МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

Кафедра компьютерных технологий и программной инженерии

ОТЧЁТ ПО ПРАКТИКЕ
ЗАЩИЩЁН С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

Ст. преп.		М.Д. Поляк
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЁТ ПО ПРАКТИКЕ

вид практики производственная

тип практики

на тему индивидуального задания Вывод управления для модели «хищник-жертва с питанием» и расширением фазового пространства

выполнен Кудряшовым Матвеем Михайловичем
фамилия, имя, отчество обучающегося в творительном падеже

по направлению подготовки 09.03.04 Программная инженерия
код наименование направления

наименование направления

направленности
код наименование направленности

наименование направленности

Обучающийся группы №	4031		М.М.Кудряшов
	номер	подпись, дата	инициалы, фамилия

Санкт–Петербург 2023

Оглавление

Вывод управления для модели «хищник-жертва с питанием»

1. Модель и цель управления

1.1. Модель

«Хищник-жертва» с мультипликативным управлением по жертвам («хищник-жертва с питанием» без расширения фазового пространства):

$$\begin{cases} \dfrac{dx_1}{dt} = \alpha_1 x_1 - \beta_1 x_1 x_2 + u(x_1, x_2) \\ \dfrac{dx_2}{dt} = -\alpha_2 x_2 + \beta_2 x_1 x_2 \end{cases}.$$

1.2. Цель управления

$\psi_1(t) = x_1(t) - x_1^*(t) \xrightarrow[t\to\infty]{} 0$, т.е. задачей управления является достижение видом x_1 целевой численности $x_1^*(t)$, где $x_1^*(t)$ удобнее всего представить константой, не зависящей от времени: $x_1^*(t) = x_1^* = const$;

2. Дискретизация

2.1. Описание модели

$$\begin{cases} x_1[k+1] = x_1[k] + hf_1(k) = F_1(k), \\ x_2[k+1] = x_2[k] + hf_2(k) = F_2(k), \end{cases}$$

$$f_1(k) = f_1(x_1[k], x_2[k]) = \alpha_1 x_1[k] - \beta_1 x_1[k] x_2[k] + u(x_1[k], x_2[k])$$

$$f_2(k) = f_2(x_1[k], x_2[k]) = -\alpha_2 x_2[k] + \beta_2 x_1[k] x_2[k]$$

2.2. Постановка задачи управления

Цель:

$$\psi[k] = x_1[k] - x_1^*[k] \xrightarrow[k\to\infty]{} 0$$

или, если $x_1^*[k] = const$, т.е. $x_1^*[k]$ не зависит от k:

$$\psi[k] = x_1[k] - x_1^* \xrightarrow[k\to\infty]{} 0$$

3. Вывод управления по методу АКАР

Уравнения Эйлера-Лагранжа:

$$T_1\psi[k] + \psi[k+1] = 0$$

$$T_1\psi[k] + x_1[k+1] - x_1^* = 0$$

$$T_1\psi[k] + x_1[k] + hf_1(k) - x_1^* = 0 T_1\psi[k] + x_1[k] - x_1^* + hf_1(k) = 0$$

$$(T_1 + 1)\psi[k] + hf_1(k) = 0$$

$$(T_1 + 1)\psi[k] + h(\alpha_1 x_1[k] - \beta_1 x_1[k] x_2[k] + u[k]) = 0$$

$$hu[k] = -(T_1 + 1)\psi[k] - h\alpha_1 x_1[k] + h\beta_1 x_1[k]x_2[k]$$

$$\boldsymbol{u[k] = \frac{-(T_1 + 1)\psi[k]}{h} - \alpha_1 x_1[k] + \beta_1 x_1[k]x_2[k]} \qquad [\mathbf{1}]$$

Если данную модель и цель управление запустить в программе то она выдаст следующее:

$$\boldsymbol{u[k] = \frac{T_1 xx - T_1 x_1}{h} - \alpha_1 x_1[k] + \beta_1 x_1[k]x_2[k]}$$

**Примечание к формуле: в программе x_1^* записано как xx*

Если провести упрощение, то формула примет вид как, управление посчитанное вручную.

Возьмём еще систему для проверки

Вторая модель для сверки

1. Модель и цель управления

1.1. Модель

«Хищник-жертва» с мультипликативным управлением по жертвам («хищник-жертва с питанием» без расширения фазового пространства):

$$\begin{cases} \dfrac{dx_1}{dt} = \alpha_1 x_1 - \beta_1 x_1 x_2 \\ \dfrac{dx_2}{dt} = -\alpha_2 x_2 + \beta_2 x_1 x_2 + u(x_1, x_2) \end{cases}.$$

1.2. Цель управления

$\psi_2(t) = x_2(t) - x_2^*(t) \xrightarrow[t\to\infty]{} 0$, т.е. задачей управления является достижение видом x_2 целевой численности $x_2^*(t)$, где $x_2^*(t)$ удобнее всего представить константой, не зависящей от времени: $x_2^*(t) = x_2^* = const$;

2. Дискретизация

2.1. Описание модели

$$\begin{cases} x_1[k+1] = x_1[k] + hf_1(k) = F_1(k), \\ x_2[k+1] = x_2[k] + hf_2(k) = F_2(k), \end{cases}$$

$$f_1(k) = f_1(x_1[k], x_2[k]) = \alpha_1 x_1[k] - \beta_1 x_1[k] x_2[k]$$
$$f_2(k) = f_2(x_1[k], x_2[k]) = -\alpha_2 x_2[k] + \beta_2 x_1[k] x_2[k] + u(x_1[k], x_2[k])$$

2.2. Постановка задачи управления

Цель:

$$\psi[k] = x_2[k] - x_2^*[k] \xrightarrow[k\to\infty]{} 0$$

или, если $x_1^*[k] = const$, т.е. $x_1^*[k]$ не зависит от k:

$$\psi[k] = x_2[k] - x_2^* \xrightarrow[k\to\infty]{} 0$$

3. Вывод управления по методу АКАР

Уравнения Эйлера-Лагранжа:

$$T_1\psi[k] + \psi[k+1] = 0$$
$$T_1\psi[k] + x_2[k+1] - x_2^* = 0$$
$$T_1\psi[k] + x_2[k] + hf_2(k) - x_1^* = 0$$
$$T_1\psi[k] + x_2[k] - x_2^* + hf_2(k) = 0$$
$$(T_1+1)\psi[k] - h(\alpha_2 x_2[k] + \beta_2 x_1[k] x_2[k] + u[k]) = 0$$
$$hu[k] = -(T_1+1)\psi[k] + h\alpha_2 x_2[k] - h\beta_2 x_1[k] x_2[k]$$
$$\boldsymbol{u[k] = \frac{-(T_1+1)\psi[k]}{h} + \alpha_2 x_2[k] - \beta_2 x_1[k] x_2[k]} \qquad [\mathbf{1}]$$

Аналогично с предыдущей моделью, проверяем результат из программы:

$$\boldsymbol{u[k] = \frac{T_1 xx - T_1 x_1}{h} + \alpha_2 x_2[k] - \beta_2 x_1[k] x_2[k]}$$

**Примечание к формуле: в программе x_1^* записано как xx*

.

Реализация программы

Для начала работы нужно инициализировать переменные, которые будут использованы в будущем для решения:

```
import sympy
# создаем символьные переменные
a1 = sympy.symbols('a1')
a2 = sympy.symbols('a2')
h = sympy.symbols('h')
t = sympy.symbols('t')
x1 = sympy.Function('x1')(t)
```

```
x2 = sympy.Function('x2')(t)
b1 = sympy.symbols('b1')
b2 = sympy.symbols('b2')
T1 = sympy.symbols('T1')
xx = sympy.symbols('xx') # xx это x1'
u = sympy.Function('u')(x1 , x2) # управление
```

Затем, средствами SymPy задаем модель и цель управления, в качестве примера возьмем первую модель:

```
# задаем модель
dx1 = a1 * x1 - b1 * x1 * x2
dx2 = -a2 * x2 + b2 * x1 * x2 + u
psi = x1 - xx # цель
```

Для дальнейшей работы нам нужно провести дискретизацию нашей модели по уравнению Эйлера

```
# Дискретизация модели
f1 = x1 + h * dx1
f2 = x2 + h * dx2
```

Задаем уравнение Эйлера-Лагранжа в дискретном виде:

```
lagrangian = T1 * psi + f2 - xx
```

Используем функцию solve для нахождения управления:

```
u_solution = sympy.solve(lagrangian, u)
```

Если запустить программу, будет выводиться следующее:

```
Модель:
dx1/dt = a1*x1(t) - b1*x1(t)*x2(t)
dx2/dt = -a2*x2(t) + b2*x1(t)*x2(t) + u(x1(t), x2(t))
Цель ψ = -xx + x1(t) --> 0

Дискретизированные f1 и f2:
f1 = h*(a1*x1(t) - b1*x1(t)*x2(t)) + x1(t)
f2 = h*(-a2*x2(t) + b2*x1(t)*x2(t) + u(x1(t), x2(t))) + x2(t)

T1*(-xx + x1(t)) + h*(-a2*x2(t) + b2*x1(t)*x2(t) + u(x1(t), x2(t))) - xx + x2(t)
Уравнение Эйлера-Лагранжа: T1*(-xx + x1(t)) + h*(-a2*x2(t) + b2*x1(t)*x2(t) + u(x1(t), x2(t))) - xx + x2(t)

Управление u = [(T1*xx - T1*x1(t) + h*(a2 - b2*x1(t))*x2(t) + xx - x2(t))/h]
```

Выводы по результатам практики

В ходе выполнения практического задания была изучена библиотека SymPy, восполнены пробелы в высшей математике и изучен метод АКАР.

Список использованной литературы

1. Лекция по методу АКАР в питоне/ Adaptive control - Lecture 6 / part 1 - Implementation of AKAR control system
2. Статья Колесниковой С.И. : Синтез системы управления нелинейным объектом второго порядка с неполным описанием (Автоматика и телемеханика №9 2018)
3. Методическое пособие от преподавателя
4. Методическое пособие по библиотеке SymPy