

УДК 62.83.52

О.И. ТОЛОЧКО (д-р техн.наук, проф.), **П.И. РОЗКАРЯКА** (канд.техн.наук),
Н.М. ГОРОБЕЦ (канд.техн.наук, доц.)

Донецкий национальный технический университет
 pavel_rozkar@mail.ru

К ВОПРОСУ ОБ ИЗМЕНЕНИИ ТИПОВЫХ СТРУКТУР ЦИФРОВЫХ СИСТЕМ УПРАВЛЕНИЯ КОМПЛЕКТНЫМИ ЭЛЕКТРОПРИВОДАМИ

Рассмотрены вопросы изменения типовых структур цифровых систем управления в современных комплектных электроприводах на примере системы векторного управления синхронным двигателем с постоянными магнитами фирмы Control Techniques. Запроектированная система регулирования скорости дополнена наблюдателем состояния, идентифицирующим момент статического сопротивления, который используется для компенсации влияния нагрузки на скорость привода в установившемся режиме.

Введение. В настоящее время в практике регулируемого электропривода широкое распространение получили преобразователи в комплекте с системой управления, реализованной на цифровом сигнальном процессоре. В этом случае пользователю предоставляются достаточно ограниченные возможности по выбору одной из типовых систем управления, предусмотренных разработчиками, и их параметрированию. При необходимости усложнения системы управления с целью улучшения качества регулирования или расширения её функциональных возможностей возникают проблемы, связанные с реализацией дополнительных устройств и сопряжением их с основным процессором.

Целью данной работы является демонстрация одного из возможных методов изменения структуры системы управления комплектным электроприводом на примере сервопривода фирмы Control Techniques.

Материал и результаты исследования.

Исследуемый комплектный электропривод предназначен для векторного управления синхронным двигателем с постоянными магнитами (СДПМ).

Схематическое изображение лабораторного комплекса представлено на рис. 1. В его состав входят две электрические машины: СДПМ *Dutymax DS* типа *R95DCS300AAA* и двигатель постоянного тока (ДПТ) независимого возбуждения *112L-4MGV*, установленные на одном валу и жестко соединенные между собой. ДПТ используется в качестве нагрузочной машины для синхронного электродвигателя. Для управления СДПМ используется частотный преобразователь *Unidrive SP1404*, а для управления ДПТ – тиристорный преобразователь *MENTOR II*. Неотъемлемой частью преобразователей являются встроенные контроллеры, которые предназначены для реализации базовых алгоритмов управления, предусмотренных разработчиками.

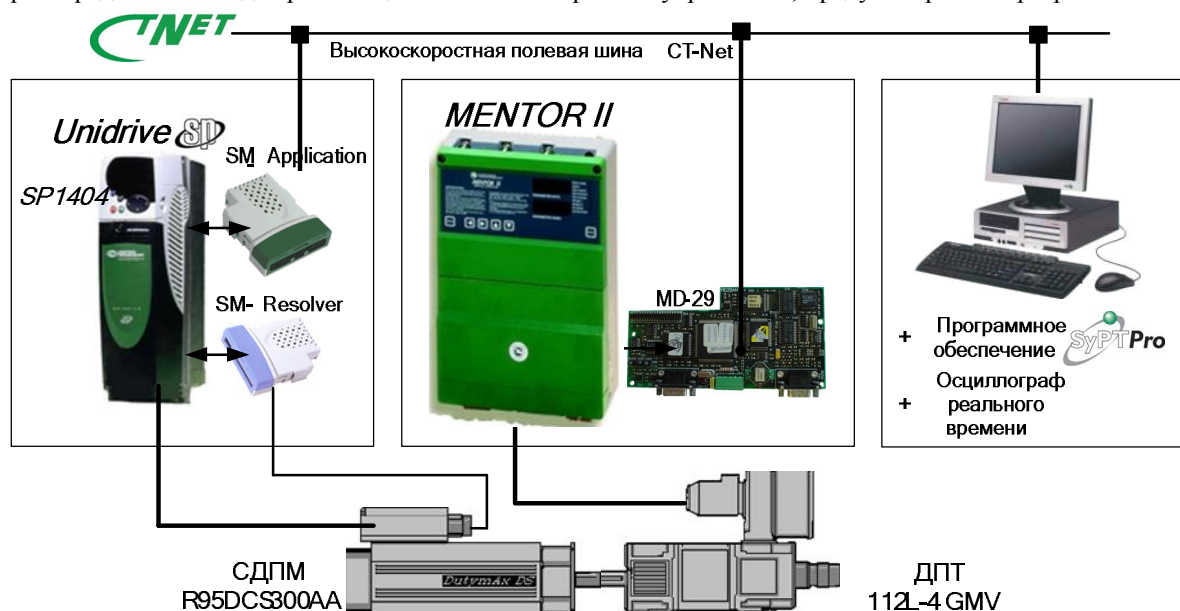


Рисунок 1 – Лабораторный комплекс для исследования системы векторного управления СДПМ

Преобразователь частоты укомплектован также сопроцессорным модулем *SM-Application*, а тиристорный преобразователь – модулем *MD29*. Оба этих дополнительных модуля предназначены для реализации приложений пользователя. Модуль *SM-Application* построен на основе высокопроизводительного специализированного процессора, оснащен *Flash*-памятью емкостью 384кБ, оперативной памятью емкостью

© Толочко О.И., Розкаряка П.И., Горобец Н.М., 2011

80кБ. Модуль MD29, реалізований на 32-х битовом RISC-процесорі INTEL 960; має 96 кбайт флеш-пам'яті і 8 кбайт оперативної пам'яті. В корпусі синхронної машини встановлений датчик положення типу резольвер. Св'язь датчика положення з преобразователем частоти здійснюється через додатковий модуль SM-Resolver. С допомогою високоскоростної шини CT-Net забезпечується обмін даними між преобразователями і комп'ютером, також входящим в склад стенда і призначеним для програмування додаткових модулів в середі SyPT-Pro і забезпечення роботи віртуального осцилографа ST-Scope в режимі реального часу [1-5].

В комплектному електроприводі передбачена стандартна полеорієнтована система векторного управління з регуляторами ортогональних складових току статора, а також з регуляторами швидкості і положення.

Рассмотрим, каким образом можно дополнить типовую систему регулирования скорости наблюдателем состояния (НС), идентифицирующим момент статического сопротивления на валу двигателя. Синтез такого НС выполнен в [6]. Он может быть использован, в частности, для компенсации статической просадки скорости при набросе нагрузки в системе с пропорциональным регулятором скорости (П-РС). Наличие такого устройства не предусмотрено стандартной системой.

Структурная схема исследуемой системы показана на рис.2.

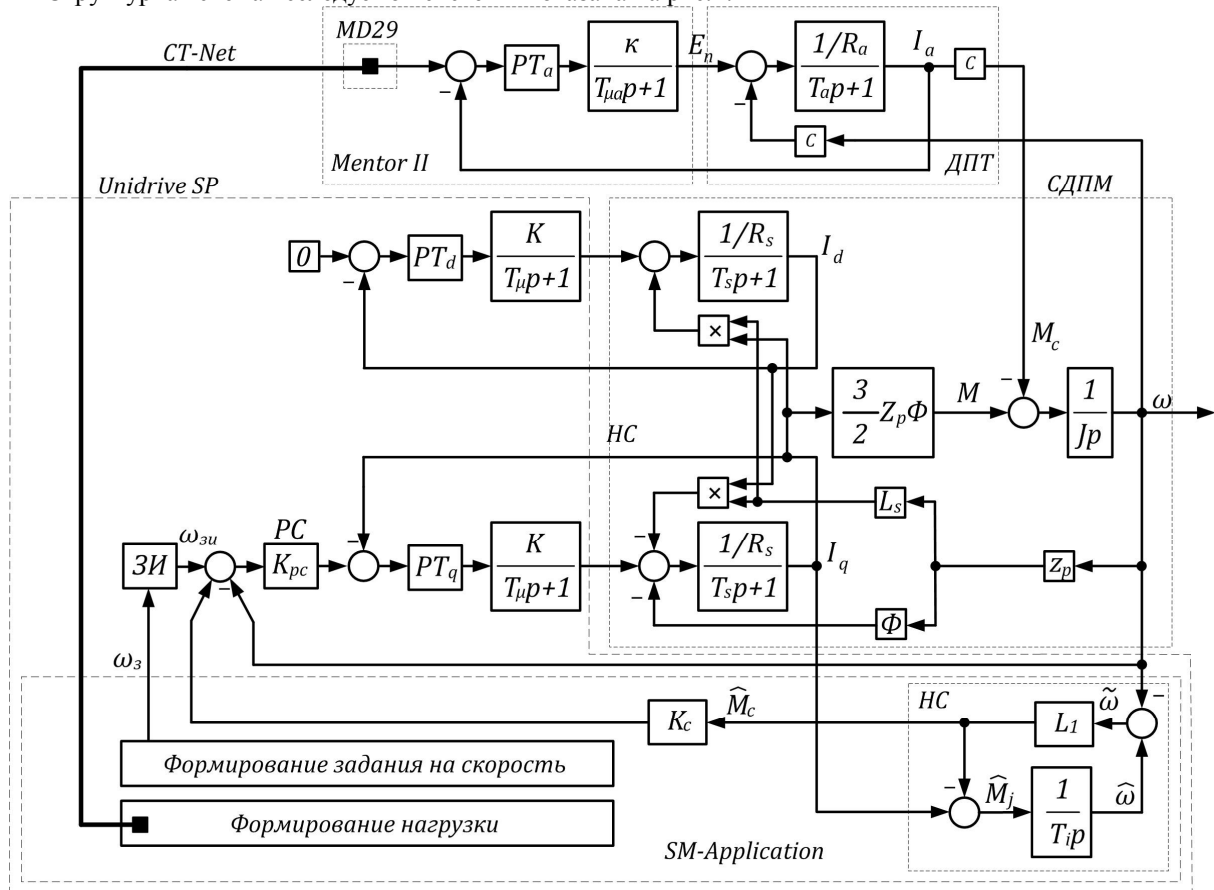


Рисунок 2 - Структурная схема исследуемой системы

На рис. 2 обозначены: PT_a – регулятор тока якоря нагрузочной машины ДПТ; PT_d, PT_q – регуляторы ортогональных составляющих тока статора исследуемой машины СДПМ; $ЗИ, PC$ – задатчик интенсивности и регулятор скорости исследуемого двигателя; $НС$ – наблюдатель состояния; E_n, I_a, M, M_c – ЭДС преобразователя, ток якоря, электромагнитный момент и момент статического сопротивления СДПМ; $\hat{\omega}, \hat{M}_c$ – идентифицированная скорость и идентифицированный момент статического сопротивления.

В передаточных функциях рис.2 использованы следующие обозначения: R_a, T_a – сопротивление и электромагнитная постоянная времени якорной цепи ТП-Д; c – конструктивный коэффициент ДПТ; R_s, L_s, T_s, z_p, Φ – сопротивление, индуктивность, электромагнитная постоянная времени статора, число пар полюсов и поток постоянных магнитов СДПМ; J – суммарный момент инерции ДПТ и СДПМ; T_ϕ – постоянная времени фильтра; K_{pc} – коэффициент усиления пропорционального регулятора скорости; $L_1 = \frac{2\Omega_0 J}{3z_p \Phi}$ – коэффициент обратной связи НС; Ω_0 – среднегеометрический корень характеристического полинома наблюдателя; $K_c = \frac{2}{3z_p \Phi} K_{pc}$ – коэффициент усиления корректирующей связи; $T_i = \frac{2J}{3z_p \Phi}$ – постоянная времени интегрирования модели механической части привода в НС.

Входными сигналами рассматриваемого наблюдателя являются моментобразующая составляющая тока статора I_q и скорость двигателя ω . Для компенсации влияния статического момента на скорость двигателя выходной сигнал НС нужно подать на вход регулятора PC с коэффициентом K_c .

Регулятор тока якоря реализован на процессоре, входящем в состав преобразователя *MENTOR II*, регуляторы токов PT_d , PT_q , PC и $3I$ – на процессоре преобразователя *Unidrive SPI404*. НС, не входящий в стандартную систему управления, выполнен на сопроцессорном модуле *SM-Application*. Также в *SM-Application* содержится блок инструкций, который формирует задание на нагрузку и по сети *CT-Net* передает данные в тиристорный преобразователь *MENTOR II* двигателя постоянного тока.

Программа реализации перечисленных блоков написана в среде *SyPTPro*.

Рассмотрим более подробно программу управления модулем *SM-Application*. Пользовательское приложение состоит из отдельных разделов (задач), которые выполняются в строго определенной последовательности. К таким разделам относятся (в порядке приоритета): «*Initial*», «*Event*», «*Pos*», «*Clock*» и «*Background*». При подаче питания на частотный преобразователь первыми выполняются инструкции, записанные в разделе «*Initial*», в котором задаются значения констант и начальные значения сигналов системы управления, а также определяется ее конфигурация. После этого начинают выполняться задачи реального времени разделов «*Pos*» (их может быть несколько, например «*Pos0*» и «*Pos1*») и «*Clock*». Инструкции, помещенные в данные разделы, циклически повторяются через фиксированные интервалы времени (периоды дискретности). Период дискретности для задачи «*Clock*» ($T_{\Delta 1}$) может принимать целочисленные значения от 1 до 200 мс, а для задач «*Pos0*» и «*Pos1*» ($T_{\Delta 2}$) – строго фиксированные значения: 250 мкс, 500 мкс, 1 мс, 2 мс, 4 мс и 8 мс.

Инструкции разделов «*Event*» имеют самый высокий приоритет, поэтому их задачи содержат очень малое число инструкций. Они прерывают работу разделов «*Pos*» и «*Clock*», и только по окончании их выполнения программа управления продолжает прерванные инструкции разделов «*Pos*» и «*Clock*». Таким образом, в разделы «*Event*» целесообразно помещать алгоритмы обработки определенных событий, например, аварийных ситуаций.

«*Background*» – является фоновой задачей. Она выполняется только в паузах между выполнениями инструкций других разделов. Эта задача организуется в виде бесконечного цикла. Если она будет завершена, то она больше не будет выполняться.

На рис. 3 представлена концепция взаимного прерывания задач.

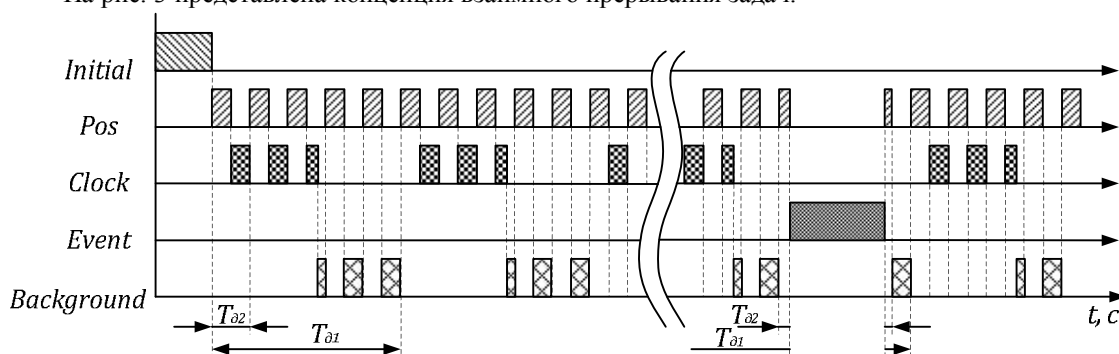


Рисунок 3 – Временная диаграмма выполнения разделов инструкций

Как видно из временной диаграммы, выполнение инструкций раздела «*Clock*» прерывается выполнением раздела «*Pos*» (он имеет более высокий приоритет, чем «*Clock*», и меньший период дискретности). Поэтому разработчики рекомендуют в разделы «*Pos*» помещать инструкции, связанные с корректировкой контуров регулирования скорости и (или) положения, а в раздел «*Clock*» – инструкции, не требующие такого быстрого выполнения как задачи раздела «*Pos*», например формирование задатчика положения.

Все инструкции программы должны располагаться только внутри определенной задачи. Инструкции разделов «*Pos*» и «*Clock*», должны быть выполнены за время, меньшее, чем их периоды дискретности; иначе задачи с меньшими приоритетами (выполняемые в паузах этих задач) не получат времени для своего выполнения. Это может привести к отключению процессора по перегрузке.

Среда разработки пользовательских программ управления *Sypt-Pro* содержит библиотеку стандартных команд и функций, а также поддерживает несколько языков программирования (DPL – язык инструкций, LD – язык лестничных диаграмм, FBD – язык функциональных блоков).

В рассматриваемой среде разработки определены 3 типа переменных: целочисленная, на которую в памяти отводится 32 бита (из них 1 бит знаковый); переменная с плавающей запятой обычной точности, также занимающая 32 бита (1 бит – знак, 8 – порядок, 23 – мантисса); переменная с плавающей запятой с двойной точностью (64 бита, из которых 1 бит – знаковый, 11 – порядок, 52 – мантисса).

Пользовательская программа управления, разработанная на языке DPL, которая дополняет проектную систему наблюдателем состояния, приведена на рис. 4. Она состоит из трех разделов: «*Initial*», «*Clock*» и «*Background*».

Параметры, инициализируемые в разделе «*Initial*», можно разделить на три группы: 1) параметры,

инициализирующие подключение различных устройств; 2) глобальные переменные, значения которых изменяются в дальнейшем в процессе работы частотного преобразователя *Unidrive SP*; 3) переменные, не предусмотренные проектировщиками (локальные переменные сопроцессорного модуля *SM-Application*).

Имена переменных, относящихся к первым двум группам, имеют следующий формат *#<Номер-меню>.<Номер-параметра>*. Размерность их значений определена проектировщиками и не может быть изменена пользователем. Перечень меню и их параметров, а также диапазон и размерность возможных значений приведены в документации [1-5]. Например, оператор *«#2.02=1»* предписывает подать задание на РС от ЗИ, а оператор *«#2.02=0»* – подать задание на РС скачком; оператор *«#2.11=0.2»* определяет темп разгона как время в секундах, затрачиваемое на изменение скорости на 1000 об/мин.

Имена переменных, относящихся к третьей группе, задаются пользователем по следующим правилам: состав имени – латинские буквы, цифры и знак подчеркивания; первый символ – буква. Тип переменной определяется ее именем. По умолчанию все переменные считаются переменными с плавающей запятой с двойной точностью. Для того чтобы сделать локальную переменную целочисленной необходимо дополнить ее имя в конце знаком «%».

Кроме переменных, рассмотренных выше в примерах, в разделе *«Initial»* рассматриваемой программы заданы: период такта выполнения раздела *«Clock»*, времена характерных точек желаемой тахограммы и желаемой нагрузочной диаграммы $t_{1\%}$, ..., $t_{5\%}$, коэффициенты обратных связей $L1$ и Kc , начальное значение интегратора W_c , на котором построен НС и его постоянная интегрирования Ti . Все времена заданы в долях периода дискретности.

В разделе *«Clock»* выполнена программная реализация наблюдателя состояния, сформированы сигналы задания на скорость, на нагрузку и корректирующая связь по идентифицированному моменту сопротивления, а также выполнена запись сигналов, необходимых для визуализации переходных процессов. Здесь использованы следующие глобальные и локальные переменные: *#1.18*, *#3.2* – заданная и измеренная скорость двигателя [об/мин] соответственно, *#4.2* – моментобразующая составляющая тока статора [А], *#3.22* – корректирующая обратная связь [об/мин], заведенная на вход РС, *#6.30* – разрешение вращения вперед, *#6.32* – разрешение вращения назад, *load%* – разрешение формирования нагрузки (1 – разрешение, 0 – запрет), *t%* – счетчик выполняемых тактов.

Интегратор, входящий в состав НС, реализован при помощи стандартной функции интегрирования *ITerm(Error%, iGain%, Limit%, Pset%, PsetE%, Reset%, Hold%)*, где *Error%* – интегрируемое значение, *iGain%* – умноженный на 1000 коэффициент усиления интегратора, *Limit%* – ограничение интегратора, *Pset%* – начальное условие интегратора, *PsetE%* – сброс интегратора в состояние *Pset%*, *Reset%* – сброс интегратора в 0, *Hold%* – удерживание текущего значения интегратора.

В разделе *«Background»* записаны инструкции, формирующие задание на контур тока якоря нагрузочной машины. Бесконечный цикл реализуется с помощью метки *«top:»* и оператора безусловного перехода *«goto»*. Здесь использованы следующие сетевые функции:

1) *Q% = CheckNode(Node%)* – функция, проверяющая наличие устройства в сети *CT-Net*, где *Node%* – адрес вызываемого устройства, которым в данном случае является модуль *MD29* с сетевым адресом 11.

2) *status% = WRNET(Node%, Menu%, Parametr%, Value%, PosD%, WaitCode%)* – функция, записывающая данные в другое устройство по сети, где *Node%* – адрес принимающего устройства, *Menu%*, *Parametr%* – номера меню и параметра, из которых формируется имя глобальной переменной для записи информации, *Value%* – пересылаемая информация, *PosD%* – количество знаков в дробных частях пересылаемых чисел, *WaitCode%* – предельное время, предоставляемое функции для передачи информации и подтверждения завершения этой операции [мс].

```
Initial{
#1.14=5           // Открытие раздела «Initial»
#15.11=5         // Выбор источника задания скорости (0 – аналоговый вход, ..., 5 – точное задание)
#15.11=5         // Такт выполнения раздела «Clock» = 5 мс
#2.02=1          // Включение ЗИ
#2.11=0.2        // Темп разгона (увеличение скорости на 1000 об/мин происходит за 0,2с)
#2.21=0.2        // Темп торможения (уменьшение скорости на 1000 об/мин происходит за 0,2с)
#3.23=1          // Разрешение корректирующей связи
t_1%=200         // Время начала разгона
t_2%=700         // Время от начала разгона до начала торможения
t_3%=300         // Время от начала торможения до обнуления счетчика
t_4%=200         // Время от начала разгона до наброса нагрузки
t_5%=200         // Длительность воздействия нагрузки
L1=0.025         // Параметр НС
Kc=1.7           // Коэффициент усиления корректирующей связи
Ti=627.0         // Постоянная интегрирования модели объекта
Nmax%=2147483647 // Максимальная величина целочисленной переменной (2^31-1)
W_cn=0           // Нулевое начальное значение интегратора
}                // Закрытие раздела «Initial»
```

```

Clock{
M_cn=(#3.2-W_cn)*L1 // Открытие раздела «Clock»
M_jn=(#4.2-M_cn) // Вычисление
W_cn=(ITerm((#4.2-M_jn)*Ti), 1000, Nmax%, 0, 0, 0, 0)) // наблюдателя
#3.22=M_cn*Kc // состояния,
load%=0 // Формирование корректирующей связи
if ((#6.30=1)or(#6.32=1)) then // Запрет на формирование нагрузки
  if ((t%>t_1%)and(t%<t_1%+t_2%)) then // Проверка разрешения на вращение
    #1.18=1500 // t_1%<t%<t_1%+t_2%
    if ((t%>t_1%+t_4%)and(t%<t_1%+t_4%+t_5%)) then // Задание на скорость =1500
      load%=1 // t_1%+t_4%<t%<t_1%+t_4%+t_5%
    endif // Разрешение формирование нагрузки
  endif
  else // Задание на скорость =0
    #1.18=0
  endif
  if (t%>t_1%+t_2%+t_3%) then // t%>t_1%+t_2%+t_3%
    t%=0 // Обнуление переменной t%
  endif
  t%=t%+1 // Приращение переменной t%
else // Обнуление переменной t%
  t%=0
endif
#18.11=#4.2*1000 // Занесение в память сигналов скорости,
#18.12=M_cn*1000 // идентифицированного статического момента,
#18.13=#3.2*4 // полного тока двигателя
} // Заккрытие раздела «Clock»
Background{ // Открытие раздела «Background»
top: // Метка
if (CheckNode(11)=1) then // Проверка наличия устройства в сети
  if (load%=1) then
    status%=WRNET(11,4,8,45,0,100) // Пересылка данных по сети
  else
    status%=WRNET(11,4,8,0,0,100) // Пересылка данных по сети
  endif
endif
goto top: // Оператор безусловного перехода
} // Заккрытие раздела «Background»

```

Рисунок 4 – Программа управления

На рис. 5, 6 представлены графики переходных процессов полученных в такой системе. Переходные процессы в системе были получены при использовании программы *ST-Score*.

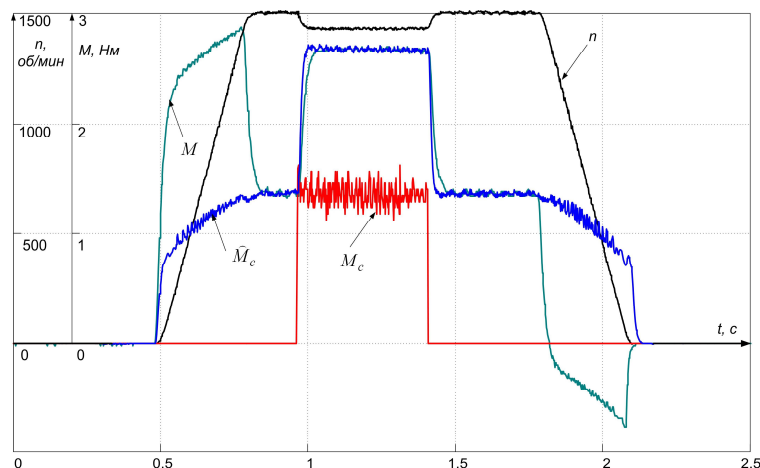


Рисунок 5 – Графики переходных процессов в системе с разомкнутой корректирующей связью по идентифицированному статическому моменту

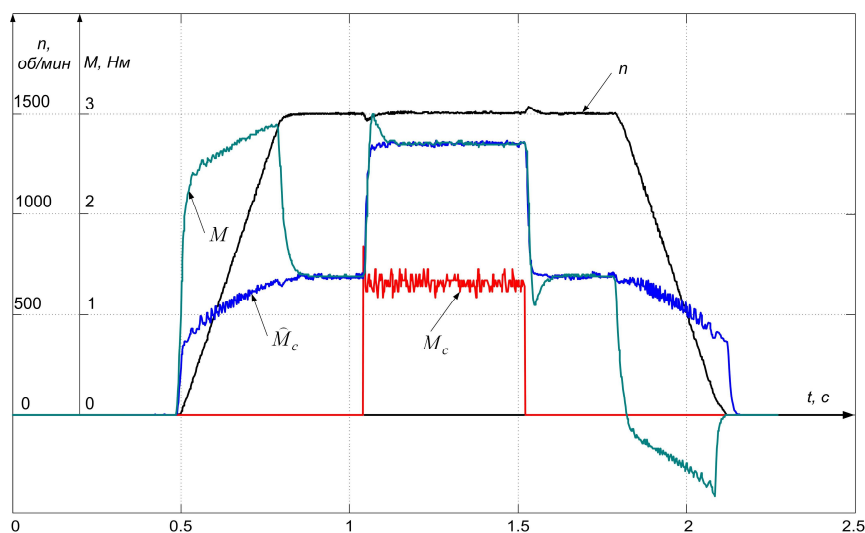


Рисунок 6 – Графіки переходних процесів в системі з замкнутою коректуючою зв'язкою по ідентифікованому статичному моменту

Таким образом, показана возможность изменения базовой структуры системы управления комплектным электроприводом с целью улучшения качества регулирования или расширения её функциональных возможностей.

ЛИТЕРАТУРА

1. Универсальный привод переменного тока для асинхронных двигателей и сервомоторов: руководство пользователя Unidrive SP. Редакция 10. – 2005. – 263 с.
2. Универсальный привод переменного тока для асинхронных двигателей и сервомоторов: расширенное руководство пользователя Unidrive SP. Редакция 7. – 2004. – 381 с.
3. Приводы постоянного тока с выходным током от 25А до 1850А: руководство пользователя Mentor II. – 2000. – №10. – 156 с.
4. Дополнительный модуль для Unidrive SP. Редакция 4: Руководство пользователя SM-Applications. – 2004. – 113 с.
5. Руководство пользователя CTNet. Редакция 6. – 2005. – 69 с.
6. Система подчиненного регулирования скорости с наблюдателем динамического и статического токов первого порядка / Коцегуб П.Х., Толочко О.И., Мариничев В.Ю., Розкаряка П.И. // Проблемы создания новых машин и технологий. Научные труды КГПУ. – 2001. – №1(10). – С. 103-109.

Надійшла до редколегії 19.09.2010

Рецензент: Є.Б.Ковальов

О.І. ТОЛОЧКО, П.І. РОЗКАРЯКА, М.М. ГОРОБЕЦЬ
Донецький національний технічний університет

O. TOLOCHKO, P. ROZKARYAKA, N. GOROBETS
Donetsk National Technical University

До питання про зміну типових структур цифрових систем управління комплектними електроприводами. Розглянуто питання зміни типових структур цифрових систем управління в сучасних комплектних електроприводах на прикладі системи векторного керування синхронним двигуном з постійними магнітами фірми Control Techniques. Запроектована система регулювання швидкості доповнена спостерігачем стану, що ідентифікує момент статичного опору, який використовується для компенсації впливу навантаження на швидкість приводу в усталеному режимі.

On the Issue of Changing Routine Structures of Digital Control Systems of Complete Electric Drive. The issues of changing routine structures of digital control systems in modern complete electrical drives for a system of vector control of synchronous motors with permanent magnets by Control Techniques is considered. A speed control system supplemented by state observer that identifies the load torque, which is used to compensate for the effect of load on the drive speed in a steady state, has been designed.