

Реализация векторного управления асинхронным электродвигателем на микроконтроллере STM32F4

Борисевич Алексей

Аннотация

В данной работе рассматривается векторное управление с ориентацией по полю (field oriented control, FOC). Показано, что алгоритм векторного управления может быть реализован с помощью прямой трансляции модели Simulink в исходный код на языке Си, с последующим его компиляцией и исполнением на отладочной плате STM32F4Discovery.

Введение

Векторное управление асинхронными двигателями – в настоящий момент доминирующая технология в области промышленного и транспортного электропривода переменного тока. Преимуществами векторного управления являются: обеспечение динамических характеристик, сравнимых с серводвигателями постоянного тока; увеличение КПД электропривода; возможность работы на малых скоростях в безредукторном приводе.

Структура статьи следующая: сначала кратко описывается алгоритм векторного управления по полю, потом описывается реализованная модель, далее описываются аппаратные средства и приводится результат реального испытания алгоритма.

1 Модель двигателя

На системном уровне описания, двигатель представляет собой элемент с тремя входами напряжения u_a, u_b, u_c , входом момента нагрузки T_m и выходами фазных токов i_a, i_b, i_c , а также выходом скорости ω (которую мы далее будем считать электрической, т.е. отличающейся от реальной механической в p раз).

Известно, что электрические величины трехфазной цепи (например, фазные токи) могут быть представлены во вращающейся со скоростью ω_s системе координат d и q как два ортогональных компонента i_q, i_d :

$$\begin{pmatrix} i_d \\ i_q \end{pmatrix} = \frac{2}{3} \begin{pmatrix} \cos \theta_s & \sin \theta_s \\ -\sin \theta_s & \cos \theta_s \end{pmatrix} \begin{pmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{pmatrix} \begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} \quad (1)$$

где $\theta_s(t) = \int_0^t \omega_s(\tau) d\tau$ – угловое положение вращающейся системы координат.

Преобразование вида (1) может быть аналогично применено и для фазных напряжений u_a, u_b, u_c . Выражение (1) называется инвариантным по амплитуде преобразованием Парка [1] или dq0-преобразованием. Важно отметить, что при соответствующем выборе ω_s (при вращении системы координат синхронно с вращением вектора

тока), величины i_q, i_d становятся постоянными и при этом $\sqrt{i_q^2 + i_d^2}$ – амплитуда переменного тока.

Рассмотрим инверсную Г-модель двигателя с ориентацией потокосцепления ротора ϕ_r вдоль оси d синхронно вращающейся системы координат [2]. В пространстве состояний модель реализуется системой дифференциальных уравнений четвертого порядка:

$$\begin{aligned}\dot{\phi}_r &= -\frac{R_R}{L_M}\phi_r + i_{sd}R_R \\ \dot{i}_{sq} &= -\frac{\omega}{L_\sigma}\phi_r - \frac{R_s}{L_\sigma}i_{sq} - \frac{R_R}{L_\sigma}i_{sq} - i_{sd}\omega_s + \frac{u_{sq}}{L_\sigma} \\ \dot{i}_{sd} &= \frac{R_R}{L_M L_\sigma}\phi_r - \frac{R_R}{L_\sigma}i_{sd} - \frac{R_s}{L_\sigma}i_{sd} + i_{sq}\omega_s + \frac{u_{sd}}{L_\sigma} \\ \dot{\omega} &= p\frac{T_e - T_m}{J}\end{aligned}\tag{2}$$

где i_{sq}, i_{sd} – токи статора во вращающейся системе dq-координат, u_{sd}, u_{sq} – напряжения статора в системе dq-координат, ω – скорость вращения вала двигателя, $T_e = p\phi_r i_{sq}$ – электромагнитный момент, развиваемый двигателем.

Параметры модели двигателя в (2) обозначены: R_R – приведенное сопротивление ротора, R_s – сопротивление статора, L_σ – приведенная индуктивность статора, L_M – приведенная индуктивность намагничивания, p – число пар полюсов, J – момент инерции ротора и нагрузки.

Система dq-координат вращается с синхронной скоростью вращения двигателя:

$$\omega_s = \omega + \frac{R_R i_{sq}}{\phi_r}\tag{3}$$

2 Векторное управление по полю

Метод векторного управления по полю можно резюмировать следующим образом.

Измеряемыми величинами являются фазные токи статора i_a, i_b, i_c и скорость вращения вала ω . Измеренные значения токов преобразуются с помощью (1) в постоянные i_{sd}, i_{sq} . Для такого преобразования необходимо знать синхронную скорость ω_s , которая зависит от измеренной скорости ω и скольжения. Таким образом, реализация векторного управления содержит в себе блок вычисления ω_s по соотношению (3) и первому уравнению в (2). Важно заметить, что вычисление ω_s использует значение потокосцепления ϕ_r , которое оценивается на основе модели двигателя (2). Отсюда, важным является знание параметров двигателя R_R и L_M .

Векторное управление для стабилизации скорости вращения двигателя состоит из трех ПИ-регуляторов, два из которых предназначены для поддержания токов статора. Обозначим i_{sd}^{ref} и i_{sq}^{ref} – уставки соответствующих токов статора. Из второго и третьего уравнения в (2) видно, что в первом приближении динамика токов статора относительно напряжений может быть представлена как система первого порядка с возмущениями, вызванными противо-ЭДС и взаимозависимостью токов.

Динамика ПИ-регуляторов может быть записана обычным образом

$$\begin{aligned}e_{i_{sd}} &= i_{sd}^{ref} - i_{sd}, \quad \dot{\xi}_{i_{sd}} = e_{i_{sd}}, \quad u_{sd} = K_p^{i_{sd}} e_{i_{sd}} + K_i^{i_{sd}} \xi_{i_{sd}} \\ e_{i_{sq}} &= i_{sq}^{ref} - i_{sq}, \quad \dot{\xi}_{i_{sq}} = e_{i_{sq}}, \quad u_{sq} = K_p^{i_{sq}} e_{i_{sq}} + K_i^{i_{sq}} \xi_{i_{sq}}\end{aligned}\tag{4}$$

При соответствующем выборе коэффициентов ПИ-регуляторов возможна достаточно точная регулировка токов и достижение $i_{sd} \approx i_{sd}^{ref}$, $i_{sq} \approx i_{sq}^{ref}$. В таком случае, можно записать сокращенную модель двигателя

$$\dot{\phi}_r = -\frac{R_R}{L_M}\phi_r + i_{sd}^{ref} R_R, \quad \dot{\omega} = p \frac{p\phi_r i_{sq}^{ref} - T_m}{J} \quad (5)$$

Заметим, что i_{sd}^{ref} – это уставка тока намагничивания, которая определяется пользователем, обычно в процентах от номинального тока. Значение i_{sq}^{ref} устанавливается ПИ-регулятором, обеспечивающим стабилизацию скорости ω :

$$e_\omega = \omega^{ref} - \omega/p, \quad \dot{\xi}_\omega = e_\omega, \quad i_{sq}^{ref} = K_p^\omega e_\omega + K_i^\omega \xi_\omega \quad (6)$$

Выходы ПИ-регуляторов (4) – это напряжения статора, которые подаются непосредственно на двигатель после обратного преобразования координат:

$$\begin{pmatrix} u_a \\ u_b \\ u_c \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{pmatrix} \begin{pmatrix} \cos \theta_s & -\sin \theta_s \\ \sin \theta_s & \cos \theta_s \end{pmatrix} \begin{pmatrix} u_{sd} \\ u_{sq} \end{pmatrix} \quad (7)$$

3 Модель в Simulink

Алгоритм векторного управления реализован в виде модели Simulink, показанной на рисунке 1.

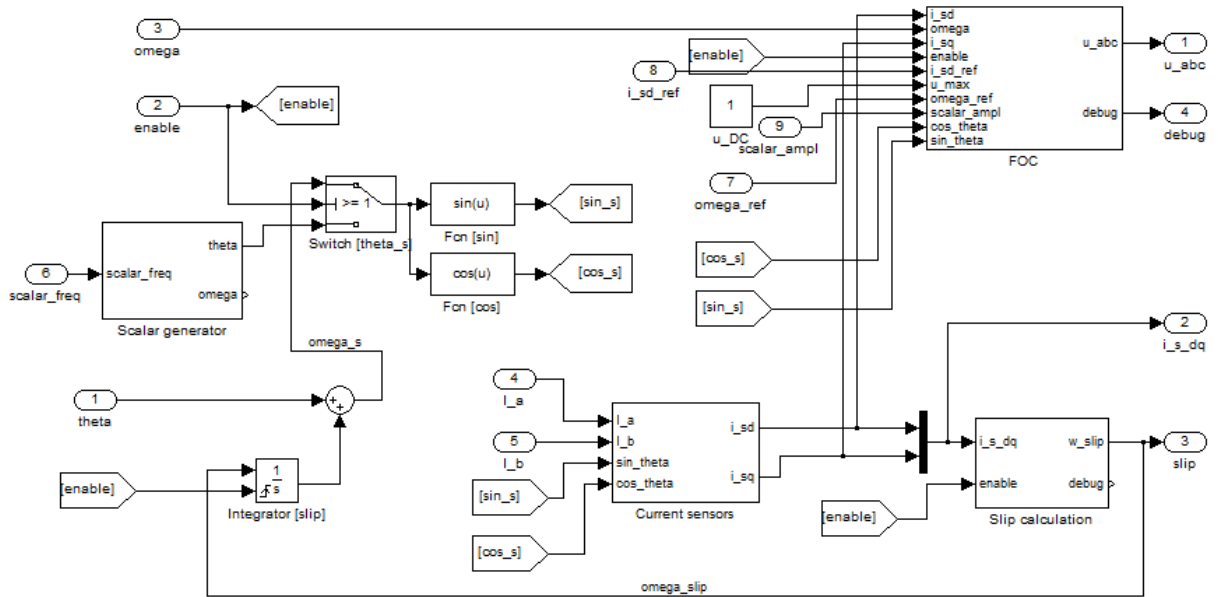


Рисунок 1. Модель векторного управления в Simulink.

Для упрощения отладки в модель включена возможность управления в скалярном (U/f) режиме, который является частным случаем векторного, в котором $u_{sd} = 0$, u_{sq} – определяет амплитуду выходного напряжения, а $\omega_s := \text{const}$ – уставка выходной частоты, $\theta_s = \omega_s t$.

Входы модели:

- theta – угловое положение ротора (электрический угол);

- enable – если enable=1, то активизируется векторное управление, если enable=0 – то скалярное;
- I_a, I_b – токи фаз А и В с датчиков (в А);
- scalar_freq – задание частоты для скалярного режима (в Гц);
- omega_ref – задание частоты для векторного режима (в рад/сек);
- i_sd_ref – задание тока намагничивания для векторного режима (в А);
- scalar_ampl – амплитуда выходного напряжения для скалярного режима (нормированное значение, от 0 до 1).

Выходы модели:

- u_abc – выходы напряжения фаз А,В,С (нормированное значение от -1 до 1);
- i_s_dq – ток статора во вращающейся синхронной системе координат (в А);
- slip – скорость скольжения;
- debug – выход для отладочных сигналов.

Модель состоит из нескольких укрупненных подсистем:

- Scalar generator – генерация $\theta_s(t)$ для скалярного режима;
- FOC – реализация регуляторов векторного алгоритма (4) и (6), а также инверсное преобразование (7) для генерации трехфазного напряжения;
- Current sensors – преобразование измеренных значений тока к вращающейся системе координат, согласно (1);
- Slip calculation – вычисление скорости скольжения согласно (3) и первому уравнению в (2).

Центральной частью реализации является подсистема FOC, показанная на рисунке 2. В данном блоке векторное управление реализуется в точности так, как это было описано в предыдущем разделе.

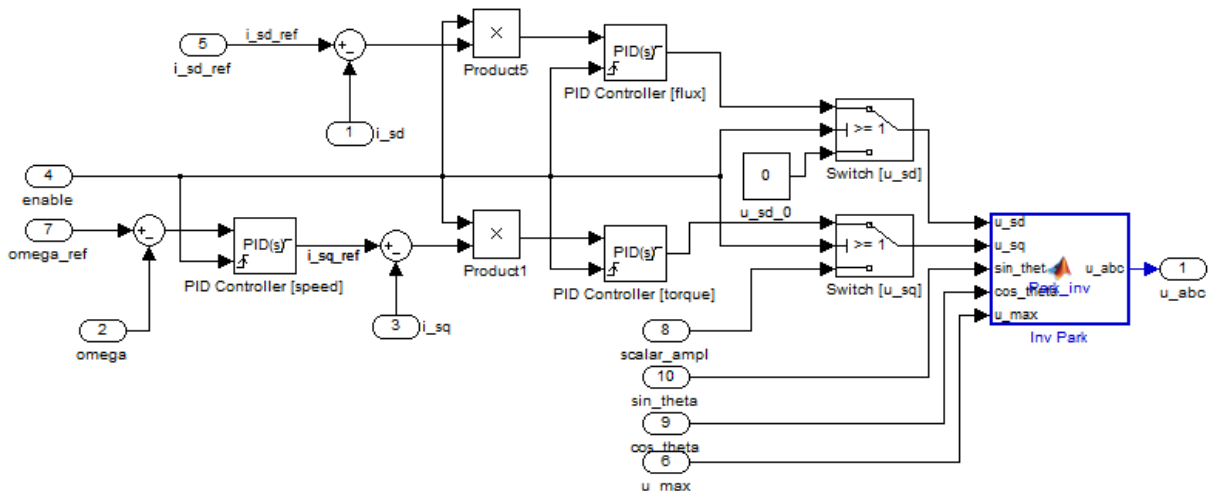


Рисунок 2. Подсистема FOC, реализующая (4), (6) и (7).

С целью оптимизации, вычисление тригонометрических функций $\sin \theta_s$ и $\cos \theta_s$ сосредоточено в одном месте модели.

Для верификации реализованного алгоритма была создана модель, в которой описанный векторный регулятор соединен с моделью асинхронного электродвигателя из библиотеки SimPowerSystems. Результаты моделирования полностью подтверждают работоспособность алгоритма управления.

4 Аппаратная реализация

В качестве аппаратной платформы выбрана плата STM32F4DISCOVERY, процессор STM32F407 которой позволяет реализовывать сложные численные алгоритмы и содержит всю необходимую периферию. Фрагменты кода для генерации ШИМ и считывания инкрементального энкодера взяты из библиотеки STM32 FOC firmware libraries v2.0 [3].

Для силового модуля был использован частотный преобразователь китайского производства. Он состоит из платы управления и платы ключей, соединяемых проводом-шлейфом. Плата STM32F4DISCOVERY была подключена вместо штатной платы управления.

В качестве датчиков тока использовались LEM HX03-P, сигнал с которых нормировался к интервалу 0 – 3 В с помощью операционного усилителя. Компенсация дрейфа нуля датчиков в микроконтроллере реализована с помощью скользящего среднего с интервалом 1 с для динамической калибровки нуля.

Задание скорости осуществляется с помощью переменного резистора, подключенного к микроконтроллеру. Для отображения числовой информации к микроконтроллеру также был подключен ЖКИ индикатор 20x4 типа AV2040.

Схема задействованных выводов микроконтроллера и их подключение представлена на рисунке 3.

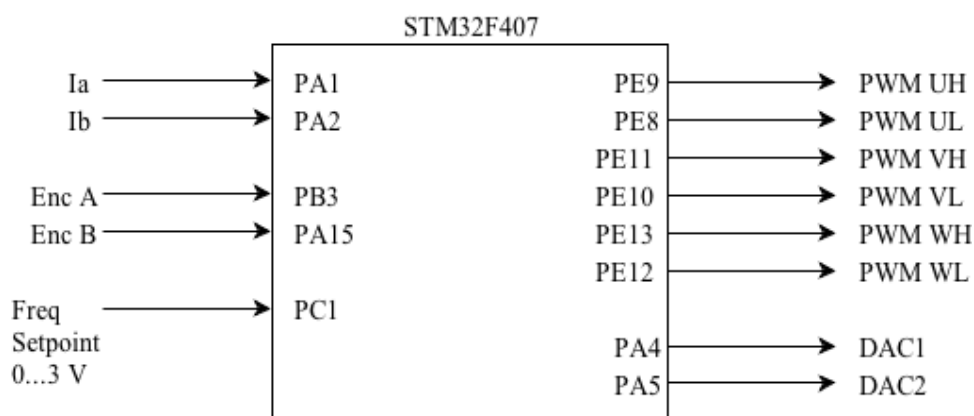


Рисунок 3. Подключение микроконтроллера.

Модель Simuink алгоритма векторного управления была транслирована в код на языке Си с помощью встроенного в MATALB средства Embedded Coder [4]. В качестве стратегии интегрирования выбрана одношаговая схема (ode1). Вся арифметика алгоритма реализована с плавающей точкой одинарной точности (float), которую аппаратно поддерживает микроконтроллер STM32F4. Ядро процессора работает на частоте 168 МГц. Время расчета одного цикла алгоритма – 150 мкс. Частота дискретизации алгоритма была выбрана в 500 мкс.

В качестве компилятора исходного кода использовался KEIL uVision 4.70. Откомпилированная программа занимает 25 кБайт памяти.

Для отладки внутренние сигналы алгоритма управления выводились на два встроенных в микроконтроллер ЦАП.



Рисунок 4. Общий вид макета векторного управления электродвигателем.

Для испытаний был выбран двигатель DRS71S4 производства SEW Eurodrive, для которого известны параметры $R_R = 11.6$ Ом, $L_M = 987$ мкГн. Для тестирования механической нагрузки, шкив двигателя был соединен с управляемым серводвигателем, создающим постоянный тормозящий момент 2 Нм. Отработка профиля скорости показана на рисунке 5.

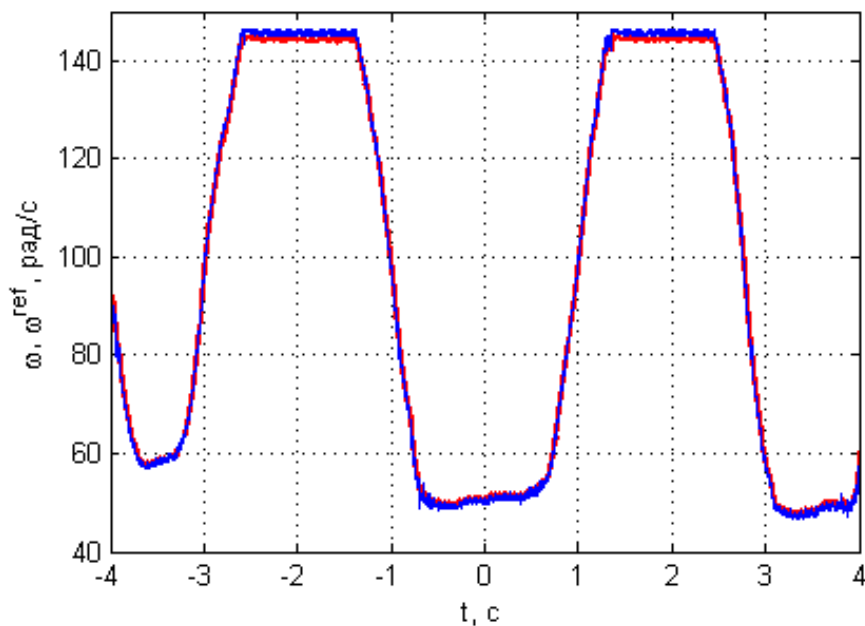


Рисунок 5. Уставка ω^{ref} и действительное значение скорости ω .

Заключение

В результате испытаний было установлено, что алгоритм векторного управления может быть реализован с помощью автоматической генерации кода из модели в Simulink. Возможностей и быстродействия микроконтроллера STM32F4 достаточно для выполнения алгоритма векторного управления с вычислениями с плавающей точкой. Наличие встроенного в микроконтроллер АЦП, а также интерфейса с энкодером позволяет создать компактное и дешевое устройство для промышленной реализации частотного преобразователя, а также в исследовательских целях для разработки новых алгоритмов управления.

Список литературы

- [1] Калачев Ю.Н. Векторное регулирование (заметки практика). Методическое пособие. — М.: ЭФО, 2013. — 63 с.
- [2] Riccardo Marino, Patrizio Tomei, Cristiano M. Verrelli, Induction Motor Control Design (Advances in Industrial Control), — Springer, 2010. — 350 pp.
- [3] UM0483, STM32F103xx AC induction motor IFOC software library V2.0. [Электронный ресурс] — Режим доступа. — URL: <http://www.stmcu.org/download/index.php?act=down&id=3275> (дата обращения 16.10.2013).
- [4] Материалы семинара «Генерация кода из моделей Simulink и последующая проверка в реальных условиях», 14 октября 2010 [Электронный ресурс] — Режим доступа. — URL: (дата обращения 16.10.2013). <http://www.compel.ru/wordpress/wp-content/uploads/2011/12/Генерация-кода-из-моделей-Simulink.pdf>