

УДК 621.3:62.5:004.413.2

DOI: 10.24160/1993-6982-2020-3-65-74

Модельно-ориентированная разработка системы управления электропривода и её исследование

И.С. Полющенко

Представлены материалы по разработке микропроцессорной системы управления электропривода на базе электрических двигателей разных типов. Разработка направлена на достижение характеристик электропривода и функциональности его управления, соответствующих аналогам. В работе использованы средства модельно-ориентированного программирования. Методика их применения заключается в автоматическом генерировании программного обеспечения из математических моделей, в которых предельно учтена микропроцессорная реализация вычислительных, измерительных и управляющих процессов.

Анализ автоматически сгенерированного программного обеспечения и результаты экспериментальных исследований электропривода на разных этапах его создания показали, что стандартные средства модельно-ориентированного программирования не позволяют в полной мере решить все задачи и в полной мере достигнуть характеристик аналогов. Выявлены особенности автоматически сгенерированного программного обеспечения, приводящие к некорректной работе и снижению рабочих характеристик электропривода. Стандартные средства не позволили разработать программное обеспечение с требуемой степенью детализации. Для устранения указанных недостатков стандартные средства были дополнены пользовательским программным обеспечением, что не противоречит концепции модельно-ориентированного программирования, так как структура программного обеспечения разработана на основе стандартных модельных элементов.

Приведены наиболее яркие примеры доработки или замены стандартных модельных блоков и подсистем на их основе в случае их некорректной работы, громоздкости и недостаточной выразительности. В первую очередь это касается критичных по времени выполнения подсистем программного обеспечения. Устранены коллизии между подсистемами программного обеспечения и достигнуто своевременное обновление информации при их взаимодействии. Применение пользовательского программного обеспечения способствовало рациональному распределению ресурсов микроконтроллера. Некоторые элементы созданы в результате доработки текста автоматически сгенерированных подпрограмм. Экспериментально опробованы и применены различные решения, улучшающие характеристики электропривода и расширяющие функционал его управления. Исследования электропривода показали его примерное соответствие аналогам по функциональности и рабочим характеристикам.

Ключевые слова: система управления электропривода, модельно-ориентированное программирование, микропроцессорное управление, регулирование координат, электрический двигатель.

Для цитирования: Полющенко И.С. Модельно-ориентированная разработка системы управления электропривода и её исследование // Вестник МЭИ. 2020. № 3. С. 65—74. DOI: 10.24160/1993-6982-2020-3-65-74.

Model-Oriented Designing of an Electric Drive Control System and Its Investigation

I.S. Polyushchenkov

Materials on the development of a microprocessor control system for an electric drive based on various types of electric motors are presented. The development is aimed at achieving the electric drive characteristics and its control functionality similar to those in the existing analogs. The development was carried out using model-oriented programming tools. The methodology of their application involves automatic generation of the software from mathematical models, in which the microprocessor implementation of computational, measurement, and control processes is taken into account in advance.

The automatically generated software and the results of electric drive experimental studies were analyzed at different stages of its development, and it has been shown from that analysis that the standard model-oriented programming tools do not make it possible to fully solve all the problems, nor do they allow the characteristics of the analogs to be fully achieved. The features of the automatically generated software that lead to incorrect operation and degradation of the electric drive operating characteristics have been identified. The standard tools did not allow the software to be developed with the required degree of detail. To overcome these shortcomings of the standard development tools, they were supplemented with user software, an approach that does not contradict the model-oriented programming concept, since the software structure has been developed on the basis of standard model elements.

The most striking examples of refining or replacing the standard model units and subsystems based on them in case of their incorrect operation, bulkiness and insufficient expressiveness are given. This applies primarily to software subsystems critical in terms of their execution time. Collisions between the software subsystems have been eliminated, and timely updating of information during their interaction has been achieved. Owing to application of the user software, more rational distribution of the microcontroller resources has been achieved. Some of its elements were developed as a result of refining the text of the automatically generated routines. Various solutions improving the electric drive characteristics and expanding its control functionality have been experimentally tested and applied. The electric drive investigation results have shown that it is approximately consistent with the analogs in terms of functionality and performance.

Key words: electric drive control system, model-oriented programming, microprocessor control, coordinate control, electric motor.

For citation: Polyushchenkov I.S. Model-Oriented Designing of an Electric Drive Control System and Its Investigation. Bulletin of MPEI. 2020;3:65—74. (in Russian). DOI: 10.24160/1993-6982-2020-3-65-74.

Введение

Микропроцессорная техника — основное средство управления техническими системами, в том числе, электроприводами. Для разработки и отладки их программного обеспечения используются различные методики и средства проектирования. Одной из современных методик является модельно-ориентированное программирование, основанное на автоматическом генерировании программного обеспечения из моделей, составленных в системе компьютерной математики Matlab [1]. Для учета микропроцессорной реализации модели дополняются элементами библиотек модельно-ориентированного программирования, интегрируемых в Matlab [2,3]. Таким образом, получается исполняемая модель программного обеспечения. Кроме сокращения времени на разработку, проектирования в графической форме, более удобной, чем структурированный текст, и генерирования программного обеспечения, независимого от Matlab, важнейшим свойством указанной методики считается то обстоятельство, что её можно успешно применять специалистом по теме проекта без опыта и квалификации программиста.

Исходя из публикаций [1 — 6], использование модельно-ориентированного программирования ограничено быстрым прототипированием при минимальной детализации модели с учётом микропроцессорной реализации. Часто детализация состоит лишь в назначении входов и выходов для ввода сигналов от датчиков и задатчиков и вывода управляющих сигналов. В этом случае программное обеспечение функционирует в микропроцессорной системе подобно компьютерной модели в Matlab. Все структурные элементы имеют одинаковые интервал повторения и приоритет выполнения. В простейших случаях, когда не проявляется ограниченность вычислительных ресурсов или при отсутствии жёстких требований по времени выполнения, подобная структура программного обеспечения вполне допустима. В сложных же системах требуется согласование выполнения многих задач. При этом вычислительные ресурсы микроконтроллера следует распределить между задачами для достижения высоких характеристик системы, детализировав программное обеспечение. Если стандартных средств модельно-ориентированного программирования для этого недостаточно, то дополнительно к ним может быть применено пользовательское программное обеспечение. Это не противоречит концепции модельно-ориентированного программирования, так как пользовательские подпрограммы интегрируются в исполняемую модель, структура которой разработана средствами моделирования. Кроме того, на их основе могут быть разработаны специализированные модельные блоки и библиотеки для использования в различных проектах.

Цель работы — модернизация микропроцессорной системы управления электропривода, разработанной с использованием средств модельно-ориентированного

программирования. Она направлена на достижение характеристик и функциональности электропривода, соответствующих аналогам [7]. Для этого на всех этапах проектирования была детально исследована работа электропривода и его отдельных подсистем, в том числе, для выявления ограничений, связанных с использованием модельно-ориентированного метода разработки. Для их устранения блоки Matlab/Simulink и подсистемы исполняемой модели на их основе были дополнены, частично или полностью были заменены модельными блоками на основе программного обеспечения на языке С.

Работа над проектом связана с решением следующих задач:

- управление электрическим двигателем постоянного тока независимого возбуждения и бесколлекторными двигателями постоянного и переменного тока;
- обеспечение работы электропривода в режимах регулирования момента и положения, а также стабилизации скорости;
- обеспечение работы электропривода в технологической системе и выполнение им вспомогательных функций.

Содержание и методология разработки

Система регулирования координат электропривода.

На рисунке 1 показана функциональная схема регулирования координат, реализованная в электроприводе и соответствующая аналогу [7].

В ней имеется контур регулирования электромагнитного момента (КРМ), который при работе электропривода в режиме стабилизации скорости охватывается контуром регулирования скорости (КРС), а в режиме слежения — контуром регулирования положения (КРП). Детализация контура регулирования момента и управления силовым преобразователем в зависимости от типа электрического двигателя ЭД на рис. 1 не представлены.

При использовании электрического двигателя постоянного тока для замыкания КРМ достаточно датчика тока в якорной цепи, так как электромагнитный момент пропорционален ему. При применении же бесколлекторных двигателей для работы контура и управления преобразователем кроме датчиков тока требуются датчики положения ротора. Управление преобразователем при использовании бесколлекторного двигателя постоянного тока осуществляется в зависимости от углового положения ротора, измеряемого по датчикам Холла [8]. При векторном управлении бесколлекторным двигателем переменного тока угловое положение ротора измеряется энкодером, установленным на его валу. Взяты пропорционально-интегральные ПИ-регуляторы момента (РМ) и скорости (РС), а также пропорционально-интегрально-дифференциальный ПИД-регулятор положения (РП). Двухконтурная следящая система по сравнению с трёхконтурной обладает повышенными быстродействием и точностью [9].

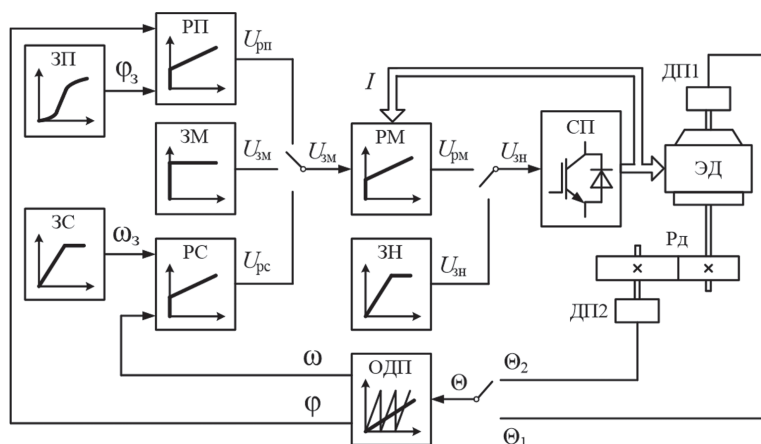


Рис. 1. Функциональная схема системы регулирования координат электропривода

В схеме имеются задатчики управляющих воздействий для разных режимов работы электропривода [10]. Задатчик скорости (ЗС) формирует процессы пуска и торможения с заданным ускорением, а задатчик перемещения (ЗП) — S-образную траекторию движения из исходной точки в заданную при введенных величинах ускорения и скорости. Предусмотрено задание момента ЗМ контуру КРМ при разомкнутых контурах КРС и КРП, а также задание напряжения питания электрического двигателя при разомкнутом контуре КРМ. Для измерения угловых положений ротора Θ_1 и оси приводного механизма Θ_2 , соединённой с ротором через редуктор Рд, использованы датчики положения ДП1 и ДП2 — инкрементальные энкодеры. Обработчик датчика положения ОДП преобразует угловое положение Θ , меняющееся в пределах $0...360^\circ$, или $0...Z_{\text{энк}}$ импульсов, в угол поворота ϕ , а также вычисляет скорость вращения ω , подсчитывая импульсы в течение интервала времени заданной продолжительности.

Архитектура программного обеспечения системы управления

Исполняемая модель системы управления электропривода составлена так, чтобы можно было регулировать координаты в соответствии со схемой, показанной на рис. 1, и выполнять вспомогательные функции управления. На ранних этапах [11, 12] программное обеспечение системы управления с микроконтроллером STM32 [13] было разработано в основном с использованием блоков Matlab/Simulink, а также библиотеки модельно-ориентированного программирования Waijung Blockset [2], интегрированной в Matlab. Исследование системы управления на разных этапах работы и накопленный опыт модельно-ориентированного программирования показали, что одних только стандартных средств и блоков Matlab недостаточно для разработки электропривода. Экспериментально выявлены следующие ограничения стандартных средств разработки:

- громоздкость и сложность подсистем на основе блоков Simulink и неоптимальность подпрограмм, сгенерированных из них автоматически;
- некорректное функционирование подпрограмм, автоматически сгенерированных из модельных подсистем на основе блоков Simulink;
- коллизии между модельными подсистемами при разных приоритетах их выполнения и интервалах повторения;
- ограниченная функциональность стандартных модельных блоков, в том числе при доступе к параметрам аппаратных модулей контроллера.

В целом, все ограничения ведут к снижению быстродействия, точности и диапазона регулирования координат, а также несвоевременному обновлению информации для различных подсистем и к их несогласованной работе. Кроме того, решение некоторых задач управления только стандартными средствами затруднительно. Поэтому были разработаны специальные подпрограммы и модельные блоки на их базе.

В результате экспериментального исследования электропривода при различных вариантах организации исполняемой модели выбрана итоговая структура, структурная схема которой продемонстрирована на рис. 2.

Детализация программного обеспечения направлена на реализацию управления электроприводом при компромиссе между имеющимися и требуемыми вычислительными ресурсами микроконтроллера:

- максимальном использовании модулей микроконтроллера в зависимости от конфигурации электропривода и типа двигателя;
- оптимизации критичных по времени выполнения подпрограмм.
- согласовании работы различных подсистем электропривода;
- распределении вычислительных и аппаратных ресурсов микроконтроллера, а также вычислительных ресурсов между задачами;



Рис. 2. Схема исполняемой модели системы управления электропривода

• своевременном обновлении информации при детектировании, захвате, обработке и генерировании сигналов.

Для выделения структурных элементов исполняемой модели (см. рис. 2), управления их выполнением и задания атрибутов их выполнения (способов тактирования, интервалов повторения и условий вызова) она реализована на основе базовой модельной схемы, функциональная схема которой изображена на рис. 3.

Каждому показанному на рис. 2 структурному элементу модели соответствует базовая схема, имеющая особенности в зависимости от назначения этого эле-

мента. В качестве элементов управления, предназначенных для вызова подсистем, использованы обработчики внешних и системных прерываний из библиотеки *Waijung Blockset*. Модельные блоки этой библиотеки работают в качестве модельных обработчиков и конфигураторов встроенных модулей микроконтроллера. Программные обработчики, заключённые в управляемые подсистемы *Matlab/Simulink* [14], реализованы в форме блоков вида *Basic Custom Code* [2], интегрирующих пользовательские подпрограммы на языке C в состав исполняемой модели. Выбор атрибутов выполнения подсистем и подпрограмм зависит от их назначе-



Рис. 3. Базовая модельная схема программного обеспечения:

a — функциональная схема; б, в — функциональные схемы модельной подсистемы обработки входов и выходов

ния в соответствии со структурой программного обеспечения (см. рис. 2). Приоритеты выполнения задач выбраны в зависимости от их значимости для управления электроприводом. Между подпрограммами с разными приоритетами могут происходить переходы, в том числе, с временным запретом прерываний, но они на рис. 2 не показаны. Задачи, требующие стабильного интервала повторения, выполняются по прерыванию от таймеров. Тактирование с помощью системного таймера использовано для задач фоновых циклов, не требующих строгой стабильности интервала повторения. Для назначения приоритетов выполнения задач, не имеющих интервала повторения, использованы системные прерывания `ext1` [13], вызываемые при программной установке системных флагов. Генерирование сигналов с различными видами широтно-импульсной модуляции (ШИМ) для управления трёхфазным мостовым преобразователем [15] в зависимости от типа электрического двигателя выполняется таймерами, работающими в режимах генерирования ШИМ [13].

Проанализированы наиболее значимые примеры применения модельных блоков на основе пользовательских подпрограмм для замены стандартных элементов Matlab. По трудоёмкости программирования они достаточно просты, а в качестве основы для них использованы автоматически сгенерированные подпрограммы.

Подсистема обработчика инкрементального энкодера

Для использования в электроприводе инкрементального энкодера разработана подпрограмма на языке C, включаемая в состав исполняемой модели в виде модельного блока. Такое решение связано с недостаточностью стандартных элементов Matlab/Simulink для осуществления использованного вычислительного алгоритма.

Для своевременного обновления информации об угле поворота электропривода при тактировании обработчика энкодера задан базовый фиксированный интервал длительностью 100 мкс. Скорость вращения электропривода измеряется путём подсчёта импульсов инкрементального энкодера с помощью таймера, работающего в режиме Quadrature Encoder Pulse (QEP) [13], в течение интервала времени наблюдения, кратного базовому интервалу. Для повышения точности за-

мера низких скоростей вращения продолжительность интервала зависит от числа импульсов, подсчитанных за предыдущий интервал наблюдения. Если подсчитанное число импульсов за такой интервал меньше порогового значения, что происходит при снижении скорости вращения, то продолжительность следующего интервала увеличивается. Если же число импульсов больше пороговой величины (при возрастании скорости), то следующий интервал наблюдения уменьшается. Минимальная и максимальная величины интервала наблюдения ограничены. Измеренная скорость усредняется для нескольких интервалов наблюдения.

Согласно представленному алгоритму в схеме, показанной на рис. 4, таймер *14* формирует базовые интервалы наблюдения, в течение которых таймер *1* в режиме QEP подсчитывает импульсы от энкодера, установленного на валу электрического двигателя. Подсчёт базовых интервалов для формирования интервалов наблюдения выполнен программно.

Аналогичным образом использованы таймеры *2* и *6*, указанные на рис. 2, при измерении угла поворота и скорости вращения оси механизма.

Подсистема регуляторов координат электропривода

Одна из причин применения пользовательских блоков — некорректная работа подпрограмм, автоматически сгенерированных из модельных схем на основе стандартных элементов Matlab. Экспериментально установлено, что в зависимости от вычислительной нагрузки микроконтроллера шаг интегрирования ПИД-регулятора из Simulink не остаётся постоянным при любом способе тактирования. Это же относится к интегрирующим элементам Simulink в модельных схемах задатчиков скорости и перемещения [16]. В результате динамика и точность регулирования координат отклоняются от заданных настроек. Поэтому для стабильности шага интегрирования разработаны подпрограммы модельных блоков регуляторов и задатчиков, тактируемых таймерами (см. рис. 2). При этом обеспечена частота повторения регуляторов скорости и положения — 3,0 кГц, регуляторов тока разных типов двигателей — 10 кГц, а задатчиков — 1,0 кГц.

Подсистема обработчика датчиков Холла

Подпрограмма обработчика сигналов от датчиков Холла при управлении бесколлекторным двигателем

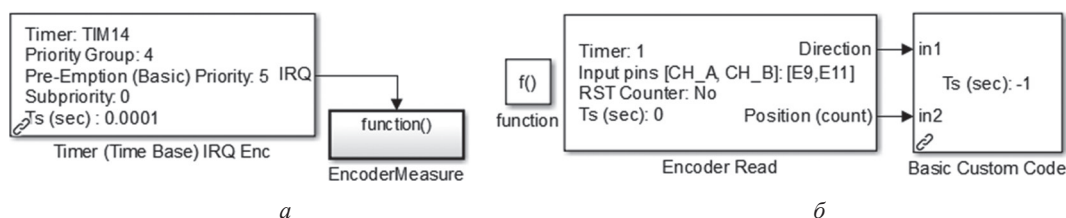


Рис. 4. Схема обработчика энкодера:

а — формирование интервалов наблюдения; б — подсистема EncoderMeasure вычисления угла поворота и скорости электропривода

постоянного тока [8, 10] критична по времени выполнения. Это связано с тем, что частота следования этих сигналов пропорциональна скорости вращения ротора, а вычислительная нагрузка, возникающая при их обработке, ограничивает её максимальную величину. Чтобы при высокой скорости вращения снизить вычислительную нагрузку, использована модельная схема, продемонстрированная на рис. 5. Она основана на разработанной подпрограмме, вызываемой в виде блока по прерыванию при перепаде сигналов датчиков Холла, которые подключены к цифровым линиям PD5 — PD7 микроконтроллера. Массив логических переменных для управления ключами силового преобразователя в данной подпрограмме вычисляется с помощью табличной функции в зависимости от сигналов датчиков Холла и заданного переменной $Uz1$ направления вращения.

Такая реализация подсистемы более компактна и экономна по времени выполнения, чем подсистема на основе модельных блоков [11]. Исследование электропривода с бесколлекторным двигателем постоянного тока, имеющим семь пар полюсов, показало, что при вращении его ротора с номинальной скоростью 10000 об/мин вычислительная нагрузка микроконтроллера, связанная с обработкой сигналов от датчиков Холла, не приводит к нарушению работы системы управления. Эта скорость вращения по частоте импульсов от датчиков Холла, частоте переключения ключей преобразователя и возникающей нагрузке контроллера соответствует электрической скорости вращения 70000 об/мин, равной скорости вращения двухполюсного двигателя.

Подсистема координатных и фазных преобразований

Выполнение координатных и фазных преобразований, связанных с векторным управлением бесколлекторным двигателем переменного тока, достаточно затратно по времени, так как для этого необходимы вычисления синуса и косинуса электрического углового положения ротора [8, 10]. Максимальная скорость вращения ротора подобного двигателя ограничена частотой повторения преобразований и своевременностью обновления информации об угловом положении

ротора. Поэтому был разработан модельный блок для векторного управления. В его подпрограмме функция синуса вычисляется таблично с дискретностью $\Delta\varphi = 0,27$ электрических градусов, исходя из разрешающей способности энкодера 1024 импульса на 360 электрических градусов. Это позволило сократить время выполнения вычислений, а частоту повторения координатных и фазных преобразований и частоту обновления информации об угловом положении ротора увеличить до 10 кГц. При этом экспериментально достигнута максимальная электрическая скорость вращения 20000 об/мин. Дальнейший рост скорости вращения приводит к несвоевременному выполнению вычислений по отношению к положению ротора. Для сравнения, при осуществлении векторного управления на основе стандартных блоков и математических функций продолжительность вычислений значительна. С учётом доступных вычислительных ресурсов микроконтроллера для такой подсистемы доступно выделение частоты повторения в 2,0 кГц. Её последующее увеличение нарушает работу системы управления в связи с недостатком вычислительных ресурсов для выполнения задач, имеющихся помимо векторного управления.

Подсистема управления силовым преобразователем

Приведем пример доступа к модулям контроллера для их настройки в зависимости от типа двигателя, поскольку средства Matlab позволяют конфигурировать их однократно при инициализации микроконтроллера. Поэтому с помощью стандартных средств Matlab невозможно, например, задать вид ШИМ при выборе типа двигателя.

Питание электрических двигателей идет от трёхфазного мостового преобразователя [15], схема которого показана на рис. 6, а. Якорная цепь двигателя постоянного тока включается между двумя плечами преобразователя. Частота ШИМ составляет 60 кГц. Для каждого типа двигателя использован характерный для него способ управления преобразователем [8 — 10] и контуры регулирования тока, тактируемые таймерами, как показано на рис. 2. Для выбранного типа двигателя включен лишь соответствующий ему таймер

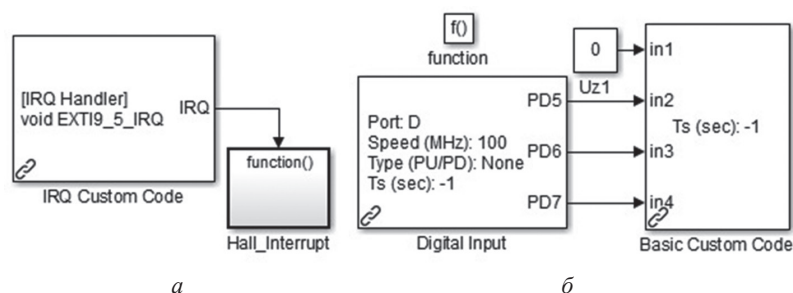


Рис. 5. Подсистема обработчика датчиков Холла:

а — модельная схема обработчика внешнего прерывания; б — модель Hall_Interrupt для обработки датчиков Холла

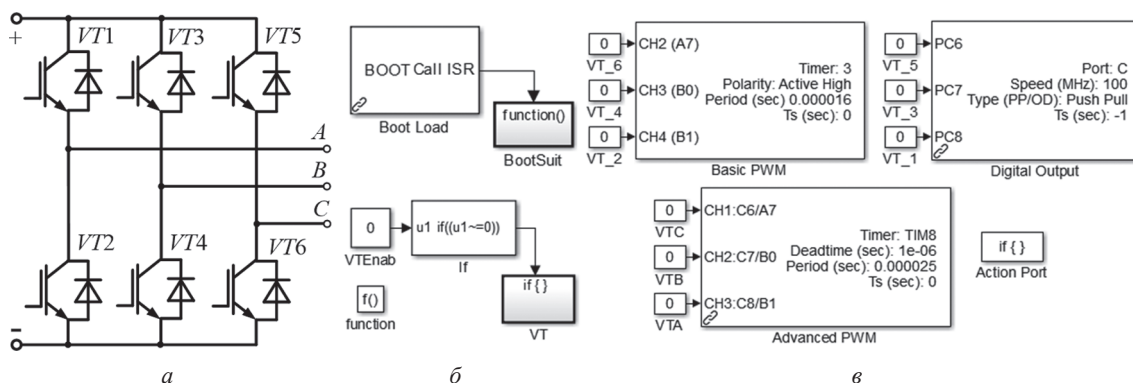


Рис. 6. Силовой преобразователь:

a — принципиальная схема; *б, в* — модельные блоки для инициализации управления и управления СП

тактирования контура регулирования тока. Реализация контура регулирования тока якорной цепи для двигателя постоянного тока рассмотрена в [18].

Для получения доступа к конфигурации генераторов ШИМ при инициализации микроконтроллера однократно выполняется подсистема BootSuit, в которую помещена подсистема VT вида If Action Subsystem [17], показанная на рис. 6, б. В подсистеме VT находятся модельные блоки генераторов ШИМ, изображенные на рис 6, в. Генератор однополярной ШИМ Basic PWM и блок выходов Digital Output предназначены для управления бесколлекторным двигателем постоянного тока. Генератор двухполярной ШИМ Advanced PWM служит для управления двигателем постоянного тока и бесколлекторным двигателем переменного тока. Так как флаг VTEenab для управления подсистемой VT равен нулю, то перечисленные генераторы и выходы микроконтроллера после его инициализации остаются выключенными. Однако автоматически сгенерированные подпрограммы для разрешения и запрета их работы присутствуют в проекте и могут быть применены в пользовательской подпрограмме для выбора типа электрического двигателя.

Другое направление модернизации стандартных блоков генераторов ШИМ связано с тем, что их входами является скважность, выраженная в процентах периода несущей частоты, и при каждом их выполнении выполняются проверка принадлежности скважности диапазону 0...100%, а также переход от процентов к числу тактов счёта таймера. Для сокращения вычислений автоматически сгенерированные подпрограммы доработаны для задания скважности непосредственно в тактах счёта таймера и помещены в подпрограммы регуляторов токов для задания скважности сразу при её определении. Снижение объёма вычислений при данном решении связано ещё и с тем, что подпрограммы стандартных блоков для генерирования ШИМ не только более длительны по времени выполнения, но должны выполняться в фоновом цикле (см. рис. 2). При этом для своевременного обновления скважности

частота их повторения значительно больше частоты повторения регуляторов тока.

Информационная подсистема и вспомогательные функции

В информационной подсистеме электропривода для работы в составе технологической системы предусмотрены обмены данными по интерфейсу UART (Universal Asynchronous Receive-Transmitter) с портами RS232 и USB-UART и по сетевой шине CAN (Controller Area Network) [12, 19]. Также возможен доступ к параметрам настройки электропривода, хранящимся в постоянном запоминающем устройстве (ПЗУ) по шине I2C (Inter-Integrated Circuit) и к их резервной копии во флэш-памяти микроконтроллера. При разработке требовалось обеспечить высокую интенсивность обмена сообщениями, снизив вычислительную нагрузку. Для этого модельные блоки Waijung Blockset обработчиков интерфейсов использованы лишь для обнаружения сообщений в приёмных буферах и записи ответных сообщений в буферы для передачи. Для доступа к параметрам электропривода, то есть к переменным программного обеспечения, применены подпрограммы на C, в которых системы команд и параметров электропривода представлены в виде массивов функций, обработка принятых сообщений осуществляется по прерыванию от таймеров, а для отправки сообщений системе управления верхнего уровня использованы системные прерывания (см. рис. 2). Такие решения позволили освободить вычислительные ресурсы микроконтроллера и повысить интенсивность обмена по UART и CAN до 150 сообщений в секунду.

Для повышения функциональности электропривода реализованы вспомогательные функции — монитор ресурсов, регистрация переходных процессов, регистрация неисправностей и событий управления в форме журнала, программируемая работа электропривода с тремя путевыми датчиками. Поскольку стандартных средств Matlab недостаточно для их разработки, то программное обеспечение было создано в виде нескольких подпрограмм. Монитор ресурсов предназна-

чен для измерения вычислительной нагрузки микроконтроллера. Счётчик монитора подсчитывает число выполнений фонового цикла в течение стабильного интервала времени, измеряемого таймером. Число выполнений фонового цикла обратно пропорционально вычислительной нагрузке. Экспериментально установлено, что она зависит от типа электрического двигателя, так как подпрограммы для управления ими отличаются по времени выполнения и по интервалам повторения, а также от интенсивности обмена данными по линиям связи. При разработке электропривода монитор ресурсов был использован для оценки целесообразности применения различных программных решений.

Регистрация переходных процессов предназначена для настройки электропривода на заданные показатели точности и динамики. Отсчёты тока, скорости вращения и угла поворота ротора электрического двигателя или оси приводного механизма захватываются по прерыванию от таймера со стабильным программируемым интервалом 1...5 мс. Эта же подсистема захватывает координаты, используемые в качестве начальных условий регуляторов и задатчиков. Контроль исправности аппаратной периферии системы управления осуществляется в фоновом цикле (см. рис. 2) с интервалом повторения около 100 мс или по запросу путём опроса контрольных линий датчиков и сравнения уставок с измеренными значениями координат.

Результаты разработки

Микропроцессорная система управления электропривода изготовлена в виде платы, помещённой в корпус (рис. 7, а). Осциллограммы координат электропривода с бесколлекторным двигателем постоянного тока, полученные с помощью приложения для персонального компьютера, показаны на рис. 7, б, 8 [12]. Мощность двигателя Махон — 50 Вт, номинальное напряжение — 25 В, имеется семь пар полюсов и номинальная скорость вращения — 10000 об/мин.

Контур регулирования тока благодаря использованию ПИ-регулятора настроен на апериодическое

протекание переходных процессов, как показано на рис. 7, б при отработке скачка задания тока $I_3 = 0,8$ А. Скорость и ускорение при отработке контуром регулирования положения S-образной траектории, представленной на рис. 8, а, равны заданным $\omega_3 = 8000$ об/мин и $\varepsilon_3 = 4000$ об/с². Исходя из конфигурации контуров скорости и положения, типов использованных регуляторов, а также согласно теории регулирования [8, 10, 20] можно заключить, что отработка скачков задания скорости $\omega_3 = 2000$ об/мин и перемещения $\varphi_3 = 10000$ импульсов энкодера (рис. 8, б) по динамике соответствуют оптимальному виду с максимальным затуханием колебаний. Приведем характеристики электропривода.

Характеристики системы управления электропривода

Мощность, Вт	200
Напряжение, В	25...50
Частота выборки токов/скорости/положения, кГц	40/10/10
Частота дискретизации регуляторов	
тока/скорости/положения, кГц	10/3,0/3,0
Скорость обмена по CAN, Мбит/с	1,0
Скорость обмена по RS232, бит/с	115200
Интенсивность обмена по CAN и RS232,	
сообщений в секунду	150
Максимальная скорость вращения двухполюсного бесколлекторного двигателя постоянного тока, об/мин	70000
Максимальная скорость вращения двухполюсного бесколлекторного двигателя переменного тока, об/мин	20000
Диапазон регулирования скорости	1:8000

Полученные экспериментальные характеристики являются результатами достижения баланса при распределении ресурсов микроконтроллера между задачами управления, а также доработки программного обеспечения для сокращения времени его выполнения и расширения функциональности управления.

Заключение

Средства модельно-ориентированного программирования совместно с пользовательскими подпрограммами позволили разработать полнофункциональное программное обеспечение системы управления электропривода с высокой степенью детализации.

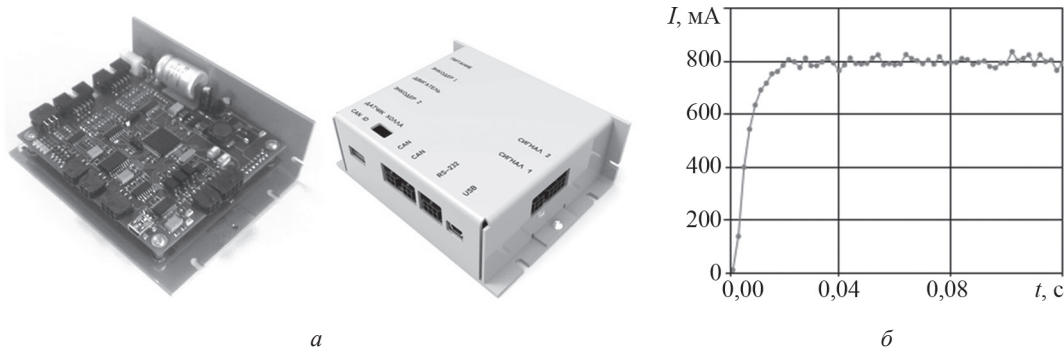


Рис. 7. Микропроцессорная система управления электропривода:

а — конструкция; б — осциллограмма отработки контуром регулирования тока скачка задания

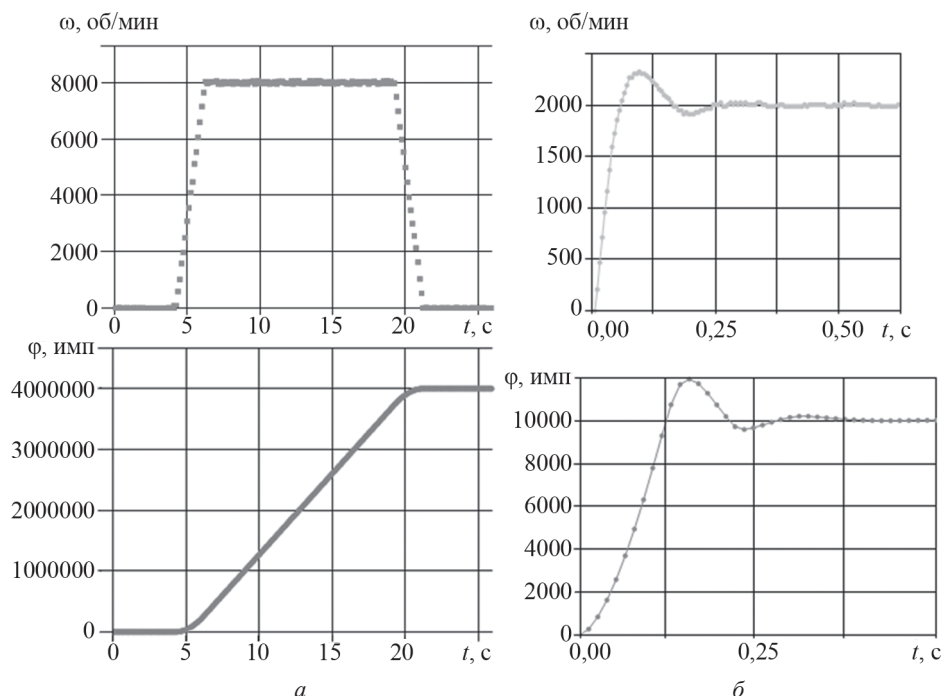


Рис. 8. Осциллограммы координат электропривода:

a — отработка задания S -образной траектории перемещения; b — отработка скачков задания скорости и перемещения

Экспериментально опробованы и применены различные решения, улучшающие характеристики электропривода и расширяющие функционал его управления, для осуществления которых стандартные средства

модельно-ориентированного программирования неудобны.

Получены экспериментальные характеристики электропривода.

Литература

1. **Model-Based Design** [Электрон. ресурс] www.mathworks.com (дата обращения 15.04.2019).
2. **Waijung Blockset** [Электрон. ресурс] <http://waijung.aimagin.com> (дата обращения 15.03.2019).
3. **Exponenta** [Электрон. ресурс] <http://www.exponenta.ru> (дата обращения 15.04.2019).
4. **Борисевич А.В., Омельченко Н.В.** Реализация векторного управления асинхронным электродвигателем на микроконтроллере STM32F4 // Современные научные исследования и инновации. 2014. № 4. Ч. 1 [Электрон. ресурс] <http://web.snauka.ru/issues/2014/04/33144> (дата обращения 15.04.2019).
5. **Krizan J., Ertl L., Bradac M., Jasansky M., Andreev A.** Automatic Code Generation from Matlab/Simulink for Critical Applications // Proc. 27th Canadian Conf. Electrical and Computer Eng. 2014. Pp. 1–6.
6. **Horvath K., Kuslits M.** Model-based Development of Induction Motor Control Algorithms with Modular Architecture // Proc. IEEE Intern. Power Electronics and Motion Control Conf. 2016. Pp. 133–138.
7. **Maxon Motor** [Электрон. ресурс] <http://www.maxon-motor.com> (дата обращения 10.04.2019).
8. **Анучин А.С.** Системы управления электроприводов. М.: Издат. дом МЭИ, 2015.

References

1. **Model-Based Design** [Elektron. Resurs] www.mathworks.com (Data Obrashcheniya 15.04.2019).
2. **Waijung Blockset** [Elektron. Resurs] <http://waijung.aimagin.com> (Data Obrashcheniya 15.03.2019).
3. **Exponenta** [Elektron. resurs] <http://www.exponenta.ru> (data obrashcheniya 15.04.2019). (in Russian).
4. **Borisevich A.V., Omel'chenko N.V.** Realizatsiya Vektornogo Upravleniya Asinkhronnym Elektrodvigatелеm na Mikrokontrollere STM32F4. Sovremennye Nauchnye Issledovaniya i Innovatsii. 2014;4;1 [Elektron. Resurs] <http://web.snauka.ru/issues/2014/04/33144> (Data Obrashcheniya 15.04.2019). (in Russian).
5. **Krizan J., Ertl L., Bradac M., Jasansky M., Andreev A.** Automatic Code Generation from Matlab/Simulink for Critical Applications. Proc. 27th Canadian Conf. Electrical and Computer Eng. 2014:1–6.
6. **Horvath K., Kuslits M.** Model-based Development of Induction Motor Control Algorithms with Modular Architecture. Proc. IEEE Intern. Power Electronics and Motion Control Conf. 2016:133–138.
7. **Maxon Motor** [Elektron. Resurs] <http://www.maxon-motor.com> (Data Obrashcheniya 10.04.2019).
8. **Anuchin A.S.** Sistemy Upravleniya Elektroprivodov. M.: Izdat. Dom MEI, 2015. (in Russian).

9. Башарин А.В., Новиков В.А., Соколовский Г.Г. Управление электроприводами. Л.: Энергоиздат, 1982.
10. Терехов В.М., Осипов О.И. Системы управления электроприводов. М.: Академия, 2006.
11. Полющенко И.С. Разработка системы управления электропривода на основе метода модельно-ориентированного программирования // Вестник МЭИ. 2016. № 6. С. 87—95.
12. Полющенко И.С. Разработка программного обеспечения для управления электроприводом в технологической системе с применением метода модельно-ориентированного программирования // Вестник МЭИ. 2017. № 4. С. 83—91.
13. STM32 Arm Cortex Microcontrollers [Электрон. ресурс] www.st.com (дата обращения 15.04.2019).
14. Дьяконов В.П. Matlab 6/6.1/6.5 + Simulink/4.5 в математике и моделировании. М.: СОЛОН-Пресс, 2008.
15. Розанов Ю.К., Соколова Е.М. Электронные устройства электромеханических систем. М.: Академия, 2004.
16. Терехин В.Б. Моделирование систем электропривода в Simulink (Matlab 7.0.1). Томск: Изд-во Томского политехн. ун-та, 2010.
17. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем. СПб.: Корона-Принт, 2001.
18. Полющенко И.С. Идентификация математической модели электромеханической системы // Вестник МЭИ. 2019. № 1. С. 69—78.
19. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. М.: Горячая линия – Телеком, 2009.
20. Егупов Н.Д. Методы классической и современной теории управления. Т. 2. М.: Изд-во МГТУ им. Н.Э. Баумана, 2000.
9. Basharin A.V., Novikov V.A., Sokolovskiy G.G. Upravlenie Elektroprivodami. L.: Energoizdat, 1982. (in Russian).
10. Terekhov V.M., Osipov O.I. Sistemy Upravleniya Elektroprivodov. M.: Akademiya, 2006. (in Russian).
11. Polyushchenkov I.S. Razrabotka Sistemy Upravleniya Elektroprivoda na Osnove Metoda Model'no-orientirovannogo Programmirovaniya. Vestnik MEI. 2016;6:87—95. (in Russian).
12. Polyushchenkov I.S. Razrabotka Programmogo Obespecheniya dlya Upravleniya Elektroprivodom v Tekhnologicheskoy Sisteme s Primeneniem Metoda Model'no-orientirovannogo Programmirovaniya. Vestnik MEI. 2017; 4: 83—91. (in Russian).
13. STM32 Arm Cortex Microcontrollers [Elektron. Resurs] www.st.com (Data Obrashcheniya 15.04.2019).
14. D'yakonov V.P. Matlab 6/6.1/6.5 + Simulink/4.5 v Matematike i Modelirovanii. M.: SOLON-Press, 2008. (in Russian).
15. Rozanov Yu.K., Sokolova E.M. Elektronnye Ustroystva Elektromekhanicheskikh Sistem. M.: Akademiya, 2004. (in Russian).
16. Terekhin V.B. Modelirovanie Sistem Elektroprivoda v Simulink (Matlab 7.0.1). Tomsk: Izd-vo Tomskogo Politekhn. un-ta, 2010. (in Russian).
17. German-Galkin S.G. Komp'yuternoe Modelirovanie Poluprovodnikovyykh Sistem. SPb.: Korona-Print, 2001. (in Russian).
18. Polyushchenkov I.S. Identifikatsiya Matematicheskoy Modeli Elektromekhanicheskoy Sistemy. Vestnik MEI. 2019;1:69—78. (in Russian).
19. Denisenko V.V. Komp'yuternoe Upravlenie Tekhnologicheskim Protseptom, Eksperimentom, Oborudovaniem. M.: Goryachaya liniya – Telekom, 2009. (in Russian).
20. Egupov N.D. Metody Klassicheskoy i Sovremennoy Teorii Upravleniya. T. 2. M.: Izd-vo MGTU im. N.E. Bauman, 2000. (in Russian).

Сведения об авторе:

Полющенко Игорь Сергеевич — кандидат технических наук, доцент кафедры электромеханических систем Смоленского филиала НИУ «МЭИ», e-mail: polyushchenkov.igor@yandex.ru

Information about author:

Polyushchenkov Igor S. — Ph.D. (Techn.), Assistant Professor of Electromechanical Systems Dept., Branch of NRU MPEI in Smolensk, e-mail: polyushchenkov.igor@yandex.ru

Статья поступила в редакцию: 12.04.2019

The article received to the editor: 12.04.2019