Міністерство освіти і науки України

Дніпровський національний університет імені Олеся Гончара

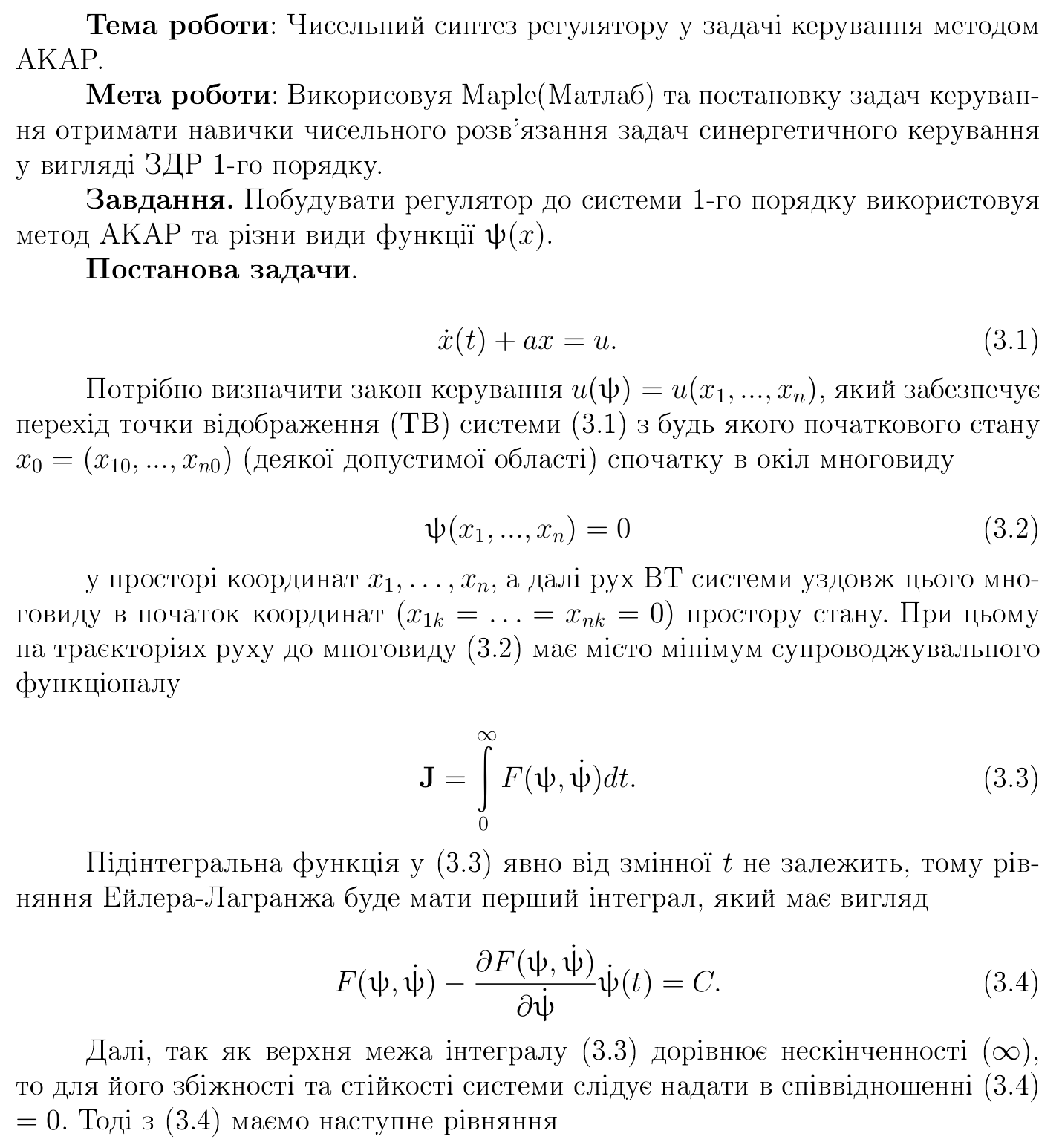
Факультет прикладної математики

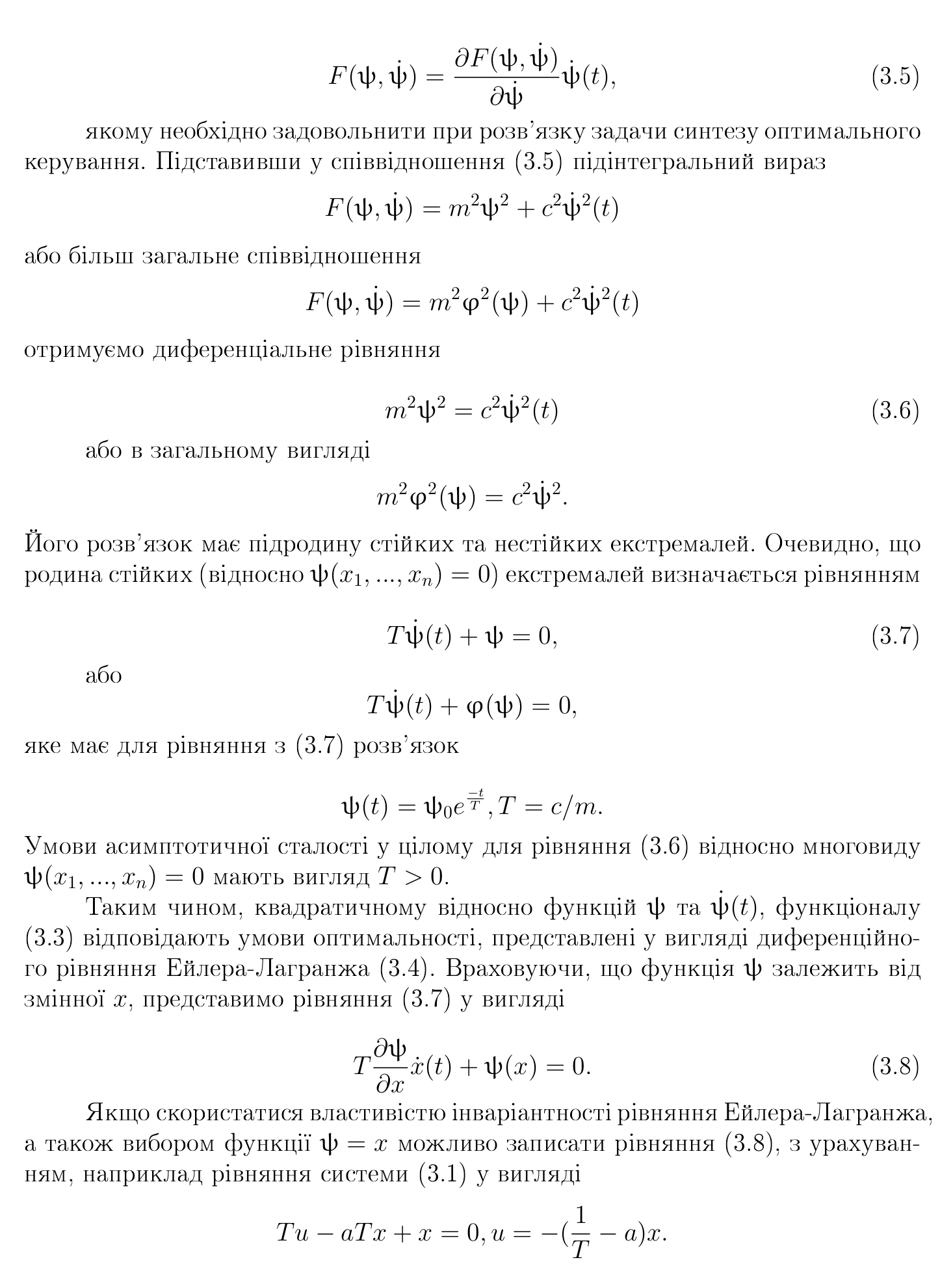
Кафедра комп'ютерних технологій

Лабораторна робота №2

|  |  |
| --- | --- |
| Виконавець: | студент групи ПК-21м-1  Панасенко Єгор  Сергійович |

# Постановка задачі





# Хід роботи

Розглянемо диференціальне рівняння:



Родина стійких екстремалей визначається рівнянням:



Візмемо:



Підставимо у рівняння:



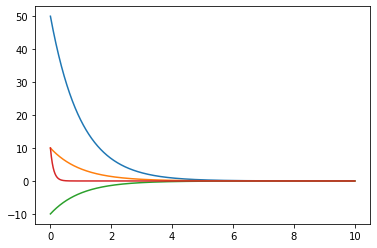


Знайдемо управління:



Підставимо у рівняння отримане управління:





Розглянемо диференціальне рівняння:



Родина стійких екстремалей визначається рівнянням:



Візмемо:



Підставимо у рівняння:



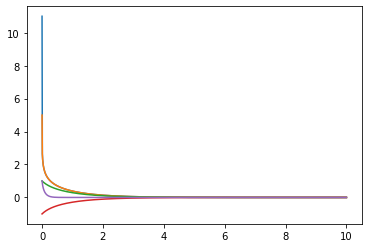


Знайдемо управління:



Підставимо у рівняння отримане управління:





# Код програми

#!/usr/bin/env python3

from sympy import \*

from sympy.solvers.ode.ode import odesimp

import matplotlib.pyplot as plt

import numpy as np

from scipy.integrate import odeint

from IPython.display import display, Latex

init\_printing(use\_latex='mathjax')

pprint = display

t, a, T = symbols("t, a, T")

x, u, psi = symbols("x, u, psi", cls=Function)

x = x(t)

psi = psi(x)

def sys(x, t, u, a, T):

return u(x, a, T) - a \* x

def find\_u\_sys(system, psi\_fun):

print("Розглянемо диференціальне рівняння:")

pprint(Eq(diff(x,t), sys(x, t, u, a, T)))

print("Родина стійких екстремалей визначається рівнянням:")

eq = Eq(T\*diff(psi, x)\*sys(x, t, u, a, T) + psi, 0)

pprint(eq)

print("Візмемо:")

pprint(Eq(psi, psi\_fun))

eq = eq.subs(psi, psi\_fun)

print("Підставимо у рівняння:")

pprint(eq)

eq = simplify(eq)

pprint(eq)

sol = solve(eq, u(x, a, T))[0]

print("Знайдемо управління:")

pprint(sol)

return lambdify([x, a, T], sol, modules='sympy')

def find\_x\_sys(system, u):

sol = sys(x, 0, u, a, T)

print("Підставимо у рівняння отримане управління:")

pprint(Eq(diff(x,t), sol))

return lambdify((x, t, a, T), sol, modules='sympy')

u\_sol = find\_u\_sys(sys, x)

x\_sol = find\_x\_sys(sys, u\_sol)

plt.figure(1)

tspan = np.linspace(0, 10, 1000)

plt.plot(tspan, odeint(x\_sol, 50, tspan, args=(10, 1)))

plt.plot(tspan, odeint(x\_sol, 10, tspan, args=(10, 1)))

plt.plot(tspan, odeint(x\_sol, -10, tspan, args=(10, 1)))

plt.plot(tspan, odeint(x\_sol, 10, tspan, args=(1, 0.1)))

plt.show(block=False)

print()

u\_sol = find\_u\_sys(sys, tanh(x))

x\_sol = find\_x\_sys(sys, u\_sol)

x\_sol = lambda x, t, a, T: -0.5\*np.sinh(2\*x)/T

tspan = np.linspace(0, 10, 1000)

plt.figure(2)

plt.plot(tspan, odeint(x\_sol, 11, tspan, args=(10, 1)))

plt.plot(tspan, odeint(x\_sol, 5, tspan, args=(10, 1)))

plt.plot(tspan, odeint(x\_sol, 1, tspan, args=(10, 1)))

plt.plot(tspan, odeint(x\_sol, -1, tspan, args=(10, 1)))

plt.plot(tspan, odeint(x\_sol, 1, tspan, args=(1, 0.1)))

plt.show()