Емельянов А.А., Медведев А.В., Кобзев А.В., Кобзев А.В.
ФГАОУ ВПО «Российский государственный профессионально-педагогический университет», Екатеринбург

ПРОГРАММИРОВАНИЕ КОНТУРА ТОКА СО ЗВЕНОМ КОМПЕНСАЦИИ ЭДС В DELPHI

Для моделирования структурных схем используются специализированные программные пакеты, например, Matlab-Simulink и Vissim. С помощью стандартных готовых блоков можно моделировать достаточно сложные объекты, но, к сожалению, с увеличением их числа и тем более структурных преобразований с ними увеличивается вероятность сбоев в решении задач (алгебраические контуры, деление на нуль и т.д.). В поисках решения иногда приходится переходить от структурных схем с известными передаточными функциями к дифференциальным уравнениям и к непосредственному программированию их в Delphi. В пакетах учебных программ полезно иметь примеры таких переходов.

В качестве примера рассмотрим структурную схему системы автоматического регулирования тока якоря со звеном компенсации ЭДС, приведенной на рис.1 [1].

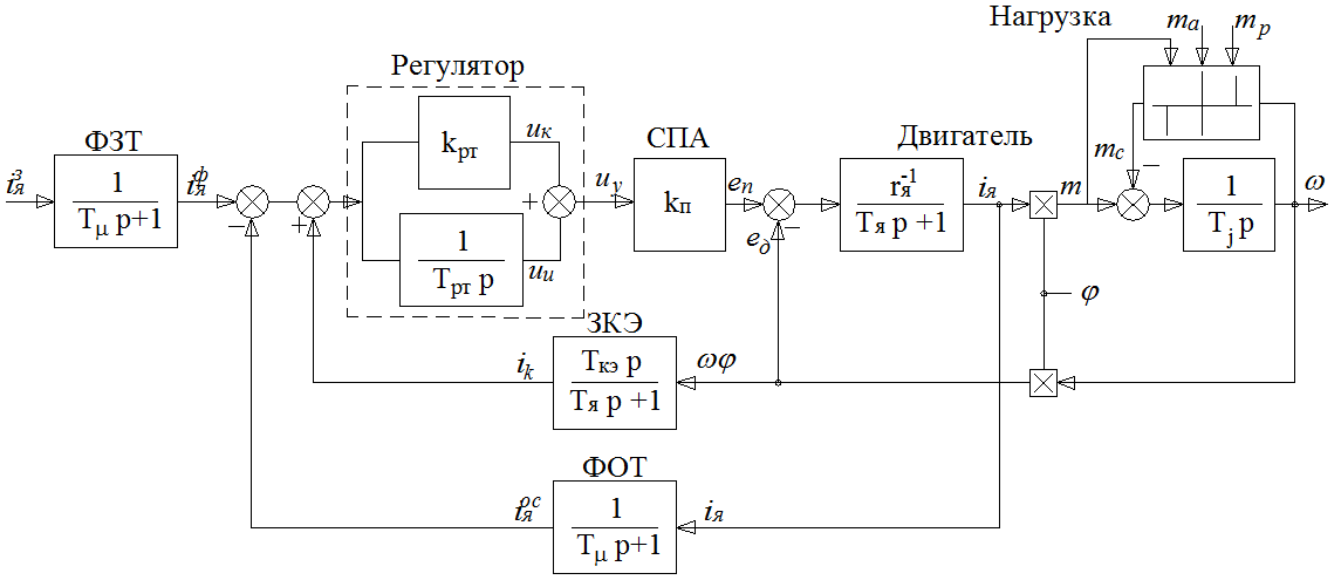

Рис.1. Структурная схема САР тока со звеном компенсации ЭДС

Как известно, для решения сложной системы необходимо разбить ее на несколько элементарных звеньев. Для них запишем входные и выходные

переменные, причем последние определяются произведением входной величины на передаточную функцию.

- **Звено ФЗТ** (фильтр в цепи задания тока) математически можно описать следующим образом:

$$\frac{i_я^з}{T_\mu p+1}=i_я^ф.$$

Тогда: $i_я^з=i_я^ф T_\mu p+i_я^ф$.

После этого можно записать дифференциальное уравнение для звена ФЗТ:

$$\frac{di_я^ф}{dt}=\frac{i_я^з-\quad}{T_\mu}.$$

- **Звено ФОТ** (фильтр в цепи обратной связи по току):

$$\frac{i_я}{T_\mu p+1}=i_я^{ос}.$$

Тогда: $i_я=i_я^{ос}T_\mu p+i_я^{ос}$.

После этого можно записать дифференциальное уравнение для звена ФОТ:

$$\frac{di_я^{ос}}{dt}=\frac{i_я-i_я^{ос}}{T_\mu}.$$

- **Звено компенсации ЭДС** (ЗКЭ):

$$\frac{\omega\varphi}{T_я p+1}=i_k.$$

где $T_{кэ}=\frac{2T_\mu}{r_я}$. Тогда $\frac{2T_\mu\varphi}{r_я}=i_k T_я p+i_k$.

Преобразуем: $\frac{2T_\mu\varphi}{r_я}=T_я p i_k$, $\frac{2T_\mu\varphi}{T_я r_я}=p i_k$.

Так как $p\omega=\frac{d\omega}{dt}=\frac{i_я\varphi}{T_j}$, то окончательно уравнение примет вид:

$$\frac{di_k}{dt}=\frac{2T_\mu\varphi\quad\varphi}{T_я r_я T_j}.$$

- **Регулятор тока**. Передаточная функция регулятора тока остается прежней, изменяется только входная переменная, тогда:

$$\frac{du_u}{dt} = \frac{i_я^ф - i_я^{ос} + i_k}{T_{рт}}.$$

Электрическая часть двигателя (Двигатель):

$$i_я = \frac{e_п - \quad \cdot \quad}{T_я p + 1}.$$

где $e_д = \omega\varphi$, $e_п = k_п \left[\qquad \right]$

После преобразования:

$$T_я p i_я = \frac{k_п \left[\qquad \right]}{r_я}.$$

После несложных преобразований получаем дифференциальное уравнение:

$$\frac{di_я}{dt} = \frac{k_п \left[\qquad \right]}{T_я r_я}$$

В итоге получаем следующую систему дифференциальных уравнений:

$$\begin{cases} \dfrac{di_я^ф}{dt} = \dfrac{i_я^з - i_я^ф}{T_\mu}, \\ \dfrac{di_я^{ос}}{dt} = \dfrac{i_я - i_я^{ос}}{T_\mu}, \\ \dfrac{di_k}{dt} = \dfrac{2T_\mu\varphi \quad \varphi \quad - \quad}{T_я r_я T_j}, \\ \dfrac{du_u}{dt} = \dfrac{i_я^ф - i_я^{ос} + i_k}{T_{рт}}, \\ \dfrac{di_я}{dt} = \dfrac{k_п \left[\qquad \right]}{T_я r_я} \\ \dfrac{d\omega}{dt} = \dfrac{i_я\varphi \quad}{T_j}. \end{cases} \qquad (1)$$

Решение системы дифференциальных уравнений (1) произведем на языке программирования высокого уровня Delphi **модифицированным методом Эйлера** (метод Рунге-Кутты 2-го порядка). Для реализации поставленной задачи запишем вышеуказанные уравнения как функции в разделе **private**:

```
function diaf(iaf,ioc,ik,uu,ia,v,t:real):real;
```

```
function dioc(iaf,ioc,ik,uu,ia,v,t:real):real;
function dik(iaf,ioc,ik,uu,ia,v,t:real):real;
function duu(iaf,ioc,ik,uu,ia,v,t:real):real;
function dia(iaf,ioc,ik,uu,ia,v,t:real):real;
function dv(iaf,ioc,ik,uu,ia,v,t:real):real;
function Mc(v,t:real):real;
```

После нажатия на сочетание клавиш Ctrl+Shift+C получим заготовки, которые компилятор создаст сам. В эти заготовки запишем уравнения:

```
function TMainForm.dia(iaf,ioc,ik,uu,ia,v,t:real): real;
 begin
  dia:=(kp*((iaf-ioc+ik)*kt+uu)-v*phi-ra*ia)/(ra*Ta);
 end;
function TMainForm.diaf(iaf,ioc,ik,uu,ia,v,t:real): real;
 begin
  diaf:=(iaz-iaf)/Tmu;
 end;
function TMainForm.dioc(iaf,ioc,ik,uu,ia,v,t:real): real;
 begin
  dioc:=(ia-ioc)/Tmu;
 end;
function TMainForm.duu(iaf,ioc,ik,uu,ia,v,t:real): real;
 begin
  duu:=(iaf-ioc+ik)/Tt;
 end;
function TMainForm.dv(iaf,ioc,ik,uu,ia,v,t:real):real;
 begin
  dv:=(ia*phi-Mc(v,t))/Tj;
 end;
function TMainForm.dik(iaf,ioc,ik,uu,ia,v,t:real): real;
 begin
  dik:=(2*Tmu*phi*(ia*phi-Mc(v,t))/(ra*Tj)-ik)/Ta;
 end;
```

Создадим раздел констант между разделами **type** и **var** с постоянными параметрами:

```
const
 Tj=0.423;   // Данные двигателя Д31: 6,8 кВт, 220 В, 37 А,
```

```
Ta=0.034;   // 880 об/мин.
ra=0.107;
phi=1; kp=1.393; Tmu=0.01; kt=ra*Ta/(2*kp*Tmu); Tt=2*kp*Tmu/ra;
```

В разделе **var** опишем глобальные переменные:

```
var
  MainForm: TmainForm; iaz,kt,Tt,M,ep,ea,ik :Real;
```

Поместим на форму 2 компонента TChart из вкладки Additional и компонент Button из вкладки Standart. Щелкнув два раза на каждом компоненте TChart левой кнопкой мыши, появится окно, в котором на вкладке Series нажимаем на кнопку Add. Далее выбираем тип графика FastLine, убираем галочку 3D и нажимаем ОК. На вкладке Legend убираем галочку напротив Visible и нажимаем Close. Перейдем на вкладку Events в окне Object Inspector, предварительно выделив кнопку.

Щелкнув два раза по позиции OnClick будет автоматически создана процедура по нажатии данной кнопки:

```
procedure TMainForm.Button1Click(Sender: TObject);
begin
end;
```

Опишем переменные необходимые только для данной процедуры. Данный раздел необходимо описать между строками «**procedure** TMainForm.Button1Click(Sender: Tobject);» и «**begin**»:

```
var
dv0,dia0,t0,duu0,dioc0,diaf0,dik0,dv1,dia1,duu1,dioc1,diaf1,dik1,
dt:Real; i:Integer;
```

Зададим начальные условия:

```
diaf0:=0; dioc0:=0; dik0:=0; duu0:=0; dia0:=0; dv0:=0; t0:=0; iaz:=0.1;
```

Назначим шаг интегрирования:

```
dt:=0.0005;
```

Далее зададим цикл:

```
 i:=0;
  while i<400 do
   begin
   end;
```

В данном цикле опишем процедуру расчета системы дифференциальных уравнений модифицированным методом Эйлера (методом Рунге-Кутты 2-го порядка).

Модифицированный метод Эйлера (метод Рунге-Кутты 2-го порядка) описывается следующим образом:

$$y_{i+} = \quad + \quad + \quad + \quad + \quad \cdot \quad \cdot \quad \cdot$$

$$x_{i+} = \quad +$$

Тогда:

```
while i<400 do
  begin
{M}  M:=dia0*phi;
 // diaf          Method of Runge-Kutta 2
   diaf1:=diaf0+(diaf(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)+
   diaf(diaf0+diaf(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt,
   dioc0,dik0,duu0,dia0,dv0+dt,t0))*0.5*dt;
 // dioc
   dioc1:=dioc0+(dioc(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)+
   dioc(diaf0,dioc0+dioc(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt,
   dik0,duu0,dia0,dv0+dt,t0))*0.5*dt;
 // dik
   dik1:=dik0+(dik(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)+dik(diaf0,
   dioc0,dik0+dik(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt,duu0,
   dia0,dv0,t0+dt))*0.5*dt;
 // duu
   duu1:=duu0+(duu(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)+duu(diaf0,
   dioc0,dik0,duu0+duu(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt,
   dia0,dv0,t0+dt))*0.5*dt;
 // dia
   dia1:=dia0+(dia(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)+dia(diaf0,
   dioc0,dik0,duu0,dia0+dia(diaf0,dioc0,dik0,duu0,dia0,dv0,
   t0)*dt,dv0,t0+dt))*0.5*dt;
 // dv
   dv1:=dv0+(dv(diaf0,dioc0,dik0,duu0,dia0,dv0,t0)+dv(diaf0,
   dioc0,dik0,duu0,dia0,dv0+dv(diaf0,dioc0,dik0,duu0,dia0,
   dv0,t0)*dt,t0+dt))*0.5*dt;
    Series1.AddXY(t0,dia0);     // Ток
    Series3.AddXY(t0,iaz);      // Ток задания
    Series2.AddXY(t0,dv0);      // Скорость
    Inc(i);
```

```
    t0:=t0+dt; diaf0:=diaf1; dioc0:=dioc1; dik0:=dik1;
    duu0:=duu1; dia0:=dia1; dv0:=dv1;
  end;
```

После нажатия на кнопку Run (F9) появится окно программы, нажимаем на кнопку и получаем следующие результаты:

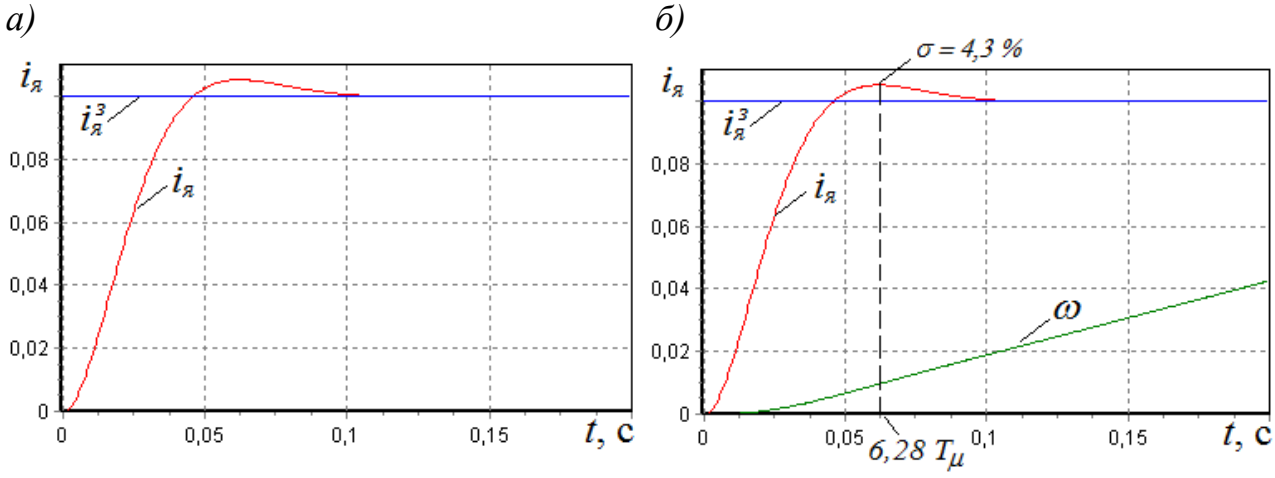

Рис.2. Реакция САР тока со звеном компенсации ЭДС на скачок задания: а - при заторможенном, б - расторможенном якоре

Проанализировав полученные результаты, можно сделать вывод, что данные графики соответствуют общепринятым теоретическим положениям, приведенных в [1].

Таким образом, в данной работе показана методика перехода от стандартных структурных схем к дифференциальным уравнениям на примере контура тока со звеном компенсации ЭДС, которые, в последующем, решены с помощью языка программирования высокого уровня Delphi модифицированным методом Эйлера (метод Рунге-Кутты второго порядка).

Литература

1. *Шрейнер Р.Т.* Системы подчиненного регулирования электроприводов. - Екатеринбург: Изд-во ГОУ ВПО «Рос. гос. проф.-пед. ун-т», 2008. - 279 с.

2. *Шрейнер Р.Т.* Моделирование моментов нагрузки электродвигателей в MATLAB. [Текст] / Шрейнер Р.Т., Емельянов А.А., Клишин А.В., Медведев А.В. Молодой учёный. 2010. № 8(19). -с. 6-12.

3. *Архангельский А.Я.* Программирование в Delphi для Windows. Версии 2006, 2007, Turbo Delphi. – М.: ООО «Бином-Пресс», 2007.– 1248 с.