozwiązywania równań różniczkowych, obliczania transformat Laplace'a i analizy funkcji w dziedzinie czasu i częstotliwości. Transformaty Laplace'a są przydatne w analizie układów dynamicznych i rozwiązywaniu równań różniczkowych. Możemy dostosowywać zakresy czasu, warunki początkowe i parametry w analizie funkcji w MATLABie, co pozwala na szeroki zakres zastosowań w inżynierii i matematyce.