

Таблица 1. Основные параметры систем размыва нефти для резервуаров PBC5000-PBC2000

Основные показатели	PBC-5000*(3)	PBC-10000	PBC-20000
	Др=22.8 м	Др=28.5 м	Др=45.6 м
1. Расход (производительность закачки) нефти через систему, м ³ /ч	200–500	500–1250	1000–2500
2. Давление закачиваемой нефти (на входе в резервуар):			
при 1 м и взлива нефти в резервуаре, кгс/см ² *(1)	1.0–2.5	1.0–2.5	1.5–3.0
при полном взливе нефти в резервуаре, кгс/см ² *(1)	2.0–3.5	2.5–4.0	2.5–4.0
3. Потребляемая мощность (на входе в резервуар):			
при 1 м взлива нефти в резервуаре, кВт*(1)	5–40	15–40	50–240
при полном взливе нефти в резервуаре, кВт*(1)	15–60	35–60	80–320
4. Количество размывающих устройств (сопел при грузенных веерных кольцевых типа СГ1ВК-100М), шт.	2	5	10
5. Рабочая жидкость	нефть, закачиваемая в резервуар		
6. Скорость истечения нефти из сопел:			
рабочая, м/с	13–24	13–24	13–24
допустимая, м/с*(2)	до 67	до 46	до 40

Вывод

Для обеспечения безопасности при работе с виброструйным магнитным активатором необходимо соблюдать все правила и ГОСТы при работах в нефтегазовой промышленности. Данная отрасль производства характеризуется высокими рисками возникновения чрезвычайных ситуаций, поэтому несоблюдение правил и стандартов безопасности могут повлечь за собой технологические потери и человеческие жертвы.

СПИСОК ЛИТЕРАТУРЫ

1. Основы виброструйной магнитной активации жидких сред / В.А. Данекер, В.В. Доленко // Современная техника и технологии, 28 февраля – 3 марта 2000 г., Томск.
2. Прозорова И.В., Лоскутова Ю.В., Юдина Н.В., Рикконен С.В., Данекер В.А. Изменение реологических свойств нефтей под воздействием виброструйной магнитной активации // Автоматизация и информационное обеспечение технологических процессов в нефтяной промышленности. – 2002. – № 2.

МОДЕЛИРОВАНИЕ РАБОТЫ АСИНХРОННОГО ЭЛЕКТРОДВИГАТЕЛЯ СРЕДСТВАМИ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

М.С. Кишканова

Томский политехнический университет, ИШЭ, ОЭЭ, группа 5А38

Научный руководитель: И.А. Розаев, к.т.н., старший преподаватель ОЭЭ ИШЭ ТПУ

В современном мире технологии развиваются с невероятной скоростью, и научные исследования в области электротехники и машиностроения становятся всё более актуальными. Одним из ключевых аспектов этих исследований является моделирование электродвигателей, которое позволяет предсказать их поведение в различных условиях и оптимизировать их характеристики [1].

Моделирование электродвигателей на языке программирования Python становится всё более популярным среди учёных и инженеров благодаря своей гибкости, простоте и эффективности. В данном докладе мы рассмотрим основные преимущества использования Python для моделирования электродвигателей и его роль в современных научных исследованиях [2]. В данной работе модель будет строиться в АВС координатах, а не в более классическом виде

в dq координатах. Этот подход позволит в дальнейшем использовать данную модель при моделировании векторного управления асинхронным двигателем.

Рассмотрим общий подход к созданию такой модели:

Необходимо описать системой дифференциальных уравнений работу асинхронного электродвигателя в abc-координатах. Эти уравнения включают в себя токи, напряжения, потокосцепления и другие параметры двигателя. Систему уравнений будем использовать в нормальной форме Коши приведенную к матричному виду [3, 4].

Необходимо выбрать метод решения уравнений. Для решения системы дифференциальных уравнений можно использовать различные методы, такие как метод Эйлера, метод Рунге–Кутты, метод конечных разностей и др. Воспользуемся методом расчета Рунге–Кутты 4 порядка.

Создайте программу на Python. Используйте выбранный метод для решения уравнений асинхронного двигателя. Программа должна включать в себя определение начальных условий, расчёт токов, напряжений и других параметров двигателя, а также вывод результатов.

Добавьте дополнительные функции. В зависимости от ваших целей, вы можете добавить в программу дополнительные функции, такие как моделирование различных режимов работы двигателя, анализ его характеристик и т. д.

Протестируйте программу. Перед использованием программы для моделирования асинхронного двигателя необходимо протестировать её на простых примерах, чтобы убедиться в правильности работы программы. Необходимо подтвердить адекватность полученной модели.

Используйте программу для моделирования. После тестирования программы вы можете использовать её для моделирования различных режимов работы асинхронного двигателя и анализа его характеристик.

Для моделирования воспользуемся параметрами трехфазного асинхронного двигателя АИР63А2 [3]. Примерный код для математической модели асинхронного двигателя может выглядеть следующим образом:

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
Pn = 0.37 # Номинальная мощность, кВт
Un = 380 # Номинальное напряжение, В
f = 50 # Частота сети, Гц
p = 2 # Число пар полюсов
omega_syn = 2 * np.pi * f / p # Синхронная угловая скорость, рад/с
R_s = 23.9232 # Сопротивление статора, Ом
R_r = 21.1188 # Сопротивление ротора, Ом
L_s = 0.9721 # Индуктивность статора, Гн
L_R = 0.9898 # Индуктивность ротора, Гн
L_m = 0.9107 # Взаимная индуктивность, Гн
p = 1 # Число пар полюсов
J_total = 0.00069 # Суммарный момент инерции, кг·м²
omega_nominal = 1500 * (2 * np.pi / 60) # Номинальная механическая скорость, рад/с
U_phase = Un / np.sqrt(3) # Фазное напряжение, В
U_1A = U_phase
U_1B = U_phase
U_1C = U_phase
M_c = 1
L1 = np.array([
    [L_s, -L_m/2, -L_m/2, L_m, -L_m/2, -L_m/2],
    [-L_m/2, L_s, -L_m/2, -L_m/2, L_m, -L_m/2],
    [-L_m/2, -L_m/2, L_s, -L_m/2, -L_m/2, L_m],
    [L_m, -L_m/2, -L_m/2, L_R, -L_m/2, -L_m/2],
    [-L_m/2, L_m, -L_m/2, -L_m/2, L_R, -L_m/2],
    [-L_m/2, -L_m/2, L_m, -L_m/2, -L_m/2, L_R]])
L2 = np.zeros((6, 6))
R = np.diag([R_s, R_s, R_s, R_r, R_r, R_r])
```

```
def motor_dynamics(t, y):
    i_1A, i_1B, i_1C, i_1a, i_1b, i_1c, omega = y
    i1 = np.array([i_1A, i_1B, i_1C, i_1a, i_1b, i_1c])
    U_1 = np.array([
        U_1A * np.sin(2 * np.pi * f * t),
        U_1B * np.sin(2 * np.pi * f * t - 2 * np.pi / 3),
        U_1C * np.sin(2 * np.pi * f * t + 2 * np.pi / 3),
        U_1a,
        U_1b,
        U_1c ])
    L1_inv = np.linalg.inv(L1)
    di1_dt = L1_inv @ (U_1 - (R + L2 * (p * omega) / np.sqrt(3)) @ i1)
    M_1 = p * (np.sqrt(3)/2) * L_m * (
        (i_1A * i_1c + i_1B * i_1a + i_1C * i_1b) -
        (i_1A * i_1b + i_1B * i_1c + i_1C * i_1a) )
    domega_dt = (p / J_total) * (M_1 - M_c)
    return np.concatenate((di1_dt, [domega_dt]))
i_initial = np.zeros(6)
omega_initial = 0
y0 = np.concatenate((i_initial, [omega_initial]))
t_span = (0, 1)
t_eval = np.linspace(t_span[0], t_span[1], 1000)
sol = solve_ivp(motor_dynamics, t_span, y0, t_eval=t_eval, method='RK45')
i_1A_sol = sol.y[0]
i_1B_sol = sol.y[1]
i_1C_sol = sol.y[2]
i_1a_sol = sol.y[3]
i_1b_sol = sol.y[4]
i_1c_sol = sol.y[5]
M_1 = p * (np.sqrt(3)/2) * L_m * (
    (i_1A_sol * i_1c_sol + i_1B_sol * i_1a_sol + i_1C_sol * i_1b_sol) -
    (i_1A_sol * i_1b_sol + i_1B_sol * i_1c_sol + i_1C_sol * i_1a_sol))
```

Далее прописываются команды построения графиков для вывода на экран. Мы выведем зависимости токов фаз статора от времени (рис. 1), угловой скорости ротора от времени (рис. 2) и электромагнитного момента от времени (рис. 3).

Этот код представляет собой упрощённую модель асинхронного двигателя, которая может быть использована для изучения его работы. Вы можете изменить параметры двигателя и методы решения уравнений в соответствии с вашими целями.

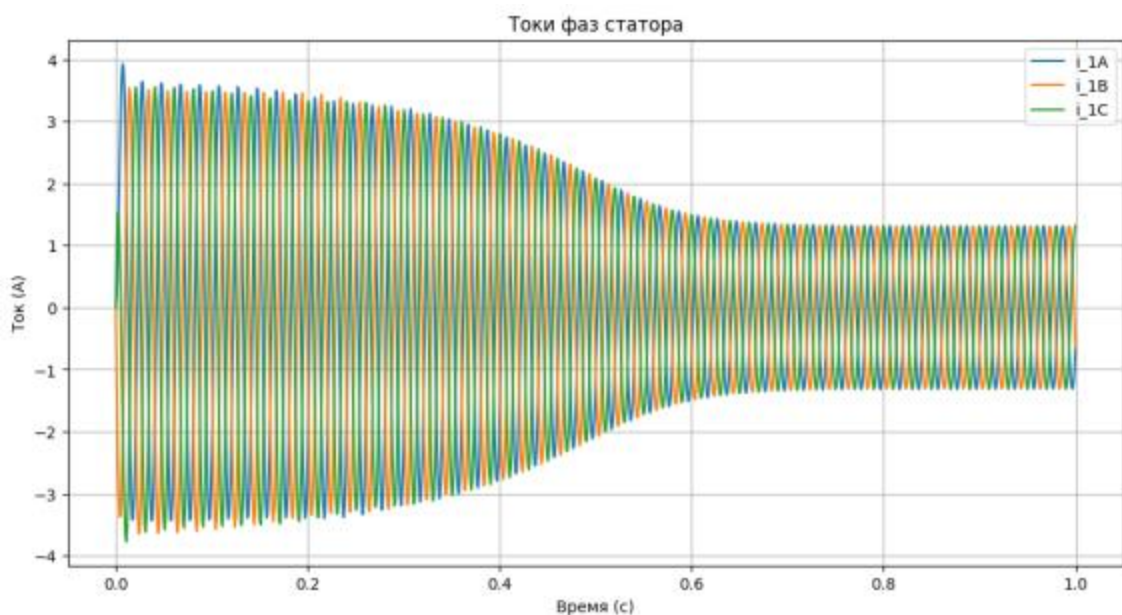


Рис. 1. Зависимость токов фаз статора асинхронного двигателя от времени

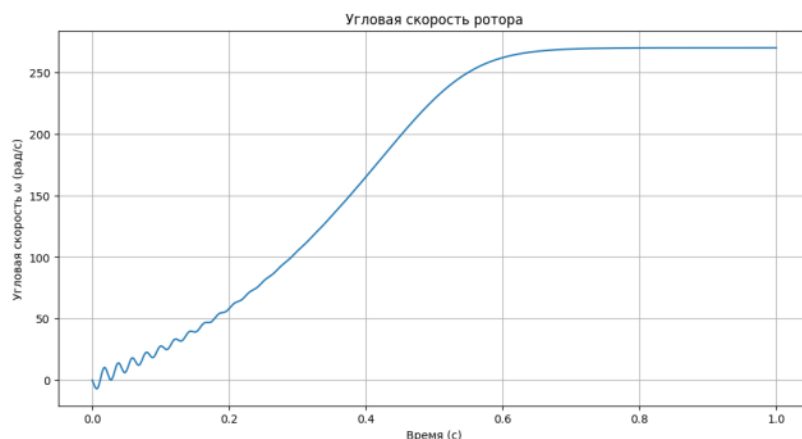


Рис. 2. Зависимость угловой скорости ротора асинхронного двигателя от времени

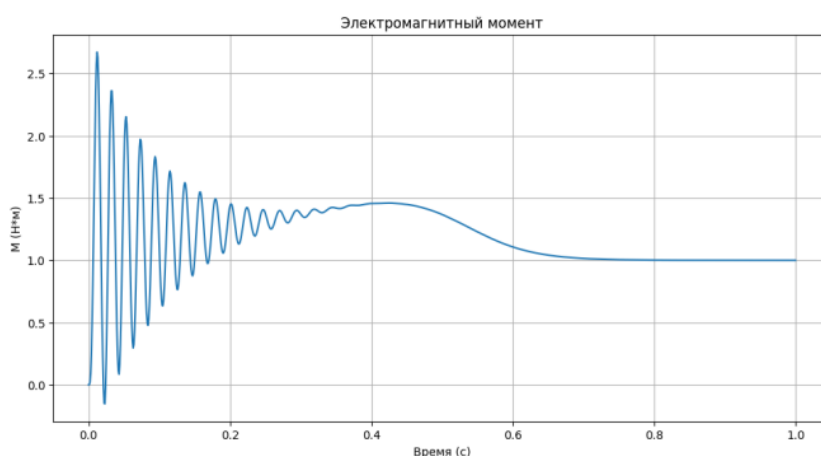


Рис. 3. Зависимость электромагнитного момента асинхронного двигателя от времени

Тем самым можно сказать, что моделирование асинхронного двигателя на Python в ABC-координатах является перспективным и удобным решением, которое имеет преимущества перед моделированием в других программах и языках программирования. Благодаря большому сообществу разработчиков, множеству библиотек и фреймворков, гибкости и удобству в работе с данными, а также возможности получения более точных и надёжных результатов. Кроме того, Python является бесплатным и открытым исходным кодом, что делает его доступным для широкого круга пользователей. Python становится мощным инструментом для проектирования и оптимизации асинхронных двигателей. Всё это делает его привлекательным выбором для широкого круга пользователей, работающих в области электротехники, машиностроения, энергетики и других смежных областях.

СПИСОК ЛИТЕРАТУРЫ

1. Буйначев С.К., Боклаг Н.Ю. Основы программирования на языке : учебное пособие. – Электрон. текстовые данные. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2014. – 92 с.
2. Однокопылов Г.И., Розаев И.А., Брагин А.Д. и др. Сравнительный анализ отказоустойчивого управления асинхронного и вентильно-индукторного электропривода // Интеллектуальные энергосистемы: труды IV Международного молодёжного форума, 10–14 октября 2016 г., Томск: в 3 т. Т. 2. – Томск: Изд-во ТПУ, 2016. – С. 89–93. Дата обращения: 15.11.2024.
3. Однокопылов Г.И., Брагин А.Д. Математическая модель асинхронного двигателя в неполнофазном режиме работы // Известия Томского политехнического университета. – 2013. – Т. 323, № 4. – С. 133–137.
4. Однокопылов Г.И. Методы и алгоритмы отказоустойчивого управления электроприводами опасных производственных объектов: диссертация на соискание ученой степени доктора технических наук: спец. 05.09.03 / науч. конс. В.Г. Букреев. – Томск: 2017.