

Приложение 3

Определение параметров синхронного электродвигателя по каталожным данным

```
% Исходные данные
% Каталожные параметры синхронного двигателя
Unom = 6; % номинальное напряжение, кВ
Pnom = 8000; % номинальная активная мощность, кВт
etan = 0.979; % КПД
cosfn = 0.9; % косинус фи
nnom = 3000; % номинальная частота вращения, об/мин
% Кратность по отношению к номинальному моменту:
Mn = 1.76; % пускового момента, о.е.
Mmax = 2.29; % максимального синхронного момента, о.е.
Mvh = 1.83; % входного момента (при s=-0.05), о.е.
In = 6.93; % кратность пускового тока, о.е.
Tr = 3.5; % постоянная времени обмотки возбуждения Td0, с

% Тип двигателя: 0 - явнополюсный, 1 - неявнополюсный
if (nnom==3000)
    model=1;
else
    model=0;
end
Snom = Pnom/(cosfn*etan); % номинальная полная мощность, кВА
Inom = Snom/(sqrt(3)*Unom); % номинальный ток, А
Mnom = Pnom*(9549.3/nnom); % номинальный момент, Нм

% Расчет параметров модели
% 1. Активное сопротивление статора
% r = 0;
% kpdn = etan;
r = 0.25*(1-etan)*cosfn;
kpdn = 0.25 + 0.75*etan;

% 2. Сверхпереходное индуктивное сопротивление по продольной
% и поперечной осям
cosfP = Mn*(cosfn*etan/In);
sinfP = sqrt(1-cosfP^2);
alphaq = 1.1;
zd2h = sqrt((1+alphaq)/(alphaq^2*In^2));
xd2h = zd2h * sinfP;
r2h = zd2h * cosfP;
xq2h = 1.1*xd2h;

% 3. Реактивность рассеяния статора
xs = 0.55*xd2h;

% 4. Переходное сопротивление по продольной оси
xdh = 1.6*xd2h;

% 5. Синхронные индуктивные сопротивления по продольной и
% поперечной осям
if model==1
    deltanom = asin(1/Mmax); % для неявнополюсных СД
else
    deltanom = asin(0.88/Mmax); % для явнополюсных СД
```

```
end
Uknom = complex(cos(deltanom), sin(deltanom));
Pnom = -cosfn;
Qnom = -sin(acos(cosfn)); % двигатель перевозбужден
Sknom = complex(Pnom, Qnom);
Iknom = Sknom/conj(Uknom);
Zs = r + xs*i;
Edelta = Uknom + Iknom*Zs;
Pmax = Mmax*etan*cosfn;
tetn = 1.0; % параметр насыщения по оси d
if model==1
    xad =(real(Edelta)/tetn-Pmax*xs)/(Pmax + imag(Iknom));
else
    xad =((0.965*real(Edelta)/tetn)-
        Pmax*xs+0.167)/(Pmax+0.965*imag(Iknom));
end
xd = xad + xs;
if model==1
    xq = xd;
else
    xq = 0.6*xd;
end
xaq = xq - xs;

% 6. Параметры обмотки возбуждения, если известна Tr
omegas = 314.159;
xr = xad^2/(xd-xdh);
rr = xr/(omegas*Tr);
xsr = xr-xad;
% Постоянная времени обмотки возбуждения Tr, если не задана,
% вычисляется по формулам
% Eqn = real(Edelta)/tetn - xad*imag(Iknom);
% rr = 0.667*(Ufn*Ifn*etan*cosfn/Pnom)*(tetn*xad/Eqn)^2;
% Tr = xr/(omegas*rr);

% 7. Параметры эквивалентных роторных контуров в начале пуска
% при s=-1 в качестве нулевого приближения
xsrđ = 1/(1/(xd2h-xs) - 1/xad - 1/xsr);
xsrq = 1/(1/(xq2h-xs)-1/xaq);
rrđ = r2h;
if model==1
    rrq = rrđ;
else
    rrq = 0.75*rrđ;
end

% 8. Для воспроизведения каталожных значений In, Mn уточняются
% значения параметров демпферных контуров методом итераций
s = -1;
for j=1:100
    [Macp,Mar,Mard,Marrd,Marq,I] = fMI(r, xs, xad, xaq, rr,xsr,
    rrđ,xsrđ, rrq,xsrq, kpdn, s);
    dMa = Mn - Macp;
    dI = In - I;
    rrđ = rrđ *(1 + dMa/Mn);
    xsrđ = xsrđ*(1 - dMa/Mn);
    xsrđ = xsrđ*(1 - dI/In);
    xsrq = xsrq*(1 - dI/In);
    [Macp,Mar,Mard,Marrd,Marq,I] = fMI(r, xs, xad, xaq, rr,xsr,
    rrđ,xsrđ, rrq,xsrq, kpdn, s);
```

```
dMa = Mn - Macp;  
dI = In - I;  
if abs(dI) < 1e-3 & abs(dMa) < 1e-3  
    break;  
end  
rrq = rrq * (1 + dMa/Mn);  
xsrq = xsrq * (1 - dMa/Mn);  
xsrD = xsrD * (1 - dI/In);  
xsrq = xsrq * (1 - dI/In);  
end  
if J==100  
    disp('Неуспешный подбор для s=-1');  
else  
    msg = sprintf('J_1 = %d', J);  
    disp(msg);  
end  
% Параметры роторных контуров в начале пуска  
rrdP = rrd;  
rrqP = rrq;  
xsrDp = xsrD;  
xsrqP = xsrq;  
  
% 9. Параметры демпферных контуров для подсинхронной  
% частоты вращения оцениваются из условия, что при s=svх=-0.05  
% двигатель развивает входной асинхронный момент Mвх.  
svh = -0.05;  
for J=1:100  
    [Macp, Mar, Mard, Marrd, Marq, I] = fMI(r, xs, xad, хаq, rr, xsr,  
rrd, xsrD, rrq, xsrq, kpdn, svh);  
    if abs(Mvh-Macp) > 1e-3  
        rrd = rrd * (1 - (Mvh - Macp) / Mvh);  
        rrq = rrq * (1 - (Mvh - Macp) / Mvh);  
        xsrD = xsrD * (1 + (1 - 0.5 * (Mvh - Macp) / Mvh)) / 2;  
        xsrq = xsrq * (1 + (1 - 0.5 * (Mvh - Macp) / Mvh)) / 2;  
    else  
        break;  
    end  
end  
if J==100  
    disp('Неуспешный подбор для s=-0,05');  
else  
    msg = sprintf('J_05 = %d', J);  
    disp(msg);  
end  
rrdvh = rrd;  
rrqvh = rrq;  
xsrDvh = xsrD;  
xsrqvh = xsrq;  
  
% 10. Коэффициенты функций изменения параметров эквивалентных  
% роторных контуров  
s1 = -1;  
fvh = 0.0;  
spr = -0.35;  
fpr = 0.54;  
a2 = (1 + fpr - 2 * fvh - ((1 - fpr) / (s1 - spr)) * (s1 + spr -  
2 * svh)) / ((s1^2 + spr^2 - 2 * svh^2) - (s1^2 - spr^2) * (s1 + spr - 2 * svh) / (s1 -  
spr));  
a1 = (1 - fpr - a2 * (s1^2 - spr^2)) / (s1 - spr);  
a0 = 1 - a1 * s1 - a2 * (s1^2);
```

```
% функция расчета средних значений момента и тока  
function[Macr,Mar,MarD,Marrd,Marq,I] = fMI(r, xs, xad, xaq,  
rr,xsr, rrd,xsrd, rrq,xsrq, kpdn, s)
```

```
sv = 0;  
ssv = s - sv;  
ted = 1;  
teq = 1;  
xd = xad + xs;  
xq = xaq + xs;  
xr = xad + xsr;  
xrd = xad + xsrd;  
xrq = xaq + xsrq;  
tmp = ted*xad;  
cr = tmp/xr;  
ar = cr*xad;  
br = (tmp + xsr)/xr;  
ror = rr/xr;  
crd = tmp/xrd;  
ard = crd*xad;  
brd = (xsrd + tmp)/xrd;  
rord = rrd/xrd;  
tmp = teq*xaq;  
arq = tmp*xaq/xrq;  
brq = (xsrq + tmp)/xrq;  
rorq = rrq/xrq;  
  
T = complex(ssv^2*(ar*brd-cr*ard), -ssv*rord*ar);  
Yrrd = complex(ror*rord+(ssv)^2*(cr*crd-br*brd),  
ssv*(ror*brd+rord*br));  
Zr = T/Yrrd;  
T = complex(ssv^2*(ard*br-crd*ar), -ssv*ror*ard);  
Zrd = T/Yrrd;  
T = complex(0, ssv*arq);  
Yrq = complex(rorq, ssv*brq);  
Zrq = T/Yrq;
```

```
Ad = xs + ted*(xad + Zr + Zrd);  
Aq = xs + teq*(xaq - Zrq);  
K = r + i*(ssv*Ad);  
M = -(1+s)*Ad;  
L = (1+s)*Aq;  
N = r + i*ssv*Aq;  
D = K*N - M*L;  
Yd = (L - i*N)/D;  
Yq = (i*M - K)/D;
```

```
% Комплексные амплитуды переменных
```

```
U = 1+0i;  
Id = U*Yd;  
Iq = U*Yq;  
Eq = Zr*Id;
```

```
Erd = Zrd*Id;  
Erq = Zrq*Iq;
```

```
% коэффициенты частного решения с индексами 1 и 2
```

```
Id1 = real(Id);  
Id2 = -imag(Id);
```

```
Iq1 = real(Iq);
Iq2 = -imag(Iq);
Eq1 = real(Eq);
Eq2 = -imag(Eq);
Erq1 = real(Erq);
Erq2 = -imag(Erq);
Erd1 = real(Erd);
Erd2 = -imag(Erd);

I = sqrt(0.5*(Id1^2+Id2^2+Iq1^2+Iq2^2));
Mascp = - 0.5*(ted*(Eq1+Erq1)*Iq1 + ted*(Eq2+Erq2)*Iq2 +
teq*Erd1*Id1 + teq*Erd2*Id2 + ...
(ted*xad-teq*xaq)*(Id1*Iq1+Id2*Iq2))*kpdn;

% Момент от ОВ и роторного контура в оси d при разомкнутом
% демпферном контуре в оси q
Aq = xs + teq*(xaq);
K = r + i*(ssv*Ad);
M = -(1+s)*Ad;
L = (1+s)*Aq;
N = r + i*ssv*Aq;
D = K*N - M*L;
Yd = (L - i*N)/D;
Yq = (i*M - K)/D;
Id = U*Yd;
Iq = U*Yq;
Eq = Zr*Id;
Erq = Zrd*Id;
Id1 = real(Id);
Id2 = -imag(Id);
Iq1 = real(Iq);
Iq2 = -imag(Iq);
Eq1 = real(Eq);
Eq2 = -imag(Eq);
Erq1 = real(Erq);
Erq2 = -imag(Erq);
Marrrd = - 0.5*(ted*(Eq1+Erq1)*Iq1 + ted*(Eq2+Erq2)*Iq2 + ...
(ted*xad-teq*xaq)*(Id1*Iq1+Id2*Iq2))*kpdn;

% Момент от ОВ при прочих разомкнутых контурах
Zr = 0 - i*ssv*ar/(ror+i*ssv*br);
Ad = xs + ted*(xad + Zr);
K = r + i*(ssv*Ad);
M = -(1+s)*Ad;
L = (1+s)*Aq;
N = r + i*ssv*Aq;
D = K*N - M*L;
Yd = (L - i*N)/D;
Yq = (i*M - K)/D;

Id = U*Yd;
Iq = U*Yq;
Eq = Zr*Id;

Id1 = real(Id);
Id2 = -imag(Id);
Iq1 = real(Iq);
Iq2 = -imag(Iq);
Eq1 = real(Eq);
Eq2 = -imag(Eq);
```

```
Mar = - 0.5*(ted*(Eq1)*Iq1 + ted*(Eq2)*Iq2 + (ted*xad-  
teq*xaq)*(Id1*Iq1+Id2*Iq2))*kpdn;  
  
% Момент от роторного контура в оси d при прочих  
% разомкнутых контурах  
Zrd = 0 - i*ssv*ard/(rord+i*ssv*brd);  
Ad = xs + ted*(xad + Zrd);  
K = r + i*(ssv*Ad);  
M = -(1+s)*Ad;  
D = K*N - M*L;  
Yd = (L - i*N)/D;  
Yq = (i*M - K)/D;  
  
Id = U*Yd;  
Iq = U*Yq;  
Erq = Zrd*Id;  
  
Id1 = real(Id);  
Id2 = -imag(Id);  
Iq1 = real(Iq);  
Iq2 = -imag(Iq);  
Erq1 = real(Erq);  
Erq2 = -imag(Erq);  
Mard = - 0.5*(ted*(Erq1)*Iq1 + ted*(Erq2)*Iq2 + (ted*xad-  
teq*xaq)*(Id1*Iq1+Id2*Iq2))*kpdn;  
  
% Момент от роторного контура в оси q при прочих  
% разомкнутых контурах  
Ad = xs + ted*(xad);  
Aq = xs + teq*(xaq - Zrq);  
K = r + i*(ssv*Ad);  
M = -(1+s)*Ad;  
L = (1+s)*Aq;  
N = r + i*ssv*Aq;  
D = K*N - M*L;  
Yd = (L - i*N)/D;  
Yq = (i*M - K)/D;  
Id = U*Yd;  
Iq = U*Yq;  
Erd = Zrq*Iq;  
Id1 = real(Id);  
Id2 = -imag(Id);  
Iq1 = real(Iq);  
Iq2 = -imag(Iq);  
Erd1 = real(Erd);  
Erd2 = -imag(Erd);  
Marq = - 0.5*(teq*Erd1*Id1 + teq*Erd2*Id2 + (ted*xad-  
teq*xaq)*(Id1*Iq1+Id2*Iq2))*kpdn;
```