

## ПРОГРАММИРОВАНИЕ КОНТУРА СКОРОСТИ В DELPHI

В пакете учебных программ необходимо иметь примеры программирования в Delphi многоконтурных систем подчиненного регулирования электроприводов. В данной статье рассмотрен контур скорости с «ПИ» регулятором.

Двукратная (ДК) САР скорости образуется путем дополнения однократной (ОК) САР еще одним контуром регулирования скорости. ДК САР скорости имеет два регулятора скорости: внутренний пропорциональный регулятор и внешний – интегральный.

В результате эквивалентных структурных преобразований получим двукратную систему автоматического регулирования скорости с «ПИ» регулятором скорости и апериодически фильтром в цепи задания скорости [1].

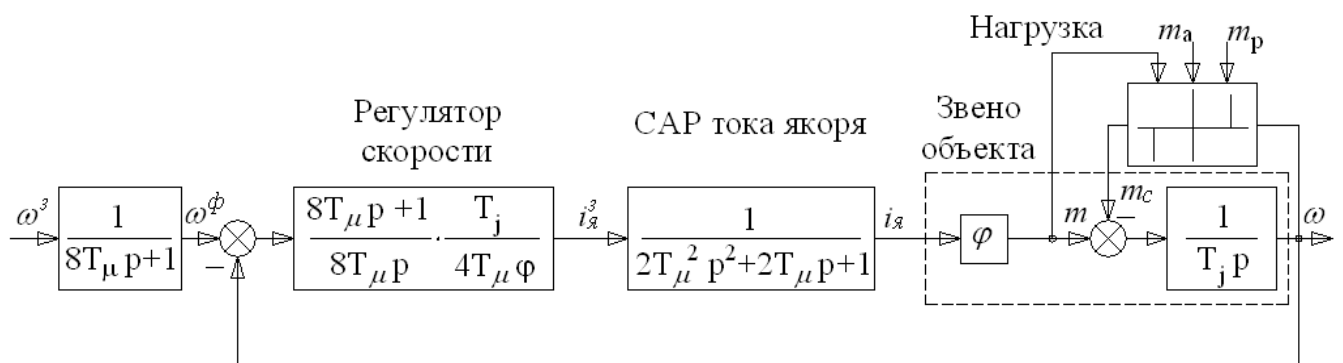


Рис.1. Структурная схема двукратной САР скорости

По структурной схеме, изображенной на рис.3 можно записать следующую систему дифференциальных уравнений, используя методику из [2]:

$$\left\{ \begin{array}{l} \frac{d\omega}{dt} = \delta I_{\mu}, \\ \frac{di_u}{dt} = I_{pc}, \\ \frac{di_{\alpha}^{\phi}}{dt} = \left[ \dots \right], \\ \frac{di_{\alpha}^{oc}}{dt} = \frac{i_{\alpha} - i_{\alpha}^{oc}}{T_{\mu}}, \\ \frac{di_k}{dt} = \frac{2T_{\mu}\phi - \phi}{T_{\alpha}r_{\alpha}T_j}, \\ \frac{du_u}{dt} = \frac{i_{\alpha}^{\phi} - i_{\alpha}^{oc} + i_k}{T_{pm}}, \\ \frac{di_{\alpha}}{dt} = \frac{k_n \left[ \dots \right]}{T_{\alpha}r_{\alpha}}, \\ \frac{d\omega}{dt} = \frac{i_{\alpha}\phi}{T_j}. \end{array} \right.$$

Решение данной системы дифференциальных уравнений произведем на языке программирования высокого уровня Delphi **методом Рунге-Кутты 4-го порядка**. Для реализации поставленной задачи запишем вышеуказанные уравнения как функции в разделе **private**:

```

function dvf(vf,iu,iaf,ioc,ik,uu,ia,v,t:real):real;
function diu(vf,iu,iaf,ioc,ik,uu,ia,v,t:real):real;
function diaf(vf,iu,iaf,ioc,ik,uu,ia,v,t:real):real;
function dioc(vf,iu,iaf,ioc,ik,uu,ia,v,t:real):real;
function dik(vf,iu,iaf,ioc,ik,uu,ia,v,t:real):real;
function duu(vf,iu,iaf,ioc,ik,uu,ia,v,t:real):real;
function dia(vf,iu,iaf,ioc,ik,uu,ia,v,t:real):real;
function dv(vf,iu,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(vf,iu,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(vf,iu,iaf,ioc,ik,uu,ia,v,t: real): real;
begin
    diaf:=((kpc*(vf-v)+iu)-iaf)/Tmu;
end;
function TMainForm.dik(vf,iu,iaf,ioc,ik,uu,ia,v,t: real): real;
begin
    dik:=(2*Tmu*phi*(ia*phi-Mc(v,t))/(ra*Tj)-ik)/Ta;
end;
function TMainForm.dioc(vf,iu,iaf,ioc,ik,uu,ia,v,t: real): real;
begin
    dioc:=(ia-ioc)/Tmu;
end;
function TMainForm.diu(vf,iu,iaf,ioc,ik,uu,ia,v,t: real): real;
begin
    diu:=(vf-v)/Tpc;
end;
function TMainForm.duu(vf,iu,iaf,ioc,ik,uu,ia,v,t: real): real;
begin
    duu:=(iaf-ioc+ik)/Tt;
end;
function TMainForm.dv(vf,iu,iaf,ioc,ik,uu,ia,v,t: real): real;
begin
    dv:=(ia*phi-Mc(v,t))/Tj;
end;
function TMainForm.dvf(vf,iu,iaf,ioc,ik,uu,ia,v,t: real): real;
begin
    dvf:=(vzad-vf)/(8*Tmu);
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;

```

$\phi=1$ ;  $k_p=1.393$ ;  $T_{mu}=0.01$ ;  $kt=r_a * T_a / (2 * k_p * T_{mu})$ ;  $T_t=2 * k_p * T_{mu} / r_a$ ;

В разделе **var** опишем глобальные переменные:

**var**

MainForm: TmainForm; iaz,kt,Tt,M,er,ea,ik,vzad,kpc,Tpc,: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 нажатию 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**

k1iu,k2iu,k3iu,k4iu,k1vf,k2vf,k3vf,k4vf,k1ik,k2ik,k3ik,k4ik,k1ioc,k2ioc,k3ioc,k4ioc,k1iaf,k2iaf,k3iaf,k4iaf,k1uu,k2uu,k3uu,k4uu,  
k1uy,k2uy,k3uy,k4uy,k1ia,k2ia,k3ia,k4ia,k1v,k2v,k3v,k4v,dv0,dia0,  
t0,duu0,dioc0,diaf0,dik0,dvf0,diu0,dv1,dia1,duu1,dioc1,diaf1,dik1,dvf1,diu1  
,t,dt,k:Real;i:Integer;

Зададим начальные условия:

$dvf_0=0$ ;  $diu_0=0$ ;  $diaf_0=0$ ;  $dioc_0=0$ ;  $dik_0=0$ ;  $duu_0=0$ ;

$dia_0=0$ ;  $dv_0=0$ ;  $t_0=0$ ;  $vzad=0.1$ ;

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 **методом Рунге-Кутты 4-го порядка**. Данный метод описывается следующим образом:

$$\begin{aligned}
 k_1 &= \dots \cdot \\
 k_2 &= \dots + \dots \cdot \\
 k_3 &= \dots + \dots \cdot \\
 k_4 &= \dots + \dots \cdot \\
 y_1 &= \dots + \dots + \dots + \dots + \dots
 \end{aligned}$$

Тогда:

```

while i<900 do
  begin
    {M} M:=dia0*phi;
    // dvf
    k1vf:=dvf(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k2vf:=dvf(dvf0+0.5*k1vf,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k3vf:=dvf(dvf0+0.5*k2vf,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k4vf:=dvf(dvf0+k3vf,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    dvf1:=(k1vf+2*k2vf+2*k3vf+k4vf)/6;
    // diu
    k1iu:=diu(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k2iu:=diu(dvf0,diu0+0.5*k1iu,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k3iu:=diu(dvf0,diu0+0.5*k2iu,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k4iu:=diu(dvf0,diu0+k3iu,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    diu1:=(k1iu+2*k2iu+2*k3iu+k4iu)/6;
    // diaf
    k1iaf:=diaf(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k2iaf:=diaf(dvf0,diu0,diaf0+0.5*k1iaf,dioc0,dik0,duu0,dia0,dv0,
    t0)*dt;
    k3iaf:=diaf(dvf0,diu0,diaf0+0.5*k2iaf,dioc0,dik0,duu0,dia0,dv0,
    t0)*dt;
    k4iaf:=diaf(dvf0,diu0,diaf0+k3iaf,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    diaf1:=(k1iaf+2*k2iaf+2*k3iaf+k4iaf)/6;
    // dioc
    k1ioc:=dioc(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
    k2ioc:=dioc(dvf0,diu0,diaf0,dioc0+0.5*k1ioc,dik0,duu0,dia0,dv0,
    t0)*dt;
    k3ioc:=dioc(dvf0,diu0,diaf0,dioc0+0.5*k2ioc,dik0,duu0,dia0,dv0,
    t0)*dt;
    k4ioc:=dioc(dvf0,diu0,diaf0,dioc0+k3ioc,dik0,duu0,dia0,dv0,t0)*dt;
  
```

```

dioc1:=dioc0+(k1ioc+2*k2ioc+2*k3ioc+k4ioc)/6;
// dik
k1ik:=dik(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
k2ik:=dik(dvf0,diu0,diaf0,dioc0,dik0+0.5*k1ik,duu0,dia0,dv0,
t0)*dt; k3ik:=dik(dvf0,diu0,diaf0,dioc0,dik0+0.5*k2ik,duu0,dia0,dv0,
t0)*dt; k4ik:=dik(dvf0,diu0,diaf0,dioc0,dik0+k3ik,duu0,dia0,dv0,t0)*dt;
dik1:=dik0+(k1ik+2*k2ik+2*k3ik+k4ik)/6;
// duu
k1uu:=duu(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
k2uu:=duu(dvf0,diu0,diaf0,dioc0,dik0,duu0+0.5*k1uu,dia0,dv0,
t0)*dt; k3uu:=duu(dvf0,diu0,diaf0,dioc0,dik0,duu0+0.5*k2uu,dia0,dv0,
t0)*dt;
k4uu:=duu(dvf0,diu0,diaf0,dioc0,dik0,duu0+k3uu,dia0,dv0,t0)*dt;
duu1:=duu0+(k1uu+2*k2uu+2*k3uu+k4uu)/6;
// dia
k1ia:=dia(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
k2ia:=dia(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0+0.5*k1ia,dv0,
t0)*dt; k3ia:=dia(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0+0.5*k2ia,dv0,
t0)*dt; k4ia:=dia(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0+k3ia,dv0,t0)*dt;
dia1:=dia0+(k1ia+2*k2ia+2*k3ia+k4ia)/6;
// dv
k1v:=dv(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0,t0)*dt;
k2v:=dv(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0+0.5*k1v,t0)*dt;
k3v:=dv(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0+0.5*k2v,t0)*dt;
k4v:=dv(dvf0,diu0,diaf0,dioc0,dik0,duu0,dia0,dv0+k3v,t0)*dt;
dv1:=dv0+(k1v+2*k2v+2*k3v+k4v)/6;
Series1.AddXY(t0,dia1); // Ток
Series2.AddXY(t0,dv0); // Скорость
Inc(i); t0:=t0+dt; dvf0:=dvf1; diu0:=diu1; 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:

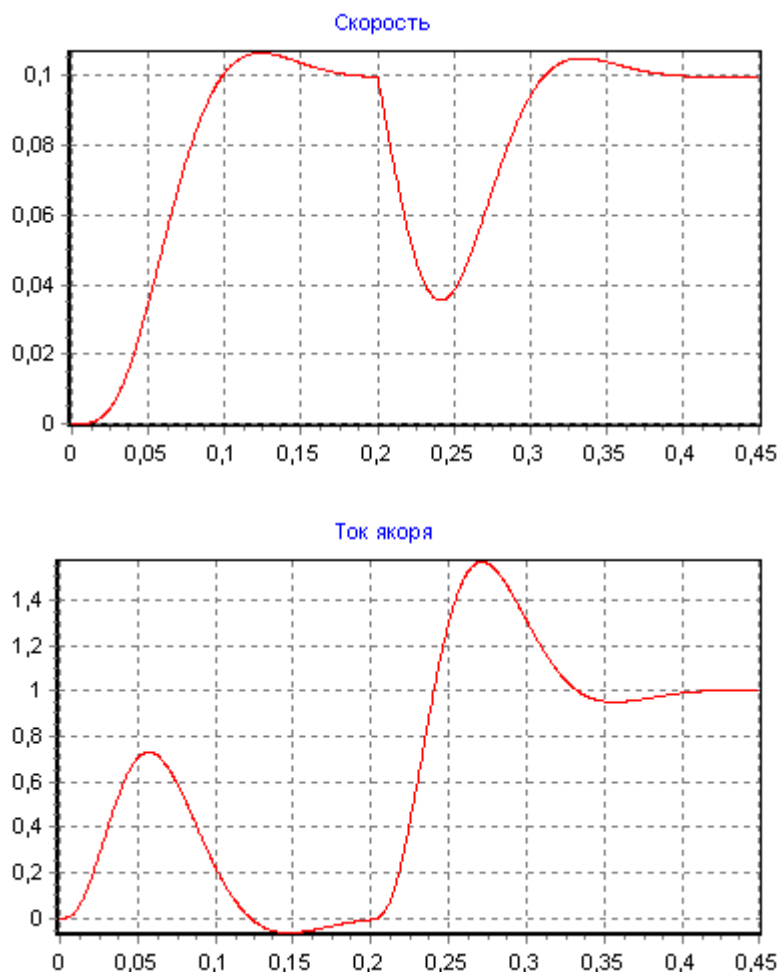


Рис. 2. Временные диаграммы скорости и тока якоря в ДК САР скорости

Полученные результаты соответствуют общепринятым теоритическим положениям, приведенным в [1].

Таким образом, в данной работе показана методика перехода от стандартных структурных схем к дифференциальным уравнениям, которые, в последующем, решены с помощью языка программирования высокого уровня Delphi методом Рунге-Кутты четвертого порядка, на примере контура скорости.

#### *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