Міністерство освіти і науки України

Національний авіаційний університет

Факультет кібербезпеки, комп'ютерної та програмної інженерії

**Імітаційне моделювання**

Лабораторна робота №4

«Основи роботи з пакетом MATLAB»

Варіант№ 38

Роботу виконав:

студент групи СП-325

Козлов Олексій

Роботу прийняла:

Нечипорук О.П.

Київ – 2020

**Моделювання випадкових чисел**

Мета лабораторної роботи

1. Вивчення основ роботи з призначеним для користувача інтерфейсом системи Matlab;
2. Придбання практичних знань і навичок програмування в системі Matlab;
3. Отримання навичок розробки та реалізації програм на основі математичної моделі об'єкта;
4. Розробка комп'ютерної моделі руху тіла.

**Хід роботи**

**Завдання**

1. Побудувати графіки декількох траєкторій руху тіла, що відповідають декільком значенням часу досягнення мети і цільову точку. Як параметри задати координати цілі (Х1, У1) і час досягнення мети, початком руху вважати точку (0,0).
2. Побудувати графік економічною траєкторії досягнення тілом цільової точки і цільову точку. Як параметри задати координати цілі (х1, у1), початком руху вважати точку (0,0). Траєкторія вважається економною, якщо мета досягається за мінімальної початковою швидкістю.

Формули:

Рівномірний рух тіла по осі х:

Рівноприскорений рух тіла по осі у:

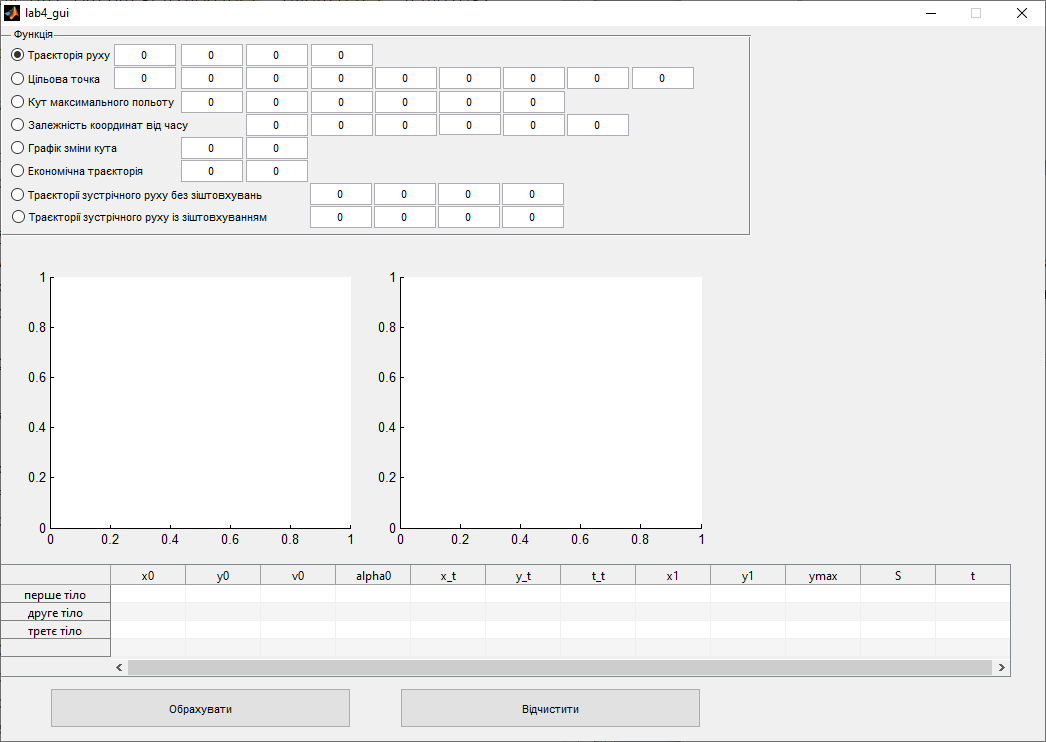
Загальний час руху тіла:

Кут початкової швидкості при направлені тіла в ціль:

Початкова швидкість при направлені тіла в ціль:

**Реалізація на мові MATLAB**

1. Інтерфейс користувача



1. Код програми
2. function table\_fill(bodys,handles)
3. Data = [zeros(1,12);zeros(1,12);zeros(1,12)];
4. for i = (1: length(bodys))
5. Data(i,:) = [bodys(i).x0,bodys(i).y0,bodys(i).v0,bodys(i).alpha,bodys(i).x\_t,bodys(i).y\_t,bodys(i).t\_t,bodys(i).x1,bodys(i).y1,bodys(i).ymax,bodys(i).S,bodys(i).t];
6. end
7. set(handles.uitable2,'Data',Data)
9. function new\_bodys = move\_in\_target\_econom(bodys)
10. G = 9.8;
11. for i = (1: length(bodys))
12. bodys(i).a = atan((bodys(i).y\_t+0.5\*G\*1^2)/bodys(i).x\_t);
13. bodys(i).alpha = bodys(i).a\*180/pi;
14. bodys(i).v0 = bodys(i).x\_t/cos(bodys(i).a)\*1;
15. bodys(i).t = (bodys(i).v0\*sin(bodys(i).a)+(bodys(i).v0^2\*sin(bodys(i).a)^2+2\*G\*bodys(i).y0)^0.5)/G;
16. bodys(i).x1 = bodys(i).x0+bodys(i).v0\*cos(bodys(i).a)\*bodys(i).t;
17. bodys(i).ymax = bodys(i).y0+bodys(i).v0\*sin(bodys(i).a)\*(bodys(i).t/2) - 0.5 \* G \* (bodys(i).t/2)^2;
18. bodys(i).ts = [0:0.1:bodys(i).t,bodys(i).t];
19. bodys(i).xs = bodys(i).x0+bodys(i).v0\*cos(bodys(i).a)\*bodys(i).ts;
20. bodys(i).ys = bodys(i).y0+bodys(i).v0\*sin(bodys(i).a)\*bodys(i).ts - 0.5 \* G \* bodys(i).ts.^2;
21. bodys(i).S = abs(bodys(i).x1-bodys(i).x0);
22. end
23. new\_bodys = bodys;
25. function new\_bodys = move\_in\_target(bodys)
26. G = 9.8;
27. for i = (1: length(bodys))
28. bodys(i).a = atan((bodys(i).y\_t+0.5\*G\*bodys(i).t\_t^2)/bodys(i).x\_t);
29. bodys(i).alpha = bodys(i).a\*180/pi;
30. bodys(i).v0 = bodys(i).x\_t/cos(bodys(i).a)\*bodys(i).t\_t;
31. bodys(i).t = (bodys(i).v0\*sin(bodys(i).a)+(bodys(i).v0^2\*sin(bodys(i).a)^2+2\*G\*bodys(i).y0)^0.5)/G;
32. bodys(i).x1 = bodys(i).x0+bodys(i).v0\*cos(bodys(i).a)\*bodys(i).t;
33. bodys(i).ymax = bodys(i).y0+bodys(i).v0\*sin(bodys(i).a)\*(bodys(i).t/2) - 0.5 \* G \* (bodys(i).t/2)^2;
34. bodys(i).ts = [0:0.1:bodys(i).t,bodys(i).t];
35. bodys(i).xs = bodys(i).x0+bodys(i).v0\*cos(bodys(i).a)\*bodys(i).ts;
36. bodys(i).ys = bodys(i).y0+bodys(i).v0\*sin(bodys(i).a)\*bodys(i).ts - 0.5 \* G \* bodys(i).ts.^2;
37. bodys(i).S = abs(bodys(i).x1-bodys(i).x0);
38. end
39. new\_bodys = bodys;
41. % --- Executes on button press in pushbutton1.
42. function pushbutton1\_Callback(hObject, eventdata, handles)
43. body = struct('x0',0,'y0',0,'alpha',0,'a',0,'v0',0,'x\_t',0,'y\_t',0,'t\_t',0,'x1',0,'y1',0,'ymax',0,'S',0,'t',0,'ts',[],'xs',[],'ys',[]);
44. bodys = [body; body; body];
46. if get(handles.radiobutton2,'Value')
47. bodys(1).x\_t = str2double(get(handles.edit5,'String'));
48. bodys(1).y\_t = str2double(get(handles.edit6,'String'));
49. bodys(1).t\_t = str2double(get(handles.edit7,'String'));
51. bodys(2).x\_t = str2double(get(handles.edit8,'String'));
52. bodys(2).y\_t = str2double(get(handles.edit9,'String'));
53. bodys(2).t\_t = str2double(get(handles.edit10,'String'));
55. bodys(3).x\_t = str2double(get(handles.edit11,'String'));
56. bodys(3).y\_t = str2double(get(handles.edit12,'String'));
57. bodys(3).t\_t = str2double(get(handles.edit13,'String'));
58. bodys = move\_in\_target(bodys);
59. table\_fill(bodys,handles)
61. hold(handles.axes1,'off')
62. cla(handles.axes2,'reset')
63. plot(handles.axes1, bodys(1).xs,bodys(1).ys)
65. grid(handles.axes1, 'on')
66. hold(handles.axes1,'on')
67. plot(handles.axes1, bodys(1).x\_t,bodys(1).y\_t,'o')
69. plot(handles.axes1, bodys(2).xs,bodys(2).ys,'red')
70. plot(handles.axes1, bodys(2).x\_t,bodys(2).y\_t,'or')
72. plot(handles.axes1, bodys(3).xs,bodys(3).ys,'black')
73. plot(handles.axes1, bodys(3).x\_t,bodys(3).y\_t,'oblack')
75. title(handles.axes1,'x(y) and targets')
76. legend(handles.axes1,{'y1(x1)','y\_1t:x\_1t','y2(x2)','y\_2t:x\_2t','y3(x3)','y\_3t:x\_3t'});
77. grid(handles.axes1, 'on')
78. end
79. if get(handles.radiobutton6,'Value')
80. bodys(1).x\_t = str2double(get(handles.edit28,'String'));
81. bodys(1).y\_t = str2double(get(handles.edit29,'String'));
83. bodys = move\_in\_target\_econom(bodys);
84. table\_fill(bodys,handles)
86. cla(handles.axes2,'reset')
87. hold(handles.axes1,'off')
88. plot(handles.axes1, bodys(1).xs,bodys(1).ys)
89. grid(handles.axes1, 'on')
90. hold(handles.axes1,'on')
91. plot(handles.axes1, bodys(1).x\_t,bodys(1).y\_t,'o')
92. title(handles.axes1,'y(x)')
93. legend(handles.axes1,{'y(x)','y\_t:x\_t'});
94. end

3. Завдання

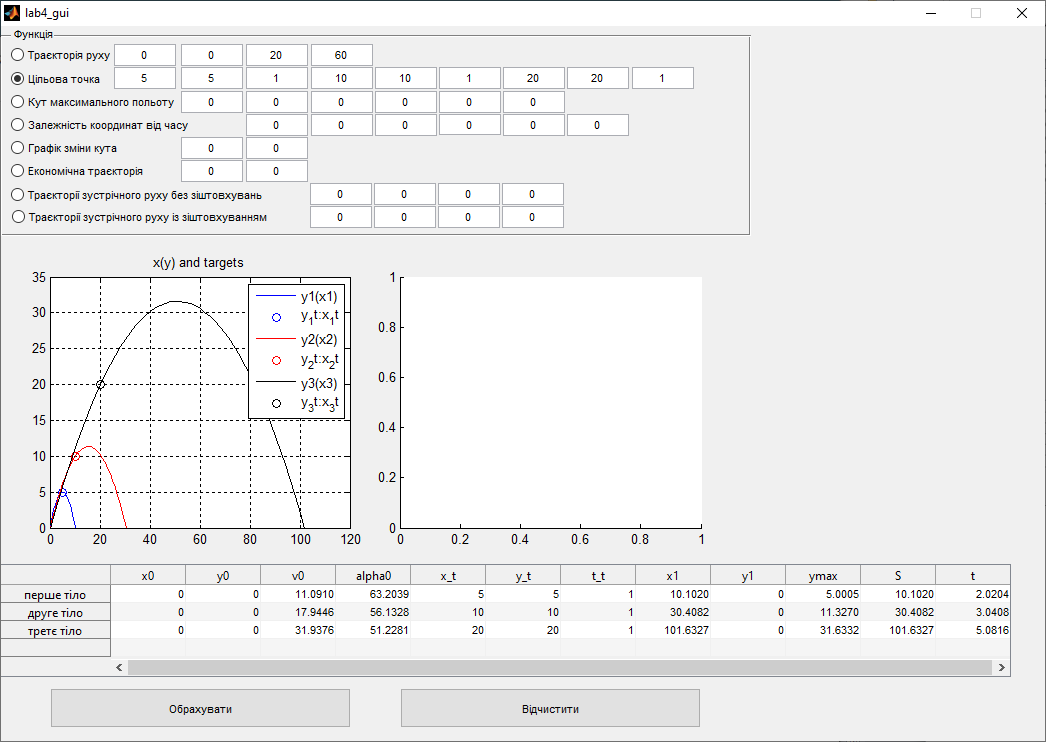
3.1. Побудувати графіки декількох траєкторій руху тіла,

що відповідають декільком значенням часу досягнення мети

і цільову точку. Як параметри задати координати цілі

(Х1, У1) і час досягнення мети, початком руху вважати точку

(0,0).



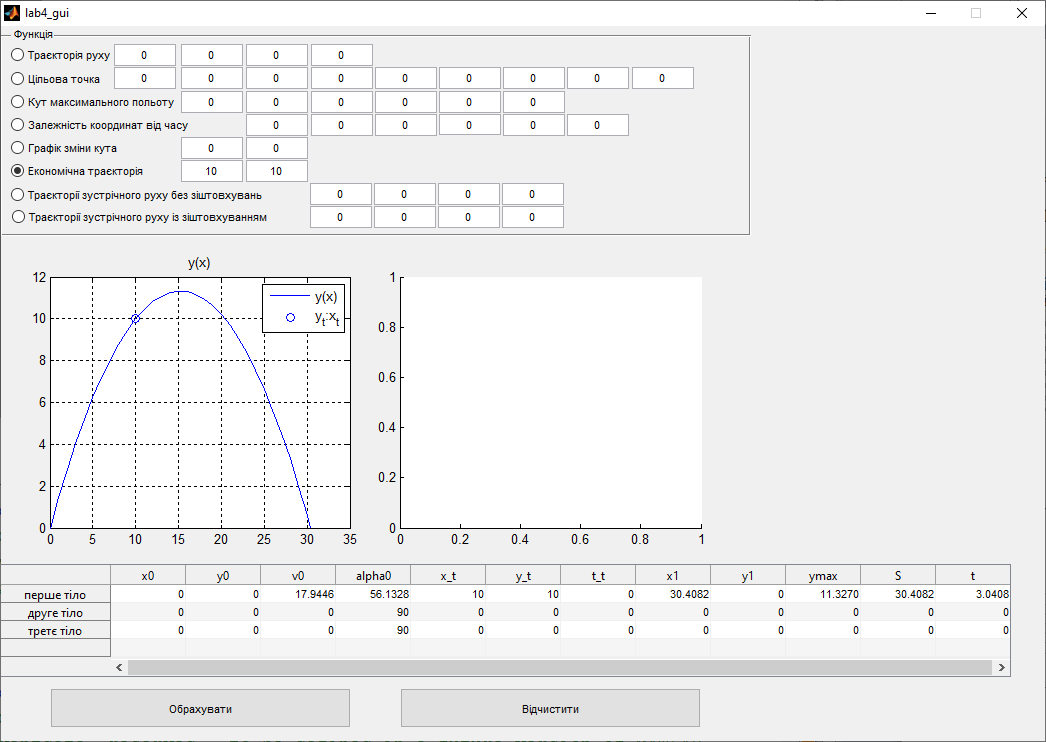
3.2. Побудувати графік економічною траєкторії досягнення тілом

цільової точки і цільову точку. Як параметри задати

координати цілі (х1, у1), початком руху вважати точку (0,0).

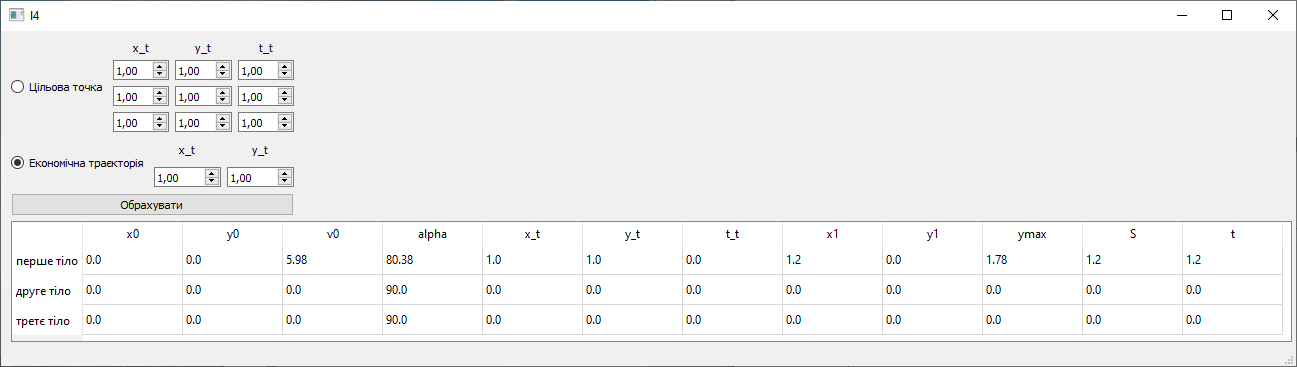
Траєкторія вважається економною, якщо мета досягається за

мінімальної початковою швидкістю.



**Реалізація на мові Python**

1. Інтерфейс користувача



1. Код програми

from PyQt5 import QtWidgets  
from PyQt5.QtWidgets import \*  
from math import \*  
  
from matplotlib import pyplot as plt  
from numpy import \*  
from IM\_L4\_ui import Ui\_MainWindow  
import sys  
  
class body:  
 x0 = 0  
 y0 = 0  
 alpha = 0  
 a = 0  
 v0 = 0  
 x\_t = 0  
 y\_t = 0  
 t\_t = 0  
 x1 = 0  
 y1 = 0  
 ymax = 0  
 S = 0  
 t = 0  
 ts = []  
 xs = []  
 ys = []  
  
 def move\_to\_target(self):  
 G = 9.8  
 self.a = atan2(self.y\_t + 0.5 \* G \* self.t\_t \*\* 2, self.x\_t)  
 self.alpha = self.a \* 180 / pi  
 self.v0 = self.x\_t / cos(self.a) \* self.t\_t  
 self.t = (self.v0 \* sin(self.a) + sqrt(self.v0 \*\* 2 \* (sin(self.a) \*\* 2) + 2 \* G \* self.y0)) / G  
 self.x1 = self.x0 + self.v0 \* cos(self.a) \* self.t  
 self.ymax = self.y0 + self.v0 \* sin(self.a) \* (self.t / 2) - 0.5 \* G \* (self.t / 2) \*\* 2  
 self.ts = arange(0, self.t, 0.1)  
 self.xs = self.x0 + self.v0 \* cos(self.a) \* self.ts  
 self.ys = self.y0 + self.v0 \* sin(self.a) \* self.ts - 0.5 \* G \* self.ts \*\* 2  
 self.S = abs(self.x1 - self.x0)  
  
 def move\_in\_target\_econom(self):  
 G = 9.8  
 self.a = atan2(self.y\_t + 0.5 \* G, self.x\_t)  
 self.alpha = self.a \* 180 / pi  
 self.v0 = self.x\_t / cos(self.a)  
 self.t = (self.v0 \* sin(self.a) + sqrt(self.v0 \*\* 2 \* (sin(self.a) \*\* 2) + 2 \* G \* self.y0)) / G  
 self.x1 = self.x0 + self.v0 \* cos(self.a) \* self.t  
 self.ymax = self.y0 + self.v0 \* sin(self.a) \* (self.t / 2) - 0.5 \* G \* (self.t / 2) \*\* 2  
 self.ts = arange(0, self.t, 0.1)  
 self.xs = self.x0 + self.v0 \* cos(self.a) \* self.ts  
 self.ys = self.y0 + self.v0 \* sin(self.a) \* self.ts - 0.5 \* G \* self.ts \*\* 2  
 self.S = abs(self.x1 - self.x0)  
  
class MyWindow(QMainWindow, Ui\_MainWindow):  
  
 def \_\_init\_\_(self):  
 QMainWindow.\_\_init\_\_(self)  
 self.setupUi(self)  
  
 self.setWindowTitle("l4")  
 self.radioButton.setChecked(True)  
 self.pushButton.clicked.connect(self.Compute\_all)  
  
 def fill\_table(self, bodys):  
 for idx in range(3):  
 self.tableWidget.setItem(idx, 0, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].x0)))))  
 self.tableWidget.setItem(idx, 1, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].y0)))))  
 self.tableWidget.setItem(idx, 2, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].v0)))))  
 self.tableWidget.setItem(idx, 3, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].alpha)))))  
 self.tableWidget.setItem(idx, 4, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].x\_t)))))  
 self.tableWidget.setItem(idx, 5, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].y\_t)))))  
 self.tableWidget.setItem(idx, 6, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].t\_t)))))  
 self.tableWidget.setItem(idx, 7, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].x1)))))  
 self.tableWidget.setItem(idx, 8, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].y1)))))  
 self.tableWidget.setItem(idx, 9, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].ymax)))))  
 self.tableWidget.setItem(idx, 10, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].S)))))  
 self.tableWidget.setItem(idx, 11, QTableWidgetItem(str(float("{0:.2f}".format(bodys[idx].t)))))  
  
 def Compute\_all(self):  
 bodys = [body(), body(), body()]  
  
 if self.radioButton.isChecked():  
 bodys[0].x\_t = float(self.doubleSpinBox.text().replace(',', '.'))  
 bodys[1].x\_t = float(self.doubleSpinBox\_4.text().replace(',', '.'))  
 bodys[2].x\_t = float(self.doubleSpinBox\_7.text().replace(',', '.'))  
  
 bodys[0].y\_t = float(self.doubleSpinBox\_2.text().replace(',', '.'))  
 bodys[1].y\_t = float(self.doubleSpinBox\_5.text().replace(',', '.'))  
 bodys[2].y\_t = float(self.doubleSpinBox\_8.text().replace(',', '.'))  
  
 bodys[0].t\_t = float(self.doubleSpinBox\_3.text().replace(',', '.'))  
 bodys[1].t\_t = float(self.doubleSpinBox\_6.text().replace(',', '.'))  
 bodys[2].t\_t = float(self.doubleSpinBox\_9.text().replace(',', '.'))  
 for current\_body in bodys:  
 current\_body.move\_to\_target()  
 self.fill\_table(bodys)  
  
 fig, ax = plt.subplots()  
 ax.plot(bodys[0].xs, bodys[0].ys, label='y1(x1)')  
 ax.plot(bodys[0].x\_t, bodys[0].y\_t, 'bo', label='x1\_t:y1\_t')  
 ax.plot(bodys[1].xs, bodys[1].ys, 'r', label='y2(x2)')  
 ax.plot(bodys[1].x\_t, bodys[1].y\_t, 'ro', label='x2\_t:y2\_t')  
 ax.plot(bodys[2].xs, bodys[2].ys, 'g', label='y3(x3)')  
 ax.plot(bodys[2].x\_t, bodys[2].y\_t, 'go', label='x3\_t:y3\_t')  
 leg = ax.legend()  
 plt.show()  
  
 if self.radioButton\_2.isChecked():  
 bodys[0].x\_t = float(self.doubleSpinBox\_10.text().replace(',', '.'))  
 bodys[0].y\_t = float(self.doubleSpinBox\_11.text().replace(',', '.'))  
 for current\_body in bodys:  
 current\_body.move\_in\_target\_econom()  
  
 self.fill\_table(bodys)  
  
 fig, ax = plt.subplots()  
 ax.plot(bodys[0].xs, bodys[0].ys, label='y(x)')  
 ax.plot(bodys[0].x\_t, bodys[0].y\_t, 'bo', label='x\_t:y\_t')  
 leg = ax.legend()  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app = QtWidgets.QApplication(sys.argv)  
 my\_app = MyWindow()  
 my\_app.show()  
 sys.exit(app.exec\_())

3. Завдання

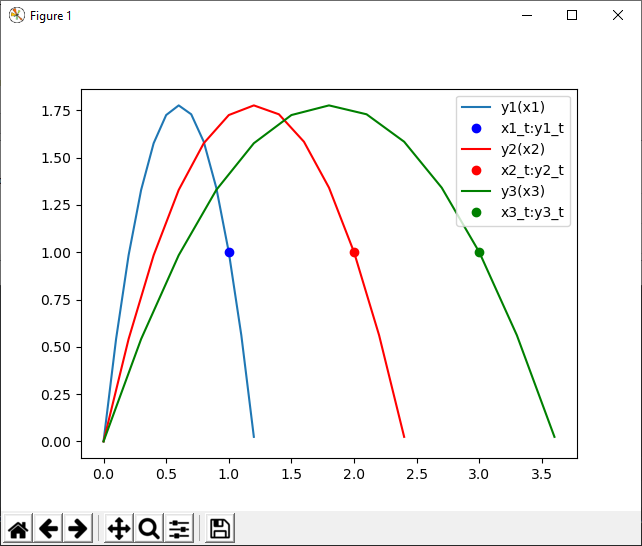
3.1. Побудувати графіки декількох траєкторій руху тіла,

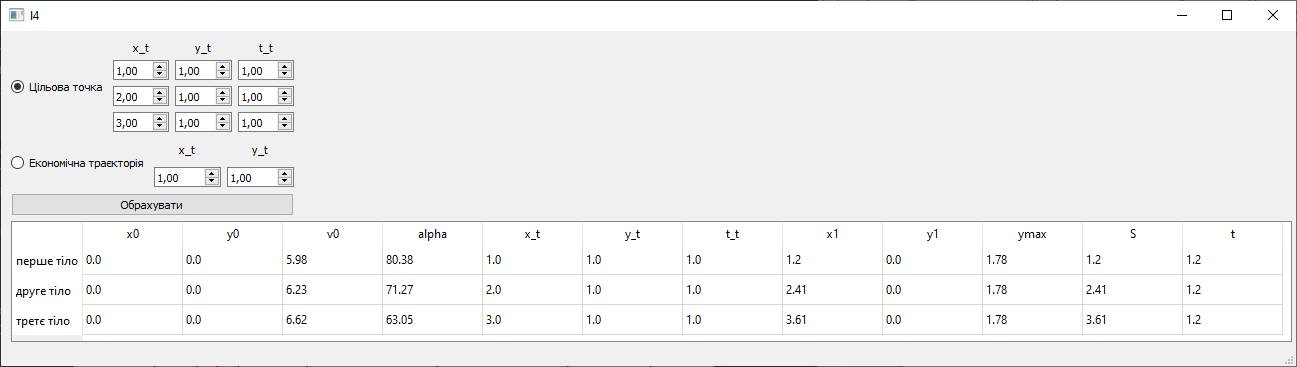
що відповідають декільком значенням часу досягнення мети

і цільову точку. Як параметри задати координати цілі

(Х1, У1) і час досягнення мети, початком руху вважати точку

(0,0).





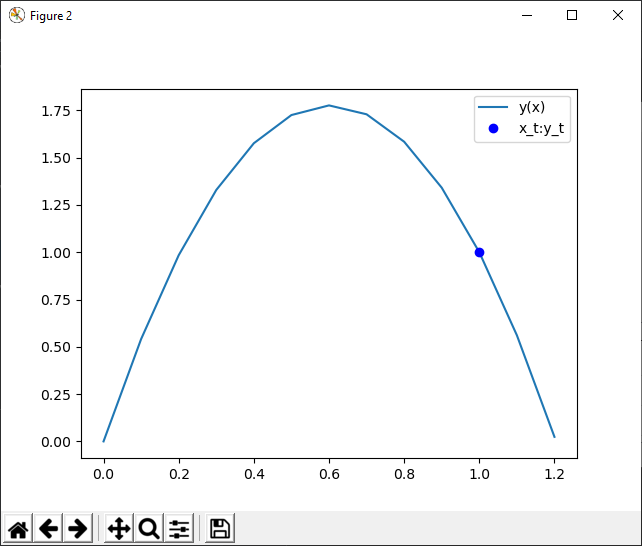
3.2. Побудувати графік економічною траєкторії досягнення тілом

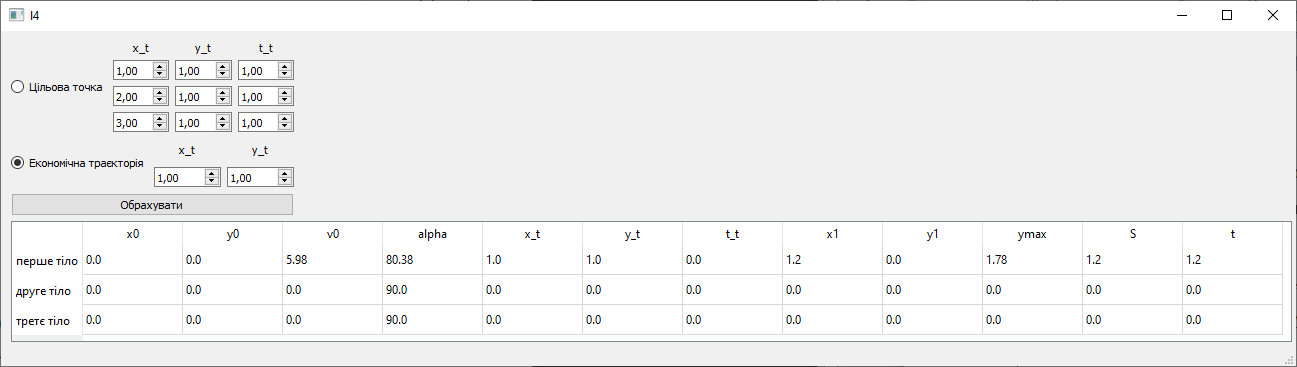
цільової точки і цільову точку. Як параметри задати

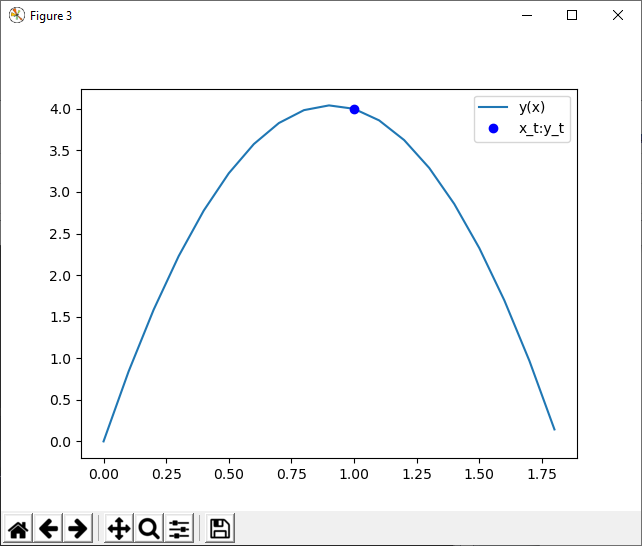
координати цілі (х1, у1), початком руху вважати точку (0,0).

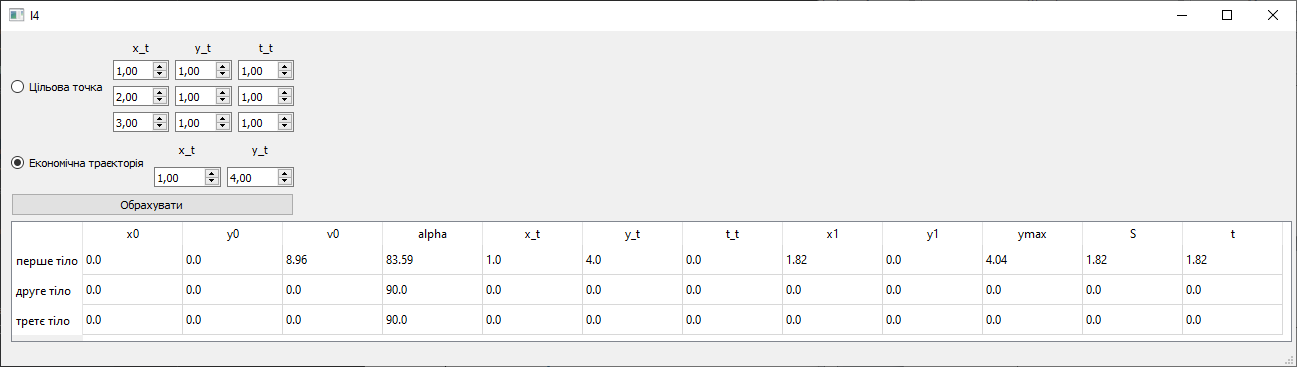
Траєкторія вважається економною, якщо мета досягається за

мінімальної початковою швидкістю.









Висновок: В ході виконання роботи ми вивчили основи роботи з інтерфейсом користувача системи Matlab, придбали практичні навички програмування на мові системи Matlab, отримали навички розробки та реалізації програм на основі математичної моделі об'єкта на прикладі розробки комп'ютерної моделі руху тіла.