

УДК 62-529

UDC 62-529

**РАЗРАБОТКА МЕТОДИКИ  
АВТОМАТНОГО  
ПРОГРАММИРОВАНИЯ ДЛЯ  
УПРАВЛЕНИЯ АСИНХРОННЫМ  
ЭЛЕКТРОДВИГАТЕЛЕМ**

**DEVELOPMENT OF A METHOD OF  
AUTOMATA-BASED PROGRAMMING  
FOR ASYNCHRONOUS MOTOR  
CONTROL**

*Мартынов А. Н., Аль-Тибби В. Х.,  
Чернышев И. Р.*

*Martynov A. N., Al-Tibbi V. H.,  
Chernyshev I. R.*

Донской государственный технический  
университет, Ростов-на-Дону, Российская  
Федерация

Don State Technical University, Rostov-on-Don,  
Russian Federation

[alexey.martynov.1997@mail.ru](mailto:alexey.martynov.1997@mail.ru)

[alexey.martynov.1997@mail.ru](mailto:alexey.martynov.1997@mail.ru)

[nb1979@mail.ru](mailto:nb1979@mail.ru)

[nb1979@mail.ru](mailto:nb1979@mail.ru)

[igorchernysh007@yandex.ru](mailto:igorchernysh007@yandex.ru)

[igorchernysh007@yandex.ru](mailto:igorchernysh007@yandex.ru)

Основной целью настоящей работы является разработка методики автоматного программирования для управления асинхронным электродвигателем. В ходе исследования разработан подход по унификации разработки системы управления приводом. Проведено математическое моделирование системы управления сверлильным станком в пакете прикладных программ с использованием предлагаемого подхода. В результате проведенной работы получен программный код на стандартах МЭК 61131.

The main aim of this work is to develop a method of automata-based programming to control asynchronous motor. The study has developed an approach to unify the development of the actuator control system. The authors have conducted the mathematical modeling of control system of drilling machine in the application package using the proposed approach. As a result the authors have received the software code according to standards IEC 61131.

**Ключевые слова:** частотный и векторный метод управления, автоматное программирование, генерация, отладочные протоколы, графические диаграммы, Matlab.

**Keywords:** frequency and vector management method; automata-based programming; generation; debug protocols; graphical charts; Matlab.

**Введение.** В настоящее время колоссальное внимание уделяется созданию технологий программирования для систем реального времени и встроенных систем, к которым предъявляются особые требования по кондиции программного обеспечения. Наиболее известный подход в этом направлении — синхронное программирование [1].

С началом совершенствования в Европе синхронного программирования, в России разрабатывается «автоматное программирование», которое можно расценивать как разновидность синхронного программирования.

**Постановка задачи.** Данная работа посвящена наращиванию эффективности итогов проектирования автоматизированных систем с использованием асинхронных двигателей за счет применения новой методики автоматного программирования.

Для достижения выше сформулированной цели работы надобно решить нижеследующие задачи:

- провести исследование способов управления асинхронными двигателями;

- выбрать способ управления;
- провести анализ имеющихся методов программирования алгоритмов управления асинхронным двигателем;
- выбрать самый уместный пакет прикладных программ для достижения установленной цели;
- разработать методику создания программы управления АД.

**Анализ состояния вопроса.** Асинхронные двигатели достаточно распространены, надежны, имеют условно низкую цену, прекрасные эксплуатационные качества.

Благодаря резкому развитию электроники и появлению дешевых устройств, для корректировки скорости вращения асинхронных двигателей применяются устройства, которые меняют частоту подводимого напряжения — полупроводниковые преобразователи частоты.

Для реализации предполагается использовать ПЧВ. В ПЧВ (преобразователь частоты векторный) применяются частотный и векторный метод управления.

Особо точное и высокоэффективное управление предоставляет режим векторного управления. При таком режиме система управления с высокой точностью фиксирует значения выходных токов и напряжений, обеспечивает вычисление характеристик двигателя, моделирует тепловые характеристики двигателя, согласует параметры двигателя с разными режимами его работы, совершает огромные объемы вычислений с весьма высокой скоростью [2,3].

Векторное управление в сопоставлении с частотным пользуется более высокой производительностью и избавляет нас от целого ряда недостатков скалярного управления. Смысл векторного управления содержится в том, чтобы контролировать не только величину и частоту напряжения питания, но и фазу. Иначе говоря, ведётся контроль над величиной и углом пространственного вектора. Такое управление дает возможность самостоятельно и практически мгновенно корректировать темп вращения и момент на валу.

Приоритеты векторного управления:

- высокая и точная регулировка скорости;
- плавный старт и вращение двигателя на любой частоте;
- при изменении нагрузки почти что не происходит перепада скорости;
- увеличенный диапазон управления и четкость регулирования;
- уменьшаются потери на намагничивание и нагрев, растёт КПД двигателя.

Недостатки векторного управления:

- необходимо задавать параметры двигателя;
- при постоянной нагрузке наблюдаются высокие колебания скорости;
- огромная вычислительная сложность.

**Описание исследования.** В качестве модельного примера в работе предлагается использовать систему управления приводом главного движения и приводом подачи вертикально-сверлильного станка. Структурное описание системы управления представлено на рис. 1

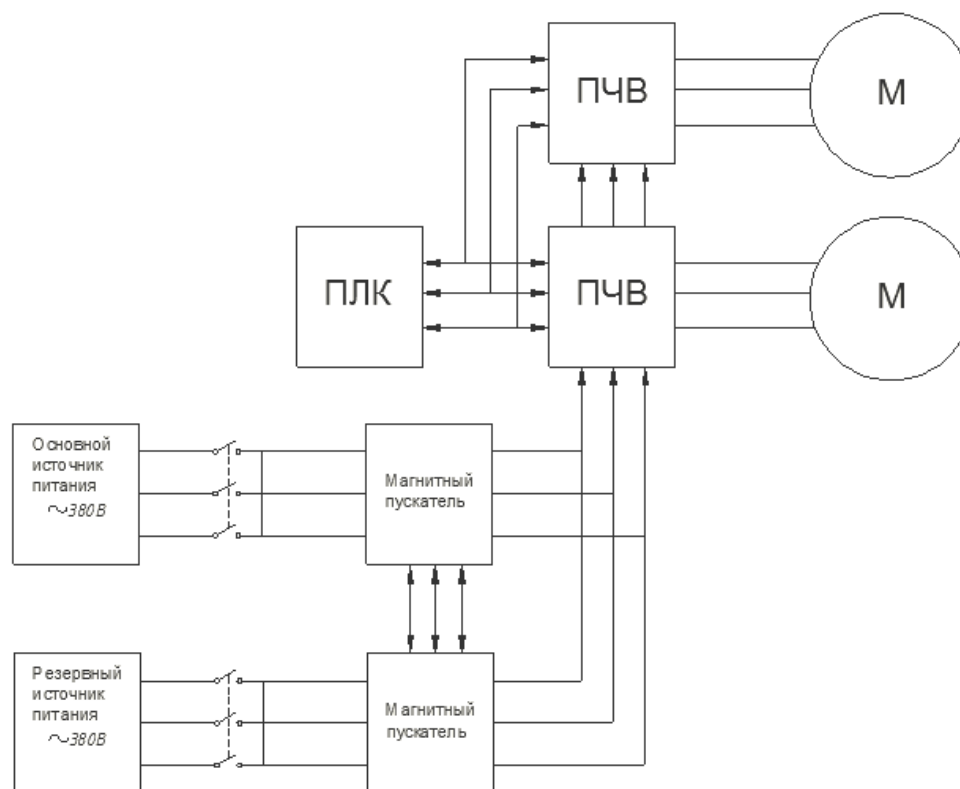


Рис. 1. Структурная схема управления асинхронным электроприводом

Исследования в области автоматного программирования относятся к программной инженерии и к программной кибернетике, а также опираются на идеи теории автоматов и теории автоматического управления. Автоматное программирование или же «программирование от состояний» — это способ разработки программного обеспечения, сделанный на расширенной модели конечных автоматов и нацеленный на создание обширного класса приложений. Наперекор общественному мнению, речь идет не только о построении и реализации конечных автоматов для применения в программах, но и о способе формирования программ в целом, поведение которых описывается автоматами.

Достоинства:

1. Генерация. Предполагается, что создание (генерация) программы начинается только лишь уже после ее проектирования. При этом в инженерной практике (в отличие от классического программирования) проект или его этап всенепременно заканчивается выпуском плановой документации.

Проектировка автоматов, обрисовывающих логику программ, при традиционном программировании никак не упорядочена и по этой причине трудна, а также формальный и изоморфный переход от автоматов к реализующим их программам приводит к тому, что программы мгновенно функционируют или им нужна минимальная отладка.

2. Отладочные протоколы. Если необходимо провести отладку для автоматных кодов могут формироваться отладочные протоколы, которые отображают поведение программ в терминах автоматов (переходов, состояний, значений входных и выходных воздействий).

3. Графические диаграммы. Увеличение периода формирования программ при использовании автоматного подхода восполняется уменьшением периода их отладки. Это приводит к тому, что для программ средней сложности разработки на принципе автоматного и традиционных подходов почти совпадают. Тем не менее, в первом варианте остаются диаграммы,

общепонятные человеку, по которым программа была создана формально, а во втором — лишь программа, восприятие логики которой для заказчика или даже для её автора зачастую является большой проблемой.

Разработка программ со сложным поведением без использования диаграмм приводит к затруднениям на всех стадиях жизненного цикла. В особенности непросто ввести в эксплуатацию программы, так как всю специфику их поведения приходится держать в уме на протяжении всего времени написания, вместо того чтобы воспроизвести их на диаграмме и на время вычеркнуть из памяти.

Для реализации методики был выбран пакет прикладных программ Matlab, который интегрирован со средой графического моделирования Simulink, а также средством создания графов состояний и переходов Stateflow. Данная среда в полной мере отвечает задачам создания моделей объектов управления, а также моделирования их систем управления используя метод автоматного программирования. В 2011 году в среду Matlab был внесен компилятор PLC Coder, который предоставляет возможность переводить модели систем управления написанных с помощью автоматного программирования в языки стандарта МЭК 61131, применяющиеся в большей части программируемых логических контроллеров [4,5].

Проанализировав состояние вопроса создания управляющих программ для АД, предложен разрабатываемый подход, который заключается в следующем:

1. Формирование схемы взаимосвязей блока управления с объектом управления и системой верхней ступени.
2. Создание списка и описания входных и выходных переменных.
3. Принятие алгоритма работы от Клиента (в виде устного описания и временных диаграмм).
4. Приближенное создание системы графов переходов конечных автоматов.
5. Отображение графов переходов с применением пакета Stateflow.
6. Разработка модели ОУ системой MATLAB – Simulink.
7. Единое моделирование системы автоматов и ОУ, с получением временных диаграмм работы.
8. Сверка результатов моделирования с алгоритмом работы. Если результаты никак не удовлетворяют требования Заказчика, в таком случае следует возвратиться к этапу проектирования графов переходов.
9. Исполнение системы графов переходов, изображенных в Stateflow на языке описания аппаратуры (МЭК 61131), с поддержкой интерпретатора пакета PLC Coder.
10. Компиляция и последующая загрузка в целевую аппаратуру.

Полученные результаты моделирования представлены программой, созданной в среде MatLab Simulink. Пример реализации программного блока управления двигателем главного движения представлен на рис 2,3 [7,8].

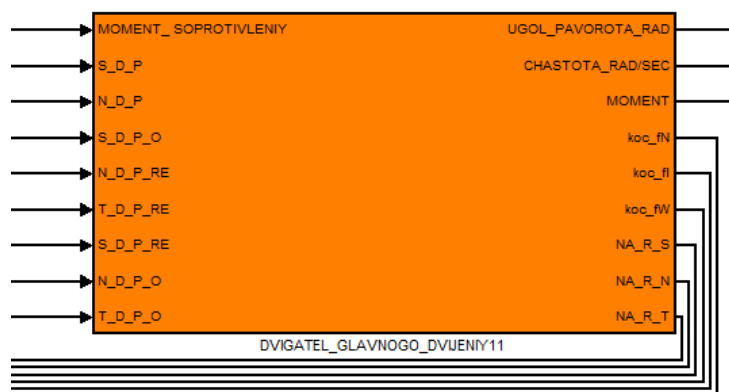


Рис. 2. Вид блока, моделирующий двигатель главного движения

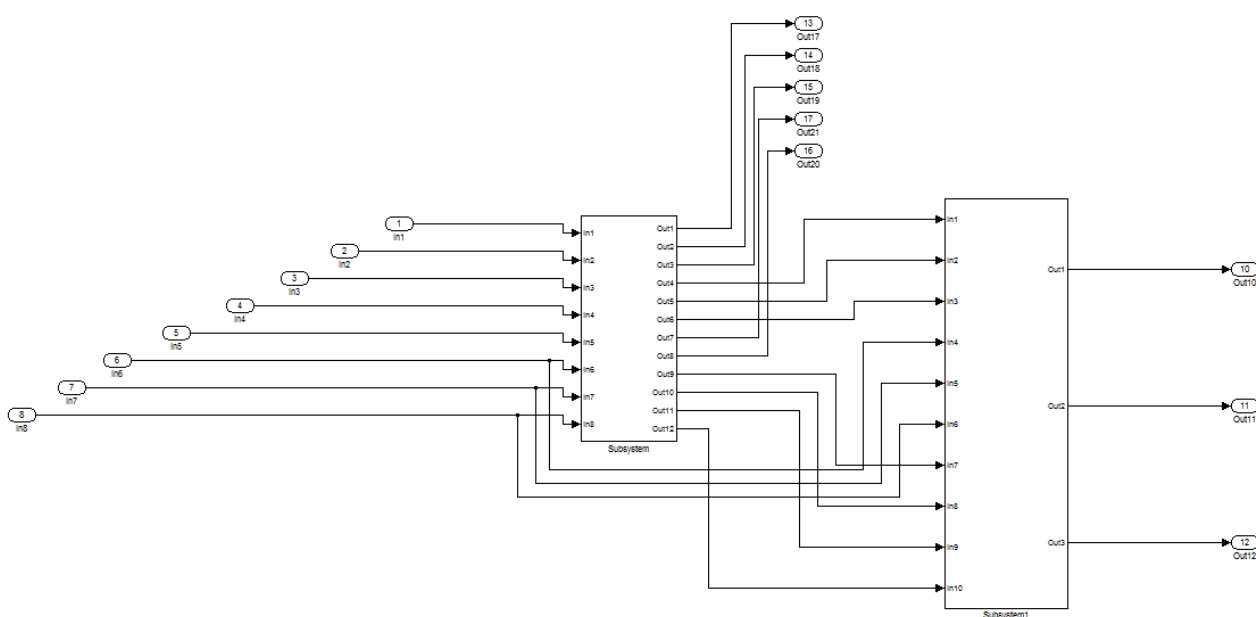


Рис. 3. Подсистема двигателя главного движения.

**Заключение.** Научная новизна работы заключается в представленной новой методике автоматного программирования, которую допускается использовать в большинстве систем автоматического управления АД. Предложенная методика позволяет значительно сжать кропотливость процесса разработки ПО для системы управления АД, а также по максимуму стандартизировать этот процесс [9]. Исследована возможность перехода от математической модели АСУ, созданной в пакете прикладных программ, позволяющая удовлетворить все требования по критериям качества с применением программного кода на языках стандарта МЭК 61131(с возможностью загрузки в реальный ПЛК).

#### Библиографический список.

1. Vyatkin, V. Distributed Execution and Cyber-Physical Design of Baggage Handling Automation with IEC 61499 [Электронный ресурс] / V. Vyatkin // IEEE Conference on Industrial Informatics, Lisbon, Portugal, 2011. — Режим доступа : [http://www.vyatkin.org/publ/INDIN11\\_YV.pdf](http://www.vyatkin.org/publ/INDIN11_YV.pdf) (дата обращения: 16.01.2018).
2. Christensen, J. H. Design patterns for system engineering with IEC 61499 / J. H. Christensen // in Conference Verteilte Automatisierung, Magdeburg, Germany, 2000. — pp. 63–71.

3. P. Valckenaers and H. Van Brussel, Holonic Manufacturing Execution Systems — CIRP Annals — 2005. — pp. 427-432.
4. Christensen, J. HMS/FB Architecture and its Implementation // in Agent Based Manufacturing. Advances in the Holonic Approach, M. Deen, Ed., ed: Springer, —2003.
5. HOLOBLOC Function Block Development Kit, ed, 2008. — Режим доступа: <http://www.holobloc.com> (дата обращения: 16.01.2018).
6. Портал искусственного интеллекта / Многоагентные системы. — Режим доступа : <http://www.aiportal.ru/articles/multiagent-systems/multiagent-systems.html> (дата обращения: 16.01.2018).
7. Передовые технологии и технические решения /Обзор стандарта IEC 61499. — Режим доступа : <http://www.picad.com.ua/0405/pdf/8-12.pdf> (дата обращения: 16.01.2018).
8. Национальная библиотека им. Н. Э. Баумана Bauman National Library/IEC 61131-3. — Режим доступа : [http://ru.bmstu.wiki/IEC\\_61131-3](http://ru.bmstu.wiki/IEC_61131-3) (дата обращения: 16.01.2018).
9. Википедия. Свободная энциклопедия/Промышленная сеть. — Режим доступа : [https://ru.wikipedia.org/wiki/Промышленная\\_сеть](https://ru.wikipedia.org/wiki/Промышленная_сеть) (дата обращения: 16.01.2018).