больше перспектив, так как путем анализа мощности вторичных гармоник или амплитуды четных гармоник можно подбирать скважность меандров для из минимизации, что может служить параметром оценки степени нелинейности динамики.

Проведенные исследования показали, что использование предварительной обработки измерительных данных с объекта идентификации с целью исключения долговременных трендов позволяет повысить робасность и адекватность получаемых коэффициентов. Метод ММИ более устойчив к адекватности оценки характеристик по сравнению с ГИ.

ЛИТЕРАТУРА

1. Corriou JP. Process control – theory and applications. 2nd ed. — Springer, 2017. 866 p.

2. Nelles O. Nonlinear system identification: from classical approaches to neural networks. Springer Verlag Publ., 2001. 785 p.

3. Олифирович Н. М., Гринюк Д. А., Оробей И. О. Гармоническая идентификация технологических объектов в реальном времени // Труды БГТУ. 2016. Сер. 3, Физ.-мат. науки и инфор. № 6, С. 117–121.

4. Oliferovich N., Hryniuk D., Orobei I. Harmonic identification of technological objects in real time // Open Conference of Electrical, Electronic and Information Sciences (eStream). Vilnius, , 2016. P. 1–4.

5. Гринюк Д. А., Олиферович Н. М, Сухорукова И. Г., Оробей И. О. Идентификация параметров динамических каналов воздушного теплообменника // Труды БГТУ. Сер. 3, Физ.-мат. науки и инфор. 2022. № 2 (260). С. 70–79.

УДК 004.431.2

Дубиковская Е.В., Гринюк Д.А.,
Чепурко М.В., Арпентий Д.О.
(Белорусский государственный технологический университет)

ИНСТРУМЕНТЫ РАСШИРЕНИЯ ВОЗМОЖНОСТЕЙ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ

Python – мощный язык программирования общего назначения, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных, машинном обучении и естественных науках. Он был реализован Гвидо ван Россумом в 1989 году. Python классифицируется как интерпретируемый язык программирования, автоматизирующий большинство фундаментальных операций (таких как управление памятью),

выполняемых на уровне процессора («машинный код»). Python считается языком более высокого уровня, чем, например, C++, из-за его выразительного синтаксиса (который во многих случаях близок к естественному языку) и богатого разнообразия встроенных структур данных, таких как списки, кортежи, множества и словари.

Философия Python основана на открытом исходном коде и развивающемся сообществе. Его синтаксис прост и удобен для изучения, а написанный код работает на любой платформе (ПК, серверы, кластеры, мобильные телефоны и даже устройства IoT). Среди доступных библиотек есть множество пакетов практически для любых задач, причем эти пакеты часто можно скачать бесплатно (как и сам Python). Все это увеличивает скорость развития и популярность этого языка.

Однако интенсивность разработки Python имеет и обратную сторону. Постоянное развитие создало проблемы с совместимостью версий. Python, будучи интерпретируемым языком высокого уровня, не может быть таким же быстрым и эффективным, как C++. В научных задачах это нивелируется тем, что некоторые библиотеки, используемые Python, написаны на C++ и предварительно скомпилированы. В данном случае Python выступает в роли скриптового языка, организующего эффективную работу кода, скомпилированного на C++.

С другой шло развитие микропроцессорной техники управления. Прямая концепция, которая заложена в МЭК 61131-3 уже стала тесной для решения задач управления технологическими процессами. Концепция разработки современных ПЛК, основной лошадкой реализации алгоритмов управления также поменялась. Если раньше это были закрытые от пользователей системы, то сейчас там доминирует Linux.

Программируемые логические контроллеры (ПЛК) и Python – отличные инструменты для инженеров по автоматизации. ПЛК – это тип компьютера, который обычно используется в промышленных системах управления и других приложениях, требующих возможности цифрового управления электрооборудованием. ПЛК спроектированы так, чтобы быть прочными и надежными, и они программируются с использованием специализированного программного обеспечения для выполнения широкого спектра функций управления.

Совместное использование ПЛК и Python – это рай для автоматизации. У этого подхода очень много преимуществ. Во-первых, ПЛК чрезвычайно надежны и могут выдерживать суровые промышленные условия. В совокупности это позволяет нам использовать возможности Python для разработки более совершенных систем управления. Это также позволяет нам легко интегрироваться с другими системами, такими как базы данных и веб-серверы, используя Python. Добавление

Python в ваш набор навыков очень важно, потому что оно резко увеличивает ваш потенциал трудоустройства и ставит вас на более высокий уровень, чем ваши конкуренты. Python также является наиболее используемым языком программирования и используется такими крупными компаниями, как НАСА, Lockheed Martin и Университет Джона Хопкинса.

Существует также огромное количество библиотек и инструментов, доступных для использования с Python благодаря его большому и активному сообществу. Это также интерпретирующий язык, то есть его не нужно компилировать перед запуском. Это упрощает тестирование и модификацию, экономя время и снижая риск ошибок. Его также можно использовать для задач обработки высокого уровня, таких как анализ данных, визуализация и машинное обучение данных, генерируемых ПЛК.

С другой стороны, Python не всегда является наиболее подходящим языком для всех задач, связанных с ПЛК. Например, интеграция Python со специализированными аппаратными или программными системами может оказаться сложнее, чем просто использование другого языка. Также очень сложно найти программистов Python, имеющих опыт программирования ПЛК, поскольку такое сочетание навыков встречается редко.

Как правило, лучше всего использовать Python с ПЛК, когда вам необходимо выполнять задачи обработки высокого уровня, такие как анализ данных, визуализация и машинное обучение данных, сгенерированных ПЛК. Чтобы использовать Python с ПЛК, вы можете использовать библиотеку или модуль, специально разработанный для связи с ПЛК, например pymodbus или python-snap7, использовать библиотеку последовательной связи общего назначения, например pySerial, для отправки и получения данных в и из ПЛК через последовательное соединение или используйте протокол удаленной связи, например Modbus или Ethernet/IP, для связи с ПЛК через сетевое соединение. Python часто является отличным выбором, но в некоторых обстоятельствах он не лучший. В некоторых случаях более подходящим может оказаться язык программирования, который более тесно связан с требованиями к оборудованию и реальному времени ПЛК, например, релейная логика или структурированный текст. Прежде чем начинать проект, обязательно оцените все варианты языка, но не сбрасывайте со счетов Python!

Если вы решили использовать Python для связи с ПЛК и управления им, вы можете использовать библиотеку Python под названием pymodbus. Pymodbus – это полная реализация протокола Modbus, использующая в качестве ядра асинхронной связи Twist. Pymodbus включает в себя полнофункциональный сервер Modbus и клиентскую

библиотеку для протоколов Modbus RTU и ASCII. Чтобы использовать pymodbus, вам необходимо установить библиотеку, а затем использовать ее для установки соединения Modbus TCP с ПЛК. Как только соединение установлено, вы можете использовать функции pymodbus для чтения и записи данных в регистры и обмотки ПЛК.

Сейчас мы наблюдаем стечение факторов, которые привлекают внимание к Python. Во-первых, Индустрия 4.0 меняет парадигму того, как мы думаем о промышленной автоматизации; а именно, акцент на «умное» оборудование с улучшенной автономностью, богатую среду больших данных и полную интеграцию с технологиями следующего поколения, такими как аддитивное производство и облачные вычисления.

Еще одним последствием Индустрии 4.0 является промышленный Интернет вещей (IIoT), который объединяет промышленное оборудование в локальную сеть для межмашинной связи в реальном времени (M2M) и обеспечивает постоянный поток данных с датчиков для аналитики. В результате мы являемся свидетелями конвергенции ИТ и ОТ, разрушения разрозненности, которая долгое время отделяла профессионалов в области информационных технологий от их коллег из операционных технологий. В конце концов, устройства IIoT используют информацию для оптимизации своей работы.

Это возвращает нас к Python. Когда мы смотрим на сильные стороны самого популярного в мире языка программирования, мы видим некоторые явные преимущества IIoT. Прежде всего, Python известен своей способностью обрабатывать огромные наборы данных. Во-вторых, Гвидо Ван Россум, изобретатель Python, разработал его так, чтобы обеспечить высокую читабельность – ключевую особенность, когда несколько инженеров будут работать над одним и тем же кодом или поддерживать его, а также функцию, которая разжигает огонь инновационных итераций. И, наконец, Python имеет открытый исходный код, имеет замечательное сообщество и является идеальным выбором для многих наиболее привлекательных современных приложений.

Без сомнения, наибольшее влияние Python оказывает в сфере машинного обучения (ML), отрасли искусственного интеллекта (ИИ), где алгоритмы учатся на данных, и никто явно не кодирует какие-либо правила. Общие промышленные применения включают профилактическое обслуживание и автономную робототехнику. Большая часть современного машинного обучения написана на Python. Такие фреймворки, как PyTorch и TensorFlow с открытым исходным кодом от Google, используют Python. AWS SageMaker, облачный сервис искусственного интеллекта Amazon, поставляется со встроенным комплектом разработки программного обеспечения Python (SDK).

Запуская либо периферийные вычисления путем внедрения графических процессоров в сами производственные устройства, либо используя ресурсы локального шлюза IIoT для туманных вычислений, мы можем использовать нашу обученную модель на месте. [1].

Компьютерное зрение. Чтобы роботизированная рука что-то взяла, ей сначала нужно знать, на что она смотрит. Введите компьютерное зрение (CV), область искусственного интеллекта, которая позволяет машинам использовать свои камеры в качестве глаз и, что еще более важно, распознавать объекты, которые они видят.

Ряд авторов рассматривают Python как альтернативу Matlab [2-4] и видят в этом большие преимущества.

В итоге умение использовать Python для решения задач автоматического управления является неоспоримым преимуществом инженера по автоматизации.

ЛИТЕРАТУРА

1. Jivan S. Parab, Madhusudan Ganuji Lanjewar, Marlon Darius Sequeira, Gourish Naik, Arman Yusuf Shaikh. Python Programming Recipes for IoT Applications, – Springer Singapore 2013 192 p.

2. Kumar, R.Mathusoothana & Lakshmi, S.L. & K.V., Shiny & P., Venkadesh. (2023). Problem Solving and Python Programming. 10.59646/pythonprog/049.

3. Raja, K. (2023). Python-based fuzzy logic in automatic washer control system. Soft Computing. 27. 1-27. 10.1007/s00500-023-07979-3.

4. Shaw, Rabindra. (2021). Innovations in Electrical and Electronic Engineering. 10.1007/978-981-16-0749-3.

УДК 681.5

Барашко О.Г., Касперович А.В.
(Белорусский государственный технологический университет)

ВИЗУАЛИЗАЦИЯ ОБЩЕЙ ЭФФЕКТИВНОСТИ ОБОРУДОВАНИЯ ПРИ АГРЕГИРОВАНИИ ИНФОРМАЦИИ

Оснащение производства современными системами управления приводит к оцифровке всех получаемых данных, создающую у персонала предприятия иллюзию их доступности. Но оцифровано не значит доступно. Данные есть, их, много, но нет средств, которые могут их агрегировать и увязать между собой в единое информационное пространство для всех уровней управления [1, 2]. Например, в случае