



AUES

Система контроля и управления ветрогенератором

Разработчик:

Магистр Алина Фазылова

Руководитель:

Профессор, PhD Теодор Илиев
(Болгария)



Описание проблемы, на решение которой направлена работа: проблема эффективного использования энергии ветра и обеспечения надежности и безопасности функционирования вертикально-осевых ветрогенератор



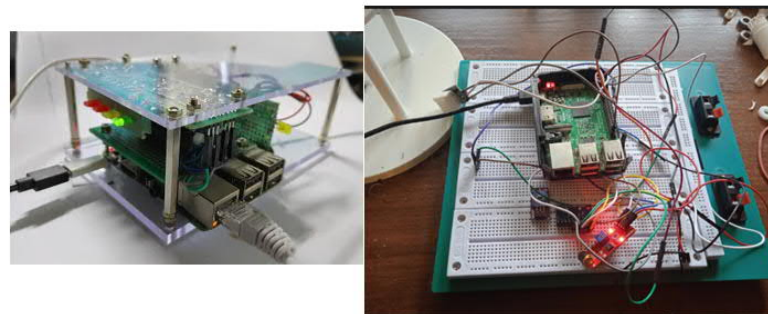
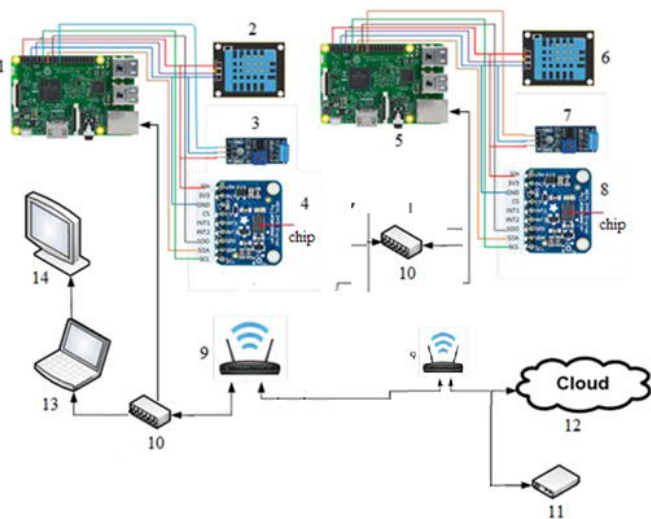
Цель работы: Создать комплексную систему управления и мониторинга ветрогенератора для эффективного использования энергии ветра и обеспечения надежности и безопасности его функционирования.

Новизна работы: разработка системы управления вертикально-осевыми ветрогенераторами, метода прогнозирования выходной мощности

Задачи исследования:

1. Разработать и проверить на промышленном ветрогенераторе систему диагностики основных узлов ветрогенератора с применением ИОТ технологий;
2. Разработать программу для точного прогноза скорости ветра в Алматы, рассчитать мощность ветрогенератора и оценить ошибку прогноза.
3. Создать новую модель управления ветрогенератором и протестировать ее на лабораторных макетах ветрогенераторов.

Настройка системы диагностики основных узлов ветрогенератора



Подключения датчиков к блоку управления для проведения экспериментальных исследований

Общая схема системы мониторинга основных узлов ветрогенератора

```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> use wind_db;
Reading table information for completion of table and column names
you can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [wind_db]> show tables;
+-----+
| Tables_in_wind_db |
+-----+
| Accelerometer      |
| temp_humi          |
| vibration          |
+-----+
4 rows in set (0.001 sec)

MariaDB [wind_db]> select * from omm;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'sel
ect * from omm' at line 2
MariaDB [wind_db]> select * from vibration;
+----+-----+
| Id | created_at |
+----+-----+
| 5  | 2021-09-07 14:23:38 |
| 6  | 2021-09-07 23:13:10 |
| 7  | 2021-09-08 14:38:01 |
+----+-----+
3 rows in set (0.001 sec)

MariaDB [wind_db]> select * from omm;
+----+-----+-----+-----+-----+-----+
| Id | temperature | humidity | wind_speed | wind_deg | created_at |
+----+-----+-----+-----+-----+-----+
| 1  | 23.90        | 0.00     | 3.25       | 4.00     | 2021-09-07 06:36:36 |
| 2  | 33.22        | 0.00     | 3.50       | 18.00    | 2021-09-07 13:15:13 |
| 3  | 33.22        | 0.00     | 3.50       | 18.00    | 2021-09-07 13:16:13 |
| 4  | 33.22        | 0.00     | 3.50       | 18.00    | 2021-09-07 13:16:25 |
| 5  | 33.22        | 0.00     | 3.50       | 18.00    | 2021-09-07 13:16:38 |
| 6  | 33.22        | 0.00     | 3.50       | 18.00    | 2021-09-07 13:19:35 |
| 7  | 33.22        | 0.00     | 3.50       | 18.00    | 2021-09-07 13:21:43 |
| 8  | 33.22        | 0.00     | 3.50       | 18.00    | 2021-09-07 13:25:47 |
| 9  | 23.54        | 0.00     | 2.99       | 188.00   | 2021-09-07 23:11:00 |
| 10 | 23.54        | 0.00     | 2.99       | 188.00   | 2021-09-07 23:11:12 |
| 11 | 32.30        | 0.00     | 2.99       | 188.00   | 2021-09-07 23:11:12 |
| 12 | 32.30        | 0.00     | 2.22       | 136.00   | 2021-09-08 14:48:30 |
+----+-----+-----+-----+-----+-----+
12 rows in set (0.001 sec)

MariaDB [wind_db]>
    
```



Ветрогенератор с установленными датчиками

ПРОГНОЗ СКОРОСТИ ВЕТРА И РАСЧЕТ ПРОГНОЗИРУЕМОЙ ВЫХОДНОЙ МОЩНОСТИ ВЕТРОГЕНЕРАТОРА

плотность воздуха равна 1.225 кг/м³, площадь поверхности ветрогенератора равна 10 м², а коэффициент мощности ветрогенератора равен 0.35.

$$P = \frac{1}{2} \rho A C_p V^3,$$

$$\alpha = \frac{2}{n+1}$$

$$\alpha = \frac{2}{10+1} = 0,1818$$

$$S_{t+1} = \alpha V_t + (1 - \alpha) \cdot S_t$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_k|$$

```

ruby
Copy code
| | Date | Time | Wind Speed (m/s) | Predicted Wind Speed (m/s) | Power Output (W) |
|---|---|---|---|---|---|
| 0 | 2023-05-01 | 00:00 | 4.6 | 4.2 | 542.28 |
| 1 | 2023-05-01 | 01:00 | 4.8 | 4.8 | 1074.24 |
| 2 | 2023-05-01 | 02:00 | 5.3 | 5.4 | 2145.30 |
| 3 | 2023-05-01 | 03:00 | 4.2 | 4.2 | 326.59 |
| 4 | 2023-05-01 | 04:00 | 4.8 | 4.8 | 1074.24 |
| 5 | 2023-05-01 | 05:00 | 5.5 | 5.4 | 1662.02 |
| 6 | 2023-05-01 | 06:00 | 4.2 | 4.2 | 326.59 |
| 7 | 2023-05-01 | 07:00 | 4.8 | 4.8 | 1074.24 |
| 8 | 2023-05-01 | 08:00 | 5.4 | 5.4 | 1662.02 |
| 9 | 2023-05-01 | 09:00 | 6.3 | 6.3 | 2569.70 |
    
```

Мощность по фактической скорости ветра (Вт)	Мощность по скорости ветра по сайту (Вт)	Мощность по скорости ветра после расчета (Вт)	MAE до обработки расчетами (%)	MAE после расчета (%)
208663.8356	158825.0781	208663.8356	9.5238	0
237081.6	237081.6	237081.6	0	0
326036.5063	337562.3781	326036.5063	1.8519	0.3704
158825.0781	158825.0781	158825.0781	0	0
237081.6	237081.6	237081.6	0	0
356665.9775	337562.3781	357030.8438	1.8519	0.8475
158825.0781	158825.0781	158825.0781	0	0
237081.6	237081.6	237081.6	0	0
337562.3781	337562.3781	337562.3781	0	0
540282.0238	540282.0238	540282.0238	0	0

Средняя ошибка по данным сайта составляет 5.11%
Средняя ошибка после расчета - 4.57%.

Час	Дата	Фактическая скорость ветра (м/с)	Прогнозируемая скорость ветра (м/с)	Выходная мощность (W)
1	23/05/01	4.6	4.2	542.28
2	23/05/01	4.8	4.8	1074.24
3	23/05/01	5.3	5.4	2145.30
4	23/05/01	4.2	4.2	326.59
5	23/05/01	4.8	4.8	1074.24
6	23/05/01	5.5	5.4	1662.02
7	23/05/01	4.2	4.2	326.59
8	23/05/01	4.8	4.8	1074.24
9	23/05/01	5.4	5.4	1662.02
10	23/05/01	6.3	6.3	2569.70

Разработка математической модели блока управления мехатронной системы шестилопастного ветрогенератора

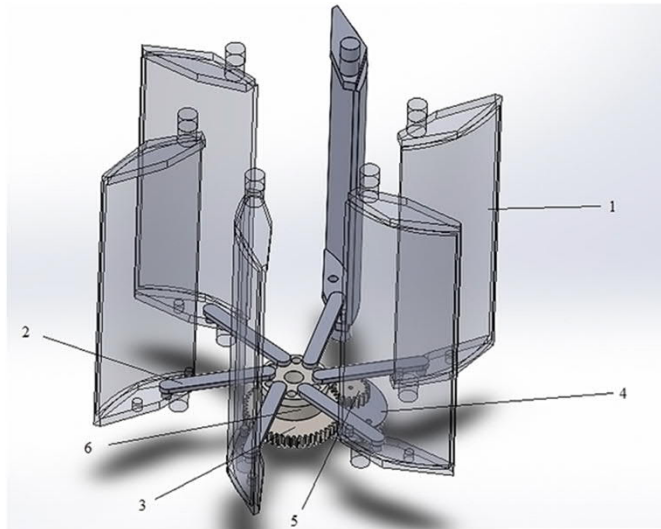


Рисунок 2.3 – Рычажная система управления:

1–лопасть, 2 – рычаг, 3 –зубчатое колесо большое, 4 – шаговый двигатель, 5 – малая шестерня, 6 – диск

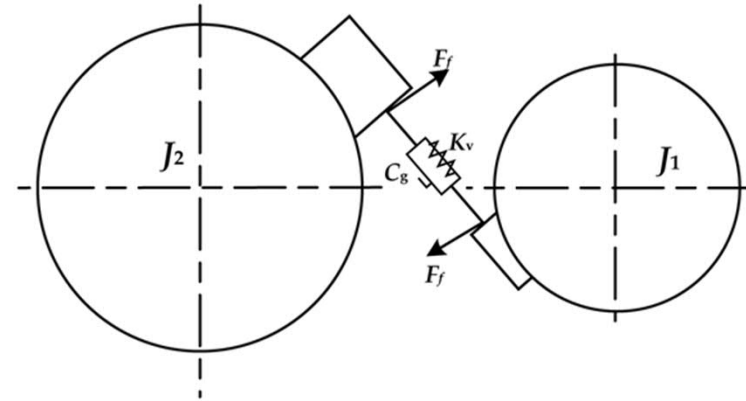
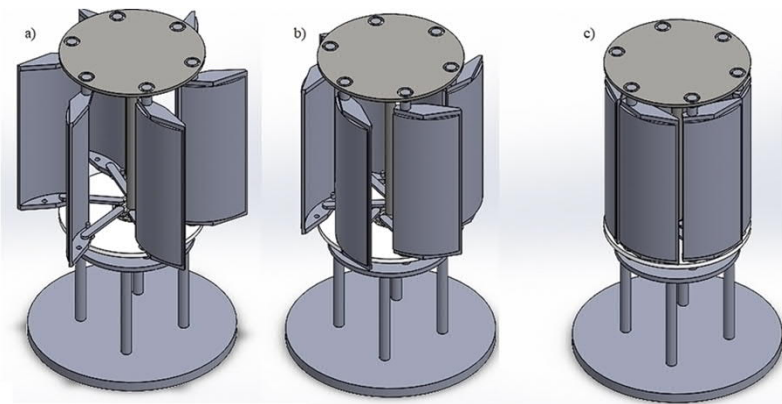


Рисунок 2.5 – зубчатая передача

K_v -динамический коэффициент, J_1, J_2 -момент инерции, C_g - коэффициент эластичности



$$J_1 \varphi''_1 + k_{g1-2}(t) * (r_{b1} \varphi_1 - r_{b2} \varphi_2) * r_{b1} + C_{g1-2} * (r_{b1} \varphi'_1 - r_{b2} \varphi'_2) * r_{b1} = T_1 - r_{f1} * [k_{g1-2}(t) * (r_{b1} \varphi_1 - r_{b2} \varphi_2) + C_{g1-2} * (r_{b1} \varphi'_1 - r_{b2} \varphi'_2)] * \mu_{1-2}$$

$$J_2 \varphi''_2 - k_{g1-2}(t) * (r_{b1} \varphi_1 - r_{b2} \varphi_2) * r_{b1} - C_{g1-2} * (r_{b1} \varphi'_1 - r_{b2} \varphi'_2) * r_{b1} = T_2 - r_{f1} * [k_{g1-2}(t) * (r_{b1} \varphi_1 - r_{b2} \varphi_2) + C_{g1-2} * (r_{b1} \varphi'_1 - r_{b2} \varphi'_2)] * \mu_{1-2}$$

Разработка математической модели блока управления мехатронной системы шестилопастного ветрогенератора

Построение динамической модели управления положением лопастей

```
import numpy as np
def system_equations(t, V):
    V1, V2, V3, V4 = V
    m1 = 5
    m2 = 8
    r_b1 = 0.3
    r_b2 = 0.8
    k = 100
    C = 5
    J1 = 2
    J2 = 1.5
    T1 = 50
    T2 = 60
    mu = 0.2
    r_f1 = 1
    r_f2 = 1
    V1_dot = V2
    V2_dot = (T1 - r_f1 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
V2 - r_b2 * V4))) * mu / J1 - (k * (r_b1 * V1 - r_b2 * V2) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J1
    V3_dot = V4
    V4_dot = (T2 - r_f2 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
V2 - r_b2 * V4))) * mu / J2 - (k * (r_b1 * V1 - r_b2 * V3) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J2
    return [V1_dot, V2_dot, V3_dot, V4_dot]
V_initial = [0, 0, 0, 0]
t_start = 0
t_end = 10
dt = 0.01
t = np.arange(t_start, t_end, dt)
V_solution = np.zeros((len(t), 4))
V_solution[0] = V_initial
for i in range(len(t) - 1):
    k1 = dt * np.array(system_equations(t[i], V_solution[i]))
    k2 = dt * np.array(system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k1))
    k3 = dt * np.array(system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k2))
    k4 = dt * np.array(system_equations(t[i] + dt, V_solution[i] +
k3))
    V_solution[i + 1] = V_solution[i] + (1 / 6) * (k1 + 2 * k2 + 2 *
k3 + k4)
for i in range(len(t)):
    print(f"time: {t[i]}, Values: {V_solution[i]}")
```

Переменные V1, V2, V3 и V4 представляют собой угловые перемещения и скорости шестеренок, участвующих в передаче

- масса первого зубчатого колеса: $m_1 = 5$ кг;
- радиус первого зубчатого колеса: $r_{b1} = 0.3$ м;
- коэффициент жесткости: $k = 100$ Н/м;
- внешняя сила Q1: $Q_1 = 50$ Н;
- коэффициент трения: $\mu = 0.2$.
- момент инерции первого зубчатого колеса: $J_1 = 2$ кг·м²;
- масса второго зубчатого колеса: $m_2 = 8$ кг;
- радиус второго зубчатого колеса: $r_{b2} = 0.8$ м;
- коэффициент диссипации: $C = 5$ Н·с/м;
- внешняя сила Q2: $Q_2 = 60$ Н;
- момент инерции второго зубчатого колеса: $J_2 = 1.5$ кг·м²;

```
1 import numpy as np
2 def system_equations(t, V):
3     V1, V2, V3, V4 = V
4     m1 = 5
5     m2 = 8
6     r_b1 = 0.3
7     r_b2 = 0.8
8     k = 100
9     C = 5
10    J1 = 2
11    J2 = 1.5
12    T1 = 50
13    T2 = 60
14    mu = 0.2
15    r_f1 = 1
16    r_f2 = 1
17    V1_dot = V2
18    V2_dot = (T1 - r_f1 * (k * (r_b1 * V1 - r_b2 * V3)
+ C * (r_b1 * V2 - r_b2 * V4))) * mu / J1 - (k
+ C * (r_b1 * V2 - r_b2 * V4) * r_b1) / J1
```

Время: 0.0, Значения: [0. 0. 0. 0.]

Время: 0.01, Значения: [0.00026136 0.05344392 0.00040142 0.08043159]

Время: 0.02, Значения: [0.00109376 0.11435906 0.00161181 0.16181691]

Время: 0.03, Значения: [0.00257659 0.18369785 0.00364137 0.24428594]

Время: 0.04, Значения: [0.00479931 0.26252566 0.00650155 0.32795972]

Время: 0.05, Значения: [0.00786266 0.35203393 0.01020496 0.41294893]

Время: 0.06, Значения: [0.01187999 0.45355467 0.01476525 0.49935215]

Время: 0.07, Значения: [0.01697872 0.56857675 0.020197 0.58725388]

Время: 0.08, Значения: [0.02330202 0.6987639 0.02651555 0.67672225]

Время: 0.09, Значения: [0.03101068 0.84597486 0.03373683 0.76780635]

Время: 0.1, Значения: [0.04028512 1.01228574 0.04187716 0.86053324]

Время: 0.11, Значения: [0.05132772 1.20001495 0.05095299 0.95490444]

Время: 0.12, Значения: [0.06436536 1.41175092 0.06098065 1.050892 1

Расчет угла поворота лопастей вертикально-осевого ветрогенератора в зависимости от времени включения мотора

```
import numpy as np
import pandas as pd
def gear_system_equations(t, V):
    V1, V2, V3, V4 = V
    m1 = 10
    m2 = 8
    r_b1 = 1
    r_b2 = 0.8
    k = 100
    C = 5
    J1 = 2
    J2 = 1.5
    T1 = 50
    T2 = 60
    mu = 0.2
    r_f1 = 1
    r_f2 = 1
    V1_dot = V2
    V2_dot = (T1 - r_f1 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
(r_b1 * V2 - r_b2 * V4))) * mu / J1 - (k * (r_b1 * V1 - r_b2 * V3) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J1
    V3_dot = V4
    V4_dot = (T2 - r_f2 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
(r_b1 * V2 - r_b2 * V4))) * mu / J2 - (k * (r_b1 * V1 - r_b2 * V3) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J2
    return [V1_dot, V2_dot, V3_dot, V4_dot]
def lever_system_equations(t, theta):
    motor_torque = 10
    motor_ramp_time = 2
    lever_length = 1
    blade_length = 0.5
    lift_coefficient = 0.1
    if t < motor_ramp_time:
        motor_torque_current = motor_torque * t / motor_ramp_time
    else:
        motor_torque_current = motor_torque
    angular_acceleration = motor_torque_current / (blade_length *
lever_length)
    theta_dot = angular_acceleration * t
    return theta_dot
V_initial = [0, 0, 0, 0]
theta_initial = 0
t_start = 0
t_end = 10
dt = 0.01
t = np.arange(t_start, t_end, dt)
V_solution = np.zeros((len(t), 4))
V_solution[0] = V_initial
theta_solution = np.zeros_like(t)
theta_solution[0] = theta_initial
for i in range(len(t) - 1):
    k1 = dt * np.array(gear_system_equations(t[i], V_solution[i]))
    k2 = dt * np.array(gear_system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k1))
    k3 = dt * np.array(gear_system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k2))
    k4 = dt * np.array(gear_system_equations(t[i] + dt,
V_solution[i] + k3))
    V_solution[i + 1] = V_solution[i] + (1 / 6) * (k1 + 2 * k2 + 2 *
k3 + k4)
    theta_dot = lever_system_equations(t[i], theta_solution[i])
    theta_solution[i + 1] = theta_solution[i] + dt * theta_dot
R = 0.8
D = 0.5
V_disk = [(R / D) * V[1] for V in V_solution]
data = {'Time (sec)': t, 'V1': V_solution[:, 0], 'V2': V_solution[:,
1], 'V3': V_solution[:, 2], 'V4': V_solution[:, 3], 'V_disk': V_disk,
'Blade angle (rad)': theta_solution}
df = pd.DataFrame(data)
df['Blade angle (deg)'] = np.rad2deg(df['Blade angle (rad)'])
print(df)
```

	Время (сек)	Угол поворота лопастей (рад)	Угол поворота лопасти (град)
0	0.00	0.000000	0.000000
1	0.02	0.000040	0.002303
2	0.04	0.000322	0.018427
3	0.06	0.001085	0.062190
4	0.08	0.002573	0.147414
5	0.10	0.005025	0.287918
6	0.12	0.008683	0.497523
7	0.14	0.013789	0.790048
8	0.16	0.020583	1.179314
9	0.18	0.029307	1.679141
10	0.20	0.040201	2.303348
11	0.22	0.053508	3.065756
12	0.24	0.069467	3.980185
13	0.26	0.088322	5.060455
14	0.28	0.110312	6.320387
15	0.30	0.135678	7.773799
16	0.32	0.164663	9.434513
17	0.34	0.197508	11.316348
18	0.36	0.234452	13.433125
19	0.38	0.275739	15.798663
20	0.40	0.321608	18.426783
21	0.42	0.372302	21.331305
22	0.44	0.428060	24.526049
23	0.46	0.489126	28.024834
24	0.48	0.555739	31.841482
25	0.50	0.628141	35.989811
26	0.52	0.706573	40.483643
27	0.54	0.791276	45.336797
28	0.56	0.882492	50.563094
29	0.58	0.980462	56.176352
30	0.60	1.085427	62.190394
31	0.62	1.197628	68.619038
32	0.64	1.317307	75.476105
33	0.66	1.444704	82.775414

```
import numpy as np
from scipy.optimize import curve_fit
time = np.array([0.00, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16,
0.18, 0.20, 0.22, 0.24, 0.26, 0.28, 0.30, 0.32, 0.34, 0.36, 0.38, 0.40,
0.42, 0.44, 0.46, 0.48, 0.50, 0.52, 0.54, 0.56, 0.58, 0.60, 0.62, 0.64,
0.66])
angle_rad = np.array([0.000000, 0.000040, 0.000322, 0.001085, 0.002573,
0.005025, 0.008683, 0.013789, 0.020583, 0.029307, 0.040201, 0.053508,
0.069467, 0.088322, 0.110312, 0.135678, 0.164663, 0.197508, 0.234452,
0.275739, 0.321608, 0.372302, 0.428060, 0.489126, 0.555739, 0.628141,
0.706573, 0.791276, 0.882492, 0.980462, 1.085427, 1.197628, 1.317307,
1.444704])
def polynomial_func(t, a, b, c):
    return a * t**2 + b * t + c
params, _ = curve_fit(polynomial_func, time, angle_rad)
a, b, c = params
print(f" Blade angle (rad) = {a:.4f} * t^2 + {b:.4f} * t + {c:.4f}")
```

Рисунок 2.9 – полиномиальная аппроксимация данных

$$\varphi(t) = at^2 + bt + c$$

Здесь t представляет время включения мотора,
(a = 293.6769, b = -81.0778, c = 4.9338)

Разработка блока управления первой модели

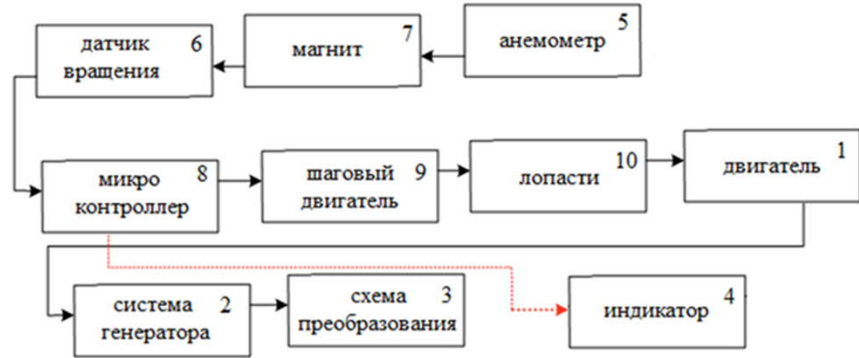


Рисунок 2.10 – Общая схема системы управления ротором

t	0	0.001	0.002	0.003	0.004	0.005
y (B)	0	0.342	1.101	2.008	2.914	3.747

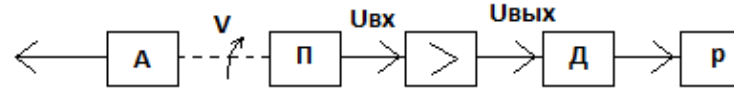
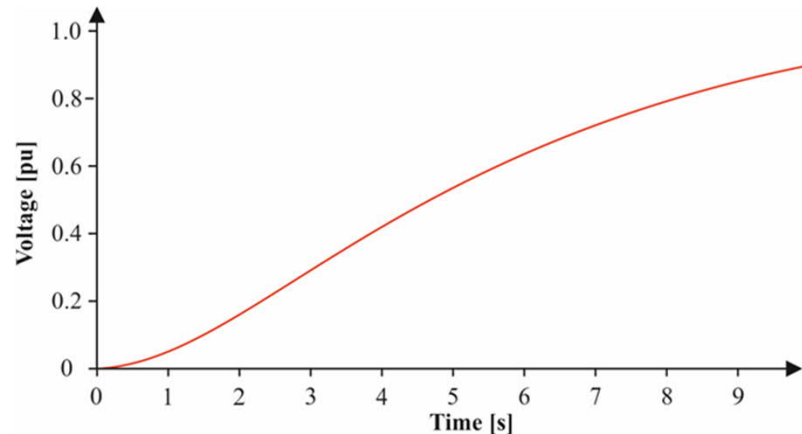
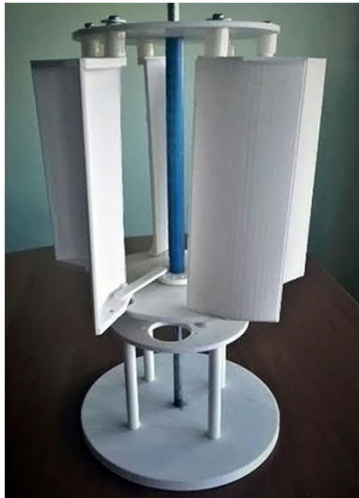


Рисунок 2.11 – Схема управления лопастями ветрогенератора

α_1 – угол отклонения, обрабатываемый микроконтроллером с анемометра;
 α_2 – угол поворота лопастей.

$$W_1(p) = \frac{u_{вх}}{\alpha_1} = K_1 \quad W_2(p) = \frac{u_{ввых}}{u_{вх}} = K_2 \quad W_3(p) = \frac{K_3}{p(1 + T_d p)} \quad W_4(p) = \frac{K_4}{p(1 + T_o p)}$$

$$W(p) = \frac{K}{p^2(1 + T_d p)(1 + T_o p)}$$

$$T_d T_o p^4 + (T_d + T_o)p^3 + p^2 + K = 0$$

$$\begin{cases} 1.101B_1 + 0.342B_2 + 0B_3 + 2.008 = 0 \\ 2.008B_1 + 1.101B_2 + 0.342B_3 + 2.914 = 0 \\ 2.914B_1 + 2.008B_2 + 1.101B_3 + 3.747 = 0 \end{cases}$$

$$\lambda^3 - 2.4218 * \lambda^2 + 1.9255 * \lambda - 0.5037 = 0$$

$$\lambda_1 = 0.67 = p;$$

$$\lambda_2 = 0.7518 = q.$$

$$W(p) = \frac{7.501}{(2.497p + 1)(3.505p + 1)}$$

Разработка блока управления

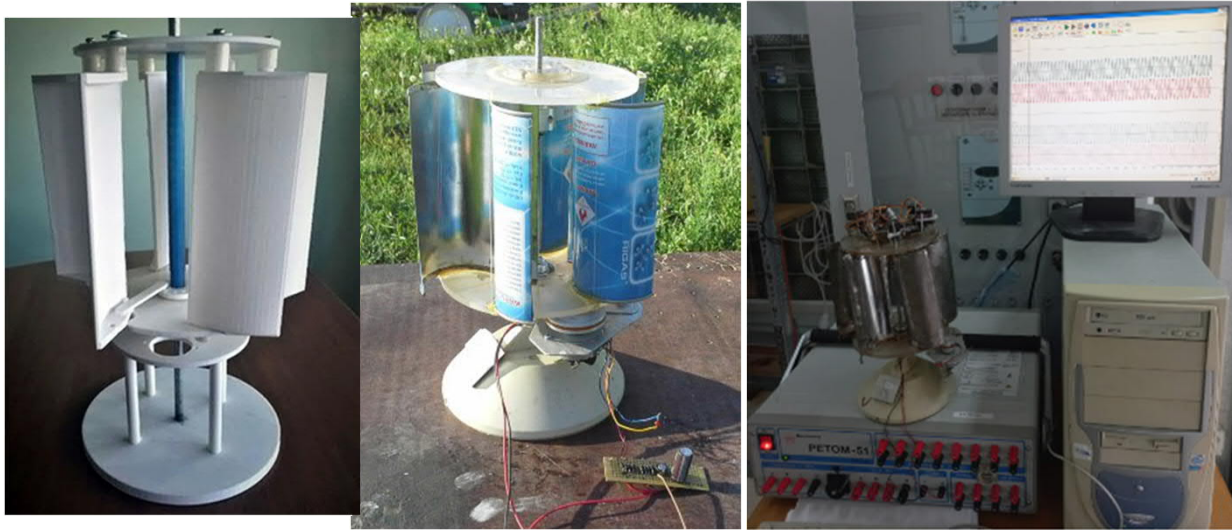
```
import time
def calculate_desired_angle(time):
    # Function to calculate the desired angle of blade rotation
    return 90.0 * (time / 7.0)
def pi_controller(current_angle, desired_angle, kp, ki, prev_error,
integral):
    # PI controller for controlling the blade position
    error = desired_angle - current_angle
    integral += error
    output = kp * error + ki * integral
    return output, error, integral
def control_system():
    # PI controller parameters
    kp = 1.0
    ki = 0.1
    # Initial values
    current_angle = 0.0
    integral = 0.0
    prev_error = 0.0
    start_time = time.time()
    while True:
        current_time = time.time() - start_time
        if current_time <= 7.0:
            # Calculate the desired angle of blade rotation
            desired_angle = calculate_desired_angle(current_time)
            # Use the PI controller to control the blade position
            control_signal, prev_error, integral =
pi_controller(current_angle, desired_angle, kp, ki, prev_error,
integral)
            # Update the current blade position (simulate the
operation of the wind turbine)
            current_angle += control_signal
            # Output the current time and blade angle
            print("Time: {:.2f}s, Angle: {:.2f}
degrees".format(current_time, current_angle))
        else:
            break
        # Delay to simulate real-time operation
        time.sleep(0.5)
    # Run the control system
    control_system()
```

```
23 # Calculate the desired angle of blade rotation
24 desired_angle = calculate_desired_angle(current_time)
25 # Use the PI controller to control the blade position
26 control_signal, prev_error, integral = pi_controller
    (current_angle, desired_angle, kp, ki, prev_error,
    integral)
27 # Update the current blade position (simulate the
    operation of the wind turbine)
28 current_angle += control_signal
29 # Output the current time and blade angle
30 print("Time: {:.2f}s, Angle: {:.2f} degrees".format
    (current_time, current_angle))
31 else:
32     break
33 # Delay to simulate real-time operation
34 time.sleep(0.5)
35 # Run the control system
36 control_system()
37
```

Time: 1.00s, Angle: 14.10 degrees
Time: 1.50s, Angle: 21.06 degrees
Time: 2.02s, Angle: 28.14 degrees
Time: 2.52s, Angle: 35.00 degrees
Time: 3.02s, Angle: 41.81 degrees
Time: 3.52s, Angle: 48.59 degrees
Time: 4.02s, Angle: 55.33 degrees
Time: 4.52s, Angle: 62.04 degrees
Time: 5.02s, Angle: 68.73 degrees
Time: 5.52s, Angle: 75.39 degrees
Time: 6.02s, Angle: 82.03 degrees
Time: 6.52s, Angle: 88.65 degrees

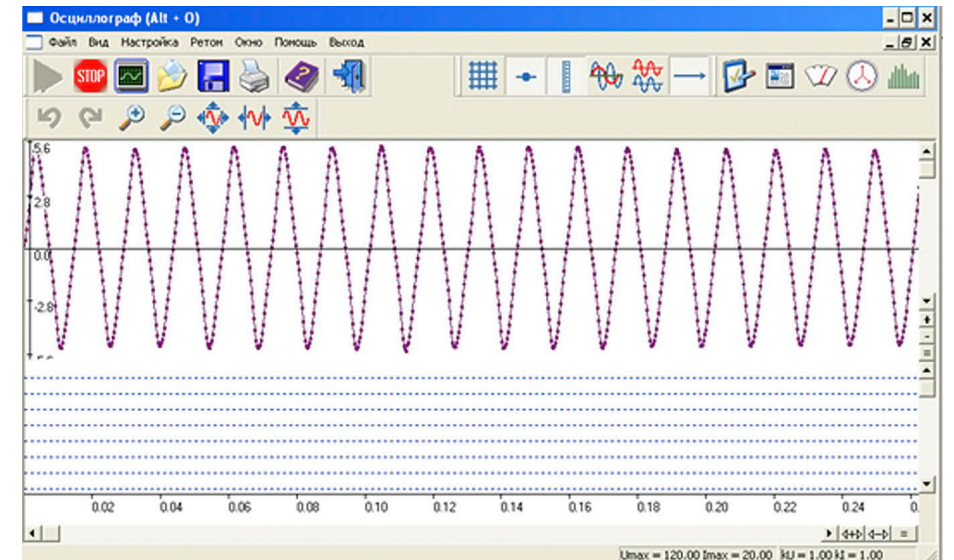
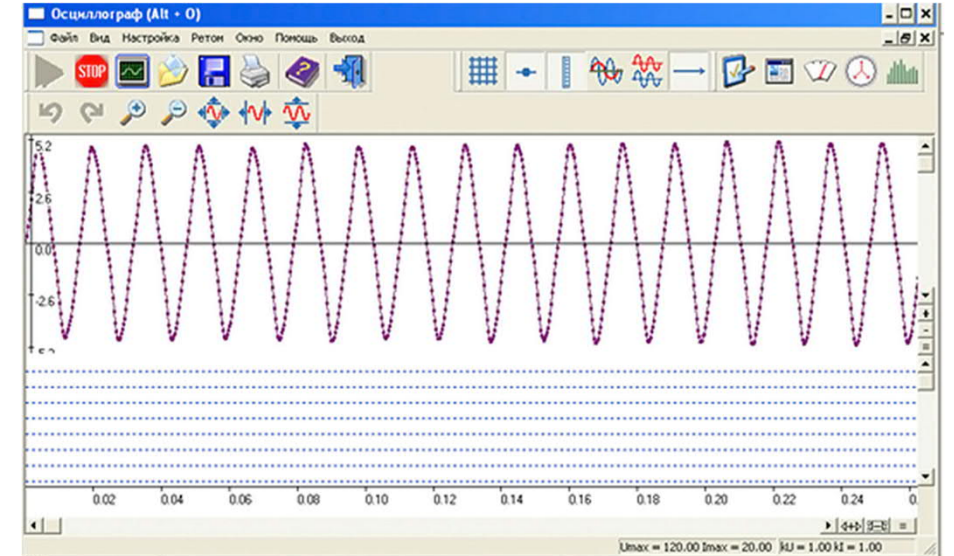
Результат работы программы по управлению лопастями ветрогенератора

Экспериментальное сравнение модели управления с лабораторной моделью ветрогенератора без системы управления



Проведение эксперимента на программно-техническом комплексе Petom-51

**разработанная система управления способна
увеличить эффективность использования
ветрогенератора на 7.69%**



ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ ОТ ВНЕДРЕНИЯ

Сравнение суточной выработки электрической энергии вертикально-осевым ветрогенератором с внедренным блоком управления и без него

суточная выработка
электроэнергии, Вт



- выработка электроэнергии ветрогенератором без внедренного блока управления
- выработка электроэнергии ветрогенератором с разработанным блоком управления

Экономия составляет 3
ветрогенератора номинальной
мощностью 10 кВт

Публикации

1. Fazylova A., Stoyanov, I.; Iliev, T., Yestemessova G. A Dynamic Control model of the blades position for the vertical-axis wind generator by a program method. Inventions 2023,8,120, <https://doi.org/10.3390/inventions8050120/>, **Scopus процентиљ- 83%, Q1**
2. Fazylova, A.; Tultayev, B.; Iliev, T.; Stoyanov, I.; Beloev, I. Development of a Control Unit for the Angle of Attack of a Vertically Axial Wind Turbine. Energies 2023, 16, 5202. <https://doi.org/10.3390/en1613520225/>, **Scopus процентиљ- 82%, Q1**
3. **Fazylova, A.**, Balbayev, G., Tultayev, B. SYSTEM OF SHORT-TERM FORECASTING OF WIND TURBINE OUTPUT POWER CONSUMPTION, 2022, **Scopus процентиљ- 40%**
4. **Alina Fazylova**, Michail Malamatoudis, Panagiotis Kogias, Optimization of the blade profile of a vertical axis wind turbine based on aerodynamic analysis, E3S Web of Conferences 404, 02002 (2023), <http://doi.org/10.1051/e3sconf/202340402002> (**Conference Proceedings Scopus процентиљ- 25%**)
5. **Fazylova, A.**, Iliev, T, A New Type of Wind Generator Blades Mechanical Automatic Control System, 2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES 2022) 24 – 26 November, 2022, Veliko Tarnovo, Bulgaria <http://doi.org/10.1109/CIEES55704.2022.9990872> (**Conference Proceedings Scopus процентиљ- 25%**)
6. **Fazylova, A.**, Balbayev, G, Ilieva D, Aliyarova M. Analysis of rotors' critical mode of operation to be employed in the design of a wind generation control unit, E3S Web of Conferences 180, 02001 (2020) TE-RE-RD 2020, <http://doi.org/10.1051/e3sconf/202018002001> (**Conference Proceedings Scopus процентиљ- 25%**)
7. **Fazylova, A.**, Iliev, T, Рычажная система управления лопастями ветрогенератора при высоких скоростях ветра Патент KZ№ 7392, Опубликовано: 26.08.2022, рег. Номер 2022/0595.2 (**Патент на полезную модель**)
8. **Fazylova, A.**, Assessment of the prospect for the use of renewable energy sources in the world and Kazakhstan, The Bulletin of Kazakh Academy of Transport and Communications named after M. Tynyshpayev, Vol. 118, No.3 (2021) pp.124-134 (Статья)
9. **Fazylova, A.**, Calculation of the parameters of the wind turbine rotor eddy current sensor, The Bulletin of Kazakh Academy of Transport and Communications named after M. Tynyshpayev, Vol. 118, No.3 (2021) pp.134-141, (Статья)
10. **Fazylova, A.**, Calculation of the parameters of the wind turbine rotor temperature sensor, The Bulletin of Kazakh Academy of Transport and Communications named after M. Tynyshpayev, Vol. 118, No.3 (2021) pp.141 – 149 (Статья)

Заключение

1. Написана программа на Python для получения данных о прогнозе скорости ветра в Алматы с сайта Accuweather. Данные использовались для расчета мощности ветрогенератора. Улучшена точность прогноза. Сравнение прогнозной и фактической мощности показало снижение средней ошибки с 5.11% до 4.57% после сглаживания данных.
2. Разработаны и исследованы системы управления лопастями вертикально-осевого ветрогенератора способная увеличить эффективность вертикально осевого ветрогенератора на 7.69%.
3. В ходе работы была разработана и проверена на промышленном ветрогенераторе система диагностики основных узлов ветрогенератора с применением ИОТ технологий. Данная сеть доказала свою работоспособность на экспериментальных исследованиях по получению данных с датчиков.

Спасибо за внимание