University of Ljubljana

J. Stefan Institute, Ljubljana, Slovenia

IJS - Report **DP-7091**

Measurements and mathematical modeling of a DC motor for the purpose of fault diagnosis

Damir Vrančić
Đani Juričić
Thomas Höfling*

November, 1994

* Institute of Automatic Control
Laboratory of Control Engineering and Process Automation
Technical University of Darmstadt
Germany

Table of contents

1. Introduction

Fault detection gains increasing attention in last two decades. Finding a process model turns to be still very important for implementing fault detection algorithms [1,2].

In this report a way of building up the grey-box model of DC motor is introduced. As a basis of building procedure were grey-box models made by Höfling [1] and Juričić [2]. The difference between those two models is that Höfling's approach is closer to white-box modelling and Juričić's model is closer to black-box one. This report tries to combine both approaches to achieve useful model of the DC motor that can be used in fault detection analysis. Main goal was to find a model which would satisfactorily describe behaviour of the DC motor through all of its working range.

DC motor consists of two sub-processes: electrical and mechanical. ***Electrical*** sub-process consists of armature inductance (L_a), armature resistance (R_a) and magnetic flux of the stator (Ψ). Armature current (I_a) is caused by armature voltage (U_a) on the coil (L_a). Because the motor is rotating, there is an opposite induced voltage on inductance proportional to the speed of the motor (ω) and magnetic flux (Ψ). Armature current through inductance is therefore:

$$I_a(t) = \frac{1}{L_a}\int_0^t \left(U_a(t) - R_a I_a(t) - \Psi\omega(t)\right)dt + I_a(0) \tag{1}$$

Second sub-process in motor is a ***mechanical*** one. It consists of inertia of the motor and a load (J). The difference in motor speed is caused by electromagnetic moment (M_{em}), load (M_l) and friction of the motor (M_f):

$$\omega(t) = \frac{1}{J}\int_0^t \left(M_{em} - M_l - M_f\right)dt + \omega(0), \tag{2}$$

where M_{em} is a function of armature current and M_f is a function of speed.

As a load (M_l), we used an electric brake. It has a non-linear characteristic between output torque (M_l) and input control voltage and is also speed-dependant. To have a constant torque through all the range of motor speed, we had to change input control voltage to electric brake during the experiment. All the measurements were taken in such a way that constant braking force is achieved.

2. Measurements and modeling

Static and dynamic measurements of the DC motor were performed. When measuring ***static*** characteristics of the motor, we changed armature voltage (U_a) from zero to full range (approx. 0V to 100V) in 200 seconds. Increasing of U_a is slow compared with electrical and mechanical[1] time constants, so measurement can be considered as static. Several measurements were taken, each with different load moment (M_l). It changed from 0 to 1.4 Nm in steps of 0.2 Nm. Maximum value of M_l=1.4 Nm is taken according to maximum power of the brake (400 W).

Measurements of armature current I_a and motor speed ω vs. armature voltage U_a are shown on Fig. 1 and 2.

Dynamic measurements were made by step changes of U_a through the period of 10s. There were measurements taken at the load moment of 0 Nm and 1.0 Nm. Fig. 3 and 4 show results.

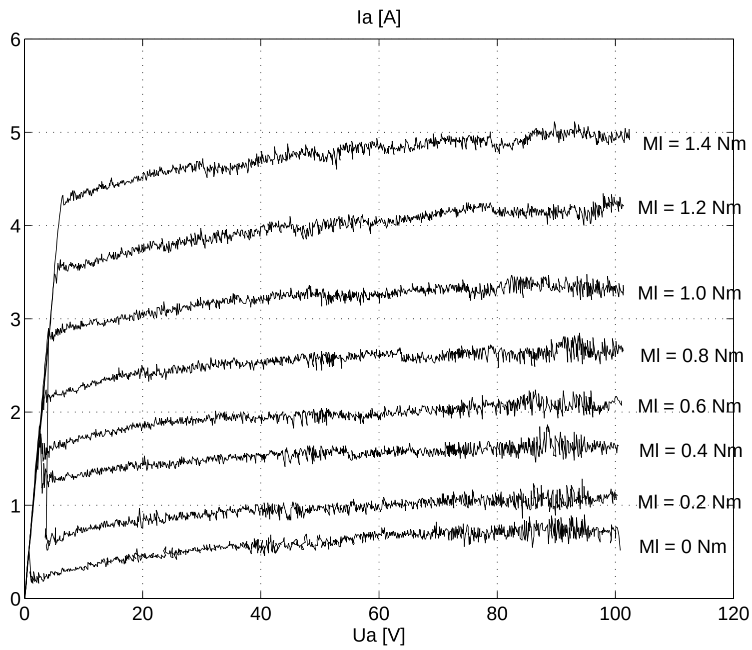

Fig. 1. Static characteristic of DC motor; $I_a = f(U_a, M_l)$

[1] Some corrections because of mechanical time constant were made when calculating M_{em} (see page 10).

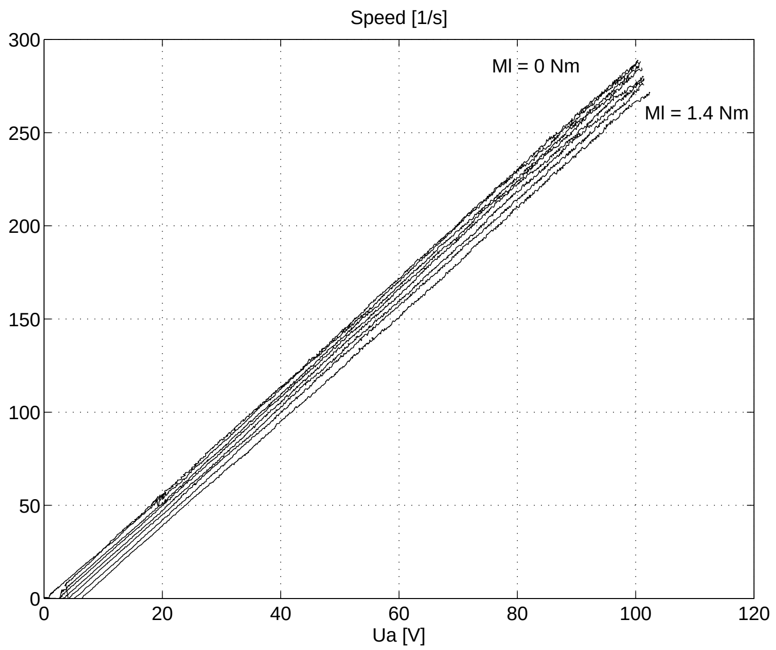

Fig. 2. Static characteristic of DC motor; $\omega = f(U_a, M_l)$

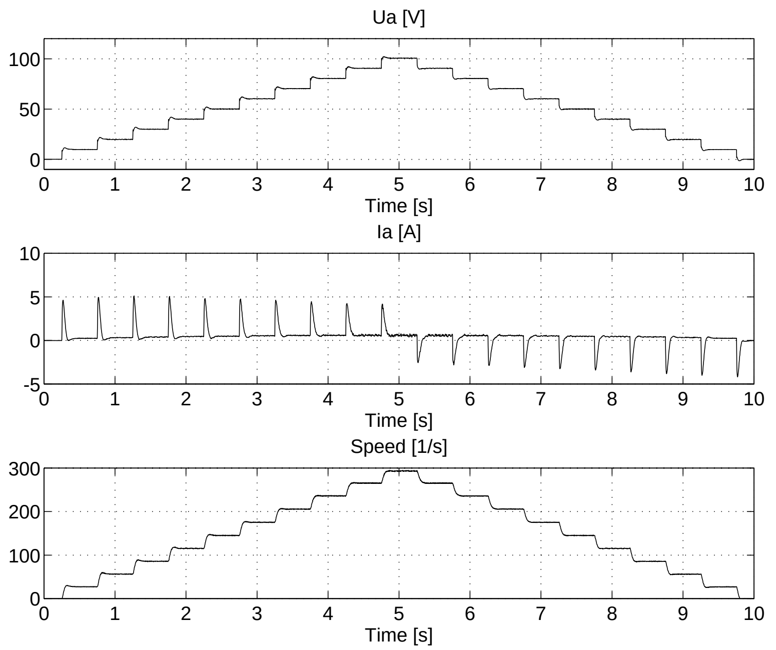

Fig. 3. Dynamic characteristic of DC motor @ $M_l = 0$ *Nm*

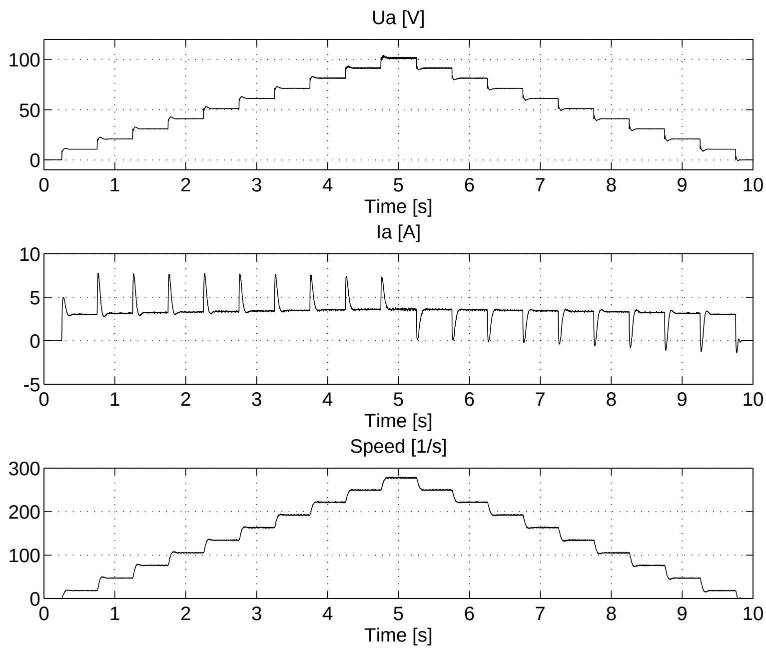

Fig. 4. Dynamic characteristic of DC motor @ M_l = 1.0 Nm

2.1 Static characteristics

From static measurements, the characteristics of $I_a = f(U_a, \omega)$, $M_f = f(\omega)$ and $M_{em} = f(I_a)$ have been calculated.

2.1.1 Characteristics of I_a

Several functions were used to describe armature current I_a:

a) $I_a' = k_1 U_a + k_2\omega + k_3 U_a \omega$

b) $I_a' = k_1 U_a + k_2\omega + k_3 U_a \omega + k_4 U_a{}^2 + k_5\omega^2$

c) $I_a' = \dfrac{U_a + k_1\omega}{k_2 + k_3\omega}$

d) $I_a' = \dfrac{U_a + k_1\omega + k_2\omega^2}{k_3 + k_4\omega}$

e) $I_a' = \dfrac{U_a + k_1\omega}{k_2 + k_3\omega + k_4\omega^2}$

To find appropriate constants $\{k_i\}$, the minimum of criterion function, written in table 1, was used. Table 1 shows models and results of criterion function. We can see, the model *b* have the smallest value of criterion function, but it is much more complex and without direct physical background. We chose model *c* which have still a small value of criterion function, but model is simple and has physical background similar to mathematical model written in eq. (1) (see [1]).

$I_a = f(U_a, \omega)$	$Crit.function = \frac{1}{T_{\max}} \int_0^{T_{\max}} (I_a' - I_a)^2 dt$					
Function	k1	k2	k3	k4	k5	crit. f.
$I_a' = k_1 U_a + k_2 \omega + k_3 U_a \omega$	0.6341	-0.2161	-1.938e-5			3.74e-2
$I_a' = k_1 U_a + k_2 \omega + k_3 U_a \omega + + k_4 U_a^2 + k_5 \omega^2$	0.6843	-0.2345	-1.371e-6	-5.710e-4	6.903e-5	2.93e-2
$I_a' = \frac{U_a + k_1 \omega}{k_2 + k_3 \omega}$	-0.3421	1.4723	9.602e-4			2.96e-2
$I_a' = \frac{U_a + k_1 \omega + k_2 \omega^2}{k_3 + k_4 \omega}$	-0.3426	2.57e-6	1.4609	1.059e-3		2.96e-2
$I_a' = \frac{U_a + k_1 \omega}{k_2 + k_3 \omega + k_4 \omega^2}$	-0.3421	1.467	1.06e-3	-2.483e-7		2.96e-2

Table 1: Mathematical models of armature current $I_a = f(U_a, \omega)$

Chosen mathematical model of I_a is therefore:

$$\boxed{I_a' = \frac{U_a - 0.3421\omega}{1.4723 + 9.602 \cdot 10^{-4} \omega}} \tag{3}$$

Optimisation procedure based on criterion function was written in program package MATLAB and is printed in appendix. The name of used MATLAB program is *DETIA1.M*.

Fig. 5 shows how model *c* fits the actual measurements of I_a.

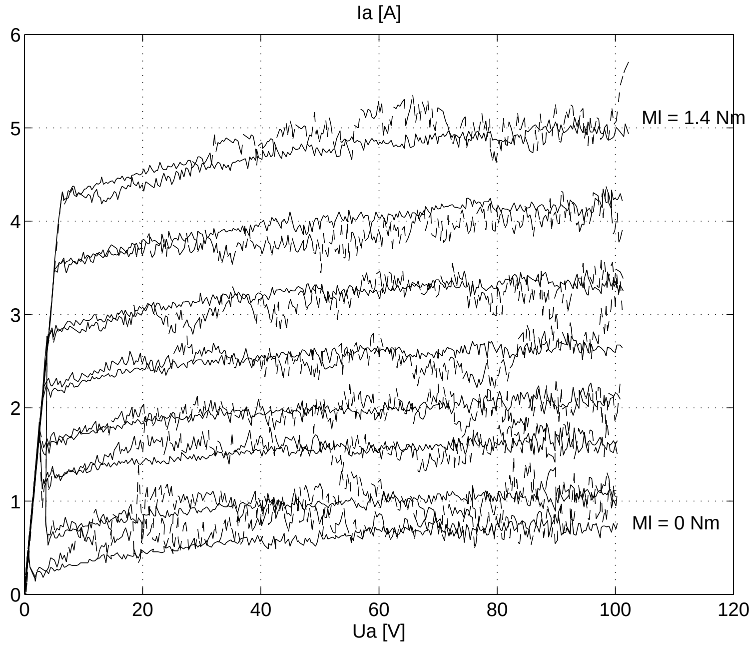

Fig. 5. Armature current (I_a); __ measurements, -- model

2.1.2 Characteristics of Mf

The second function which was determined was a moment of friction (M_f). To detect it, we added no load to the motor (M_l=0), connected a maximum voltage to the armature (U_a) and then disconnected it. That caused I_a=0 and M_{em}=0[2]. Eq. (2) changes:

$$\omega(t) = -\frac{1}{J}\int_0^t M_f(t)dt + \omega(0) \tag{4}$$

where $M_f = f(\omega)$. With a little modification of (4), we get:

$$J\big(\omega(t) - \omega(0)\big) = -\int_0^t M_f(t)dt \tag{5}$$

The measurement of speed (ω), when the armature current (I_a) was cut off is shown on Fig. 6.

[2] If there is no armature current, there should be no electromagnetic moment

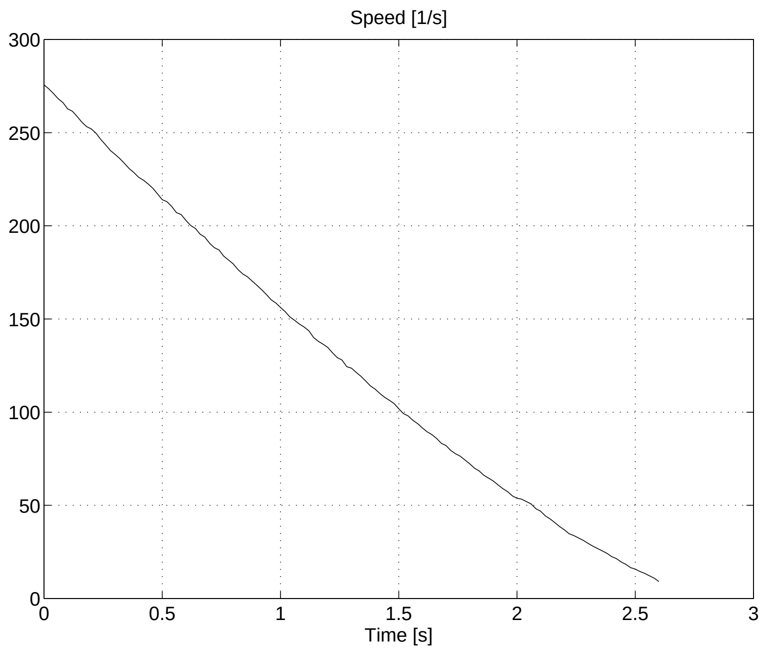

Fig. 6. Speed of the motor vs. time @ I_a=0

Three models were used to find friction moment M_f as a function of ω:

a) $M_f = k_1 + k_2\omega$

b) $M_f = k_1 + k_2\sqrt{\omega}$

c) $M_f = k_1 + k_2\omega + k_3\sqrt{\omega}$

$M_f = f(\omega)$	$Crit.function = \frac{1}{T_{\max}}\int_0^{T_{\max}}\left(J(\omega(t)-\omega(0)) + T\int_0^t M_f(t)\right)^2 dt$			
Function	k1	k2	k3	Crit. f.
$M_f = k_1 + k_2\omega$	0.1547	3.629e-4		1.16e-5
$M_f = k_1 + k_2\sqrt{\omega}$	0.1136	8.262e-3		4.58e-6
$M_f = k_1 + k_2\omega + k_3\sqrt{\omega}$	0.05304	-5.116e-4	0.01968	9.48e-7

Table 2: Mathematical models of the friction moment M_f=f(ω)

We can see, third model gives the best result. Fig. 7 to 9 show us measured speed of the DC motor and mathematical result when using all three models. Fig. 9 represents the result of the chosen model which obviously gives superior result.

The chosen model (*c*) of M_f is:

$$\boxed{M_f = 0.05304 - 5.116 \cdot 10^{-4}\,\omega + 0.01968\sqrt{\omega}} \qquad (6)$$

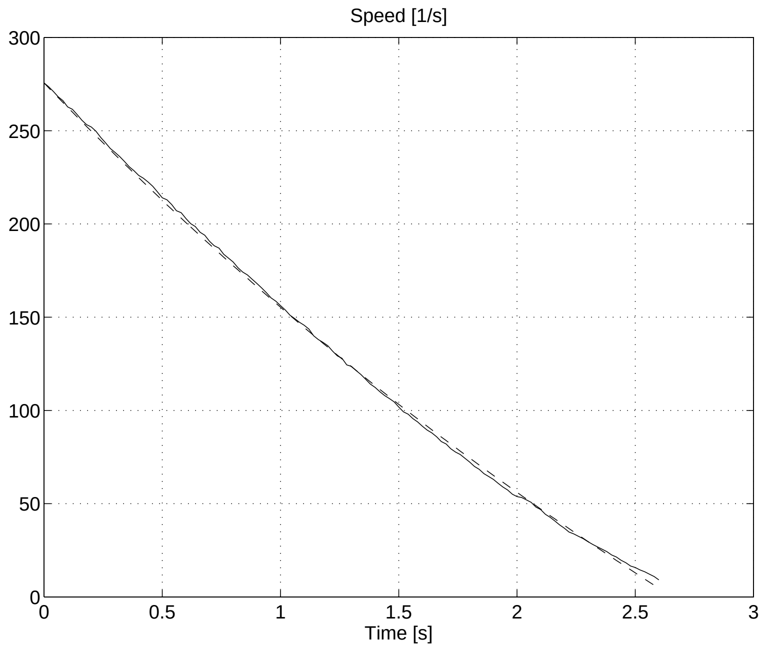

Fig. 7. Detection of M_f - model a; __ measurement of the speed, -- model

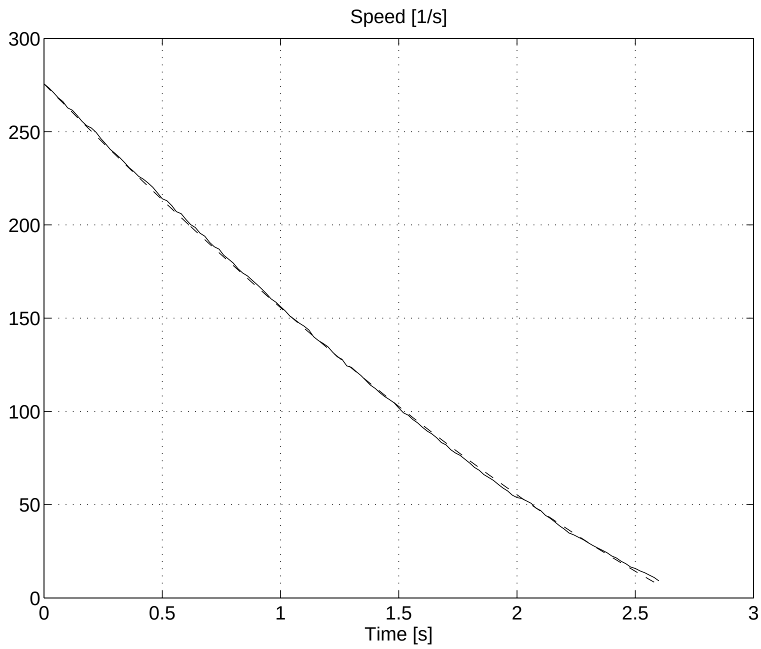

Fig. 8. Detection of M_f - model b; __ measurement of the speed, -- model

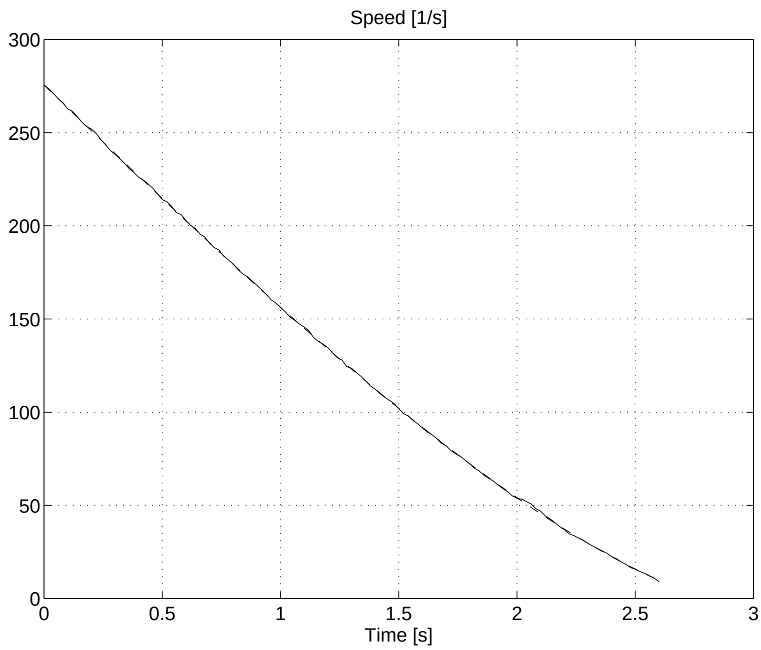

Fig. 9. Detection of M_f - model c; __ measurement of the speed, -- model

Optimisation procedure based on criterion function is given in appendix. The name of MATLAB program is *DETMF1.M*.

2.1.3 Characteristics of Mem

To made entire static model of the DC motor, we had to find an electromagnetic moment (M_{em}). It is caused by the current flowing through the coil of the rotor. We tried the following models:

a) $M_{em} = k_1 I_a$

b) $M_{em} = k_1 I_a + k_2 I_a^{\ 2}$

c) $M_{em} = k_1 I_a + k_2 I_a^{\ 2} + k_3 I_a^{\ 3}$

$M_{em} = f(I_a)$	$Crit.function = \frac{1}{T_{max}} \int_0^{T_{max}} \left(M_{em} - J\frac{d\omega}{dt} + M_l + M_f \right)^2 dt$			
Function	k1	k2	k3	Crit. f.
$M_{em} = k_1 I_a$	0.36132			5.68e-3
$M_{em} = k_1 I_a + k_2 I_a^{\ 2}$	0.44737	-0.02314		8.35e-4
$M_{em} = k_1 I_a + k_2 I_a^{\ 2} + k_3 I_a^{\ 3}$	0.41956	-4.1251e-3	-2.8642e-3	7.40e-4

Table 3: Mathematical models of electromagnetic moment $M_{em}=f(I_a)$

If eq. (2) is derived, we get:

$$J\frac{d\omega}{dt} = M_{em} - M_l - M_f \tag{7}$$

In a strictly static model, the left side of (7) is equal to 0. Because this is not quite true (see page 2), we add it to the criterion function in table 3 though its presence does not change the result. Because the value is so small, we used pre-defined value of J taken from Höfling [1]. We did not do it when calculating I_a (see chapter 2.1.1), because electrical time constant is much smaller compared with mechanical one.

Results obtained by three models can be seen on Fig. 10 to 12. The second model (*b*) gives the most satisfactory result. The mathematical model is therefore:

$$\boxed{M_{em} = 0.44737 I_a - 0.02314 I_a^{\,2}} \tag{8}$$

Optimisation procedure based on criterion function is written in appendix. The name of MATLAB program is *DETMEM1.M*.

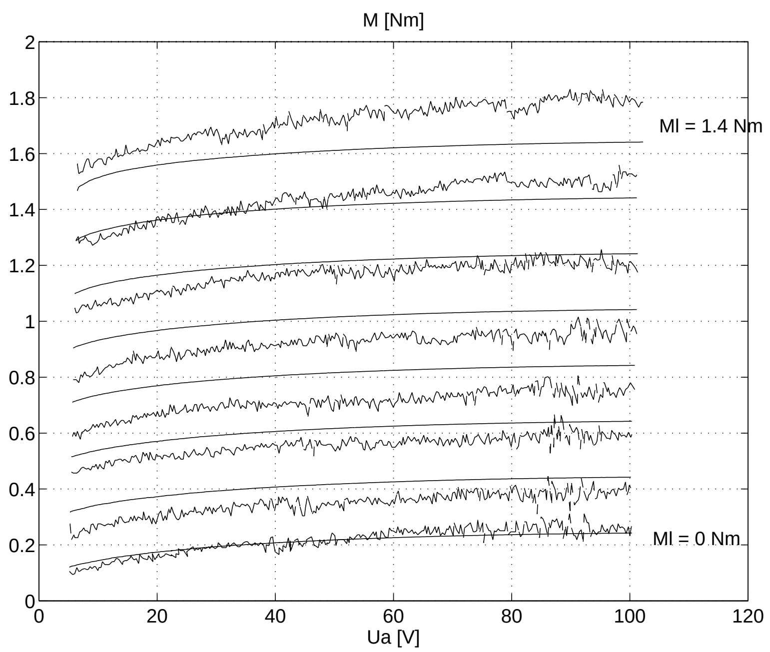

Fig. 10. Detection of M_{em} - model a; __ measurements -- model

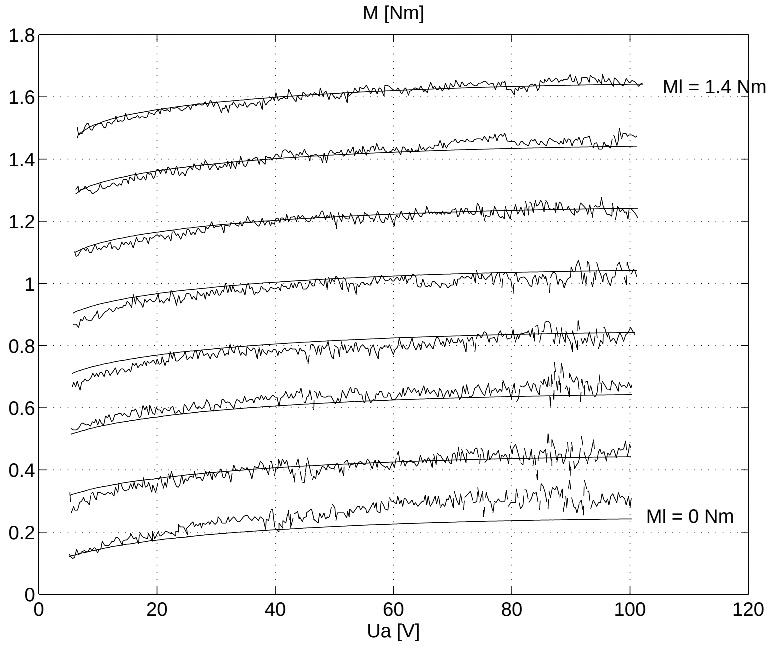

Fig. 11. Detection of M_{em} - model b; __ measurements, -- model

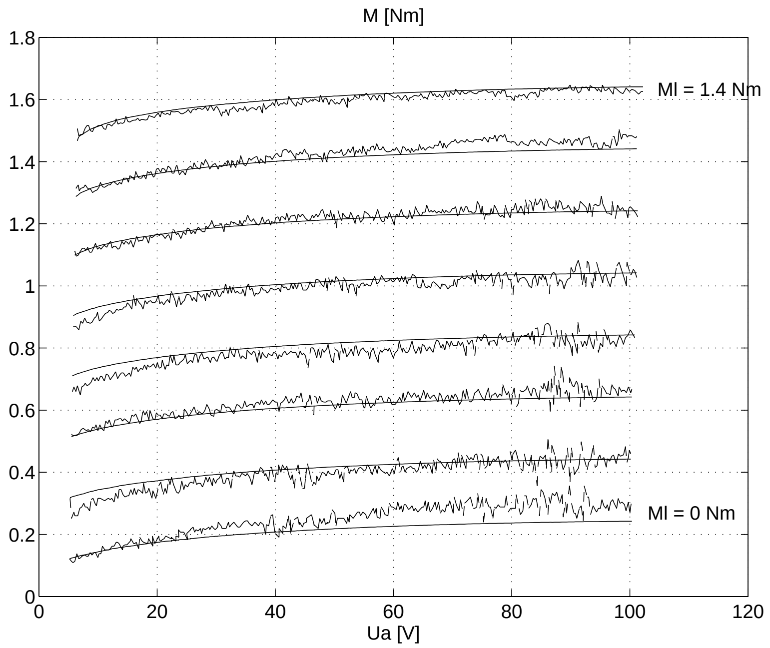

Fig. 12. Detection of M_{em} - model c; __ measurements, -- model

2.2 *Dynamic characteristics*

To detect dynamic constants we used dynamic measurements (see Fig. 3 and 4). There are two dynamic constants to be detected: armature inductance (L_a) and inertia (J). Eq. (1) and (2) describe system dynamics.

2.2.1 Estimation of La

Eq. (1) and (3) lead us to the next expression:

$$\begin{aligned} I_a'(t) = {} & \frac{1}{L_a}\int\left(U_a(t) - 0.34212\omega(t) - 1.4723 I_a'(t) - 9.6016\cdot 10^{-4} I_a'(t)\omega(t)\right)dt + \\ & + I_a'(0) \end{aligned} \tag{9}$$

Realised discrete algorithm to calculate I_a' can be as follows:

$$\begin{aligned} I_a'(i) = {} & I_a'(i-1) + \\ & + \frac{T_s}{L_a}\left(U_a(i) - 0.34212\omega(i) - 1.4723 I_a'(i-1) - 9.6016\cdot 10^{-4} I_a'(i-1)\omega(i)\right), \end{aligned} \tag{10}$$

where T_s is a sampling time. To find appropriate L_a, we were searching for such I_a' which would fit I_a. To diminish possible static errors, we used only dynamic measurements of the first current peak (see Fig. 3 and 13) as a used data.

Criterion function to find L_a was:

$$Crit.f. = \frac{1}{T_{\max}} \int_0^{T_{\max}} \left(I_a' - I_a\right)^2 dt \tag{11}$$

where we were searching for the minimum.

The results were L_a=11.71 mH for dynamic measurement at M_l=0Nm (see Fig. 3) and L_a=11.63mH when fitted measurements were those at M_l=1Nm (Fig. 4). How the model fits real measurement is shown on Fig. 13 and 14. As a resulting L_a, we took the mean value of the upper results and have

$$\boxed{L_a = 11.67mH} \tag{12}$$

Optimisation procedure based on criterion function is written in appendix. The name of MATLAB program is *DETLA1.M*.

Fig. 13 and 14 show us the difference between measurements and a model at M_l=0Nm and M_l=1.0Nm.

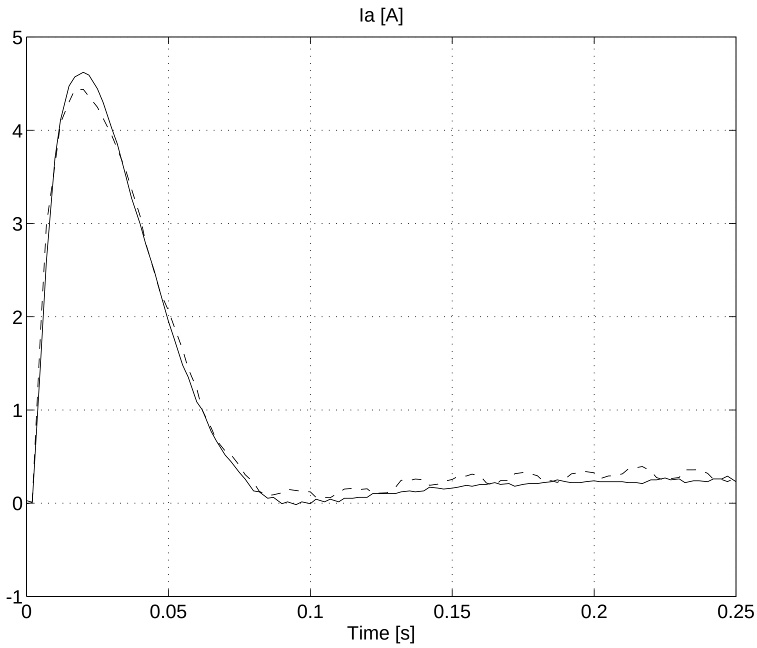

Fig. 13. Detection of L_a - L_a=11.71 mH @ M_l = 0 Nm; __ measurements, -- model

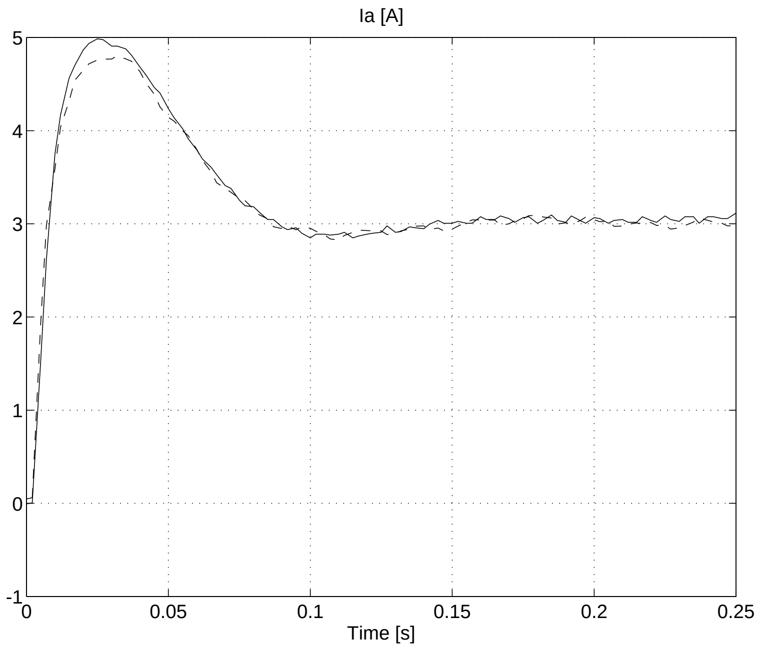

Fig. 14. Detection of L_a - L_a=11.63 mH @ M_l = 0 Nm; __ measurements, -- model

2.2.2 Detection of J

The same procedure was used to detect inertia moment (*J*). Instead of using eq. (1) and (3), we used (2), (6) and (8). We used only dynamic measurements at M_l=0Nm and we got:

$$\boxed{J = 1.889 \cdot 10^{-3}\,kgm^2} \tag{13}$$

Optimisation procedure based on criterion function is written in appendix. The name of MATLAB program is *DETJ1.M*.

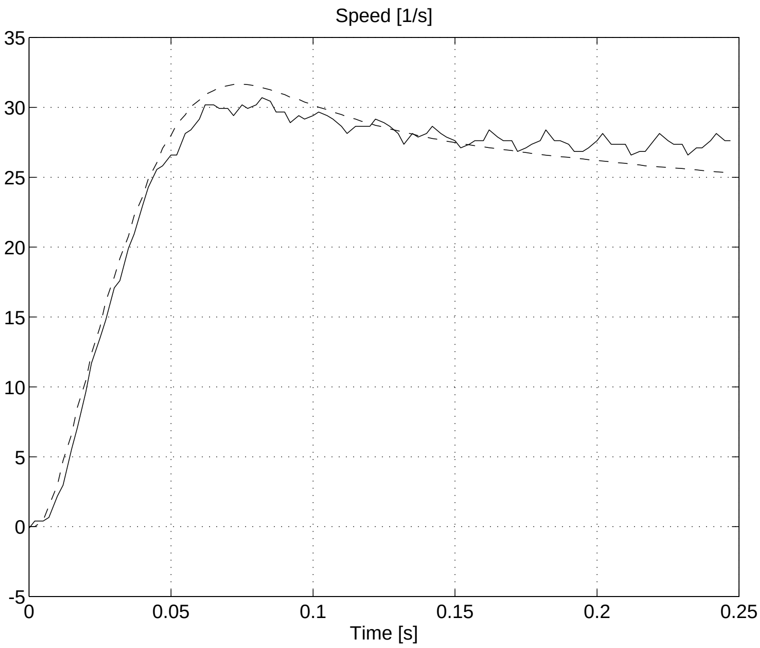

Fig. 15. Dynamic response; J=1.889 mH, M_l=0 Nm; __ measurements, -- model

2.3 Model of DC motor

The final model of the DC motor is:

$$\boxed{\begin{aligned}1.889e-3*\frac{d\omega}{dt} &= 0.44737I_a - 0.023142I_a^{\ 2} - \\ &\quad -0.05304 + 5.116e-4\omega - 1.968e-2\sqrt{\omega} - M_l\end{aligned}} \tag{14}$$

$$\boxed{1.167e-2\frac{dI_a}{dt} = U_a - 0.34212\omega - 1.4723I_a - 9.6016e-4I_a\omega} \tag{15}$$

The model was built and simulated in program package SIMULINK. The results, how model fits static and dynamic measurements, are shown on Fig. 15 to 27.

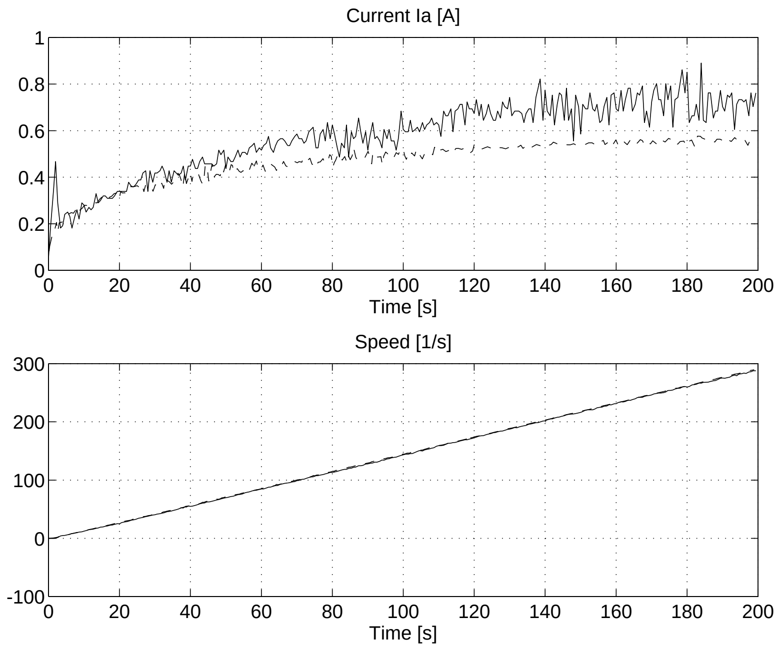

Fig. 16. Static data @ M_l = 0 Nm; __ measurements, -- model

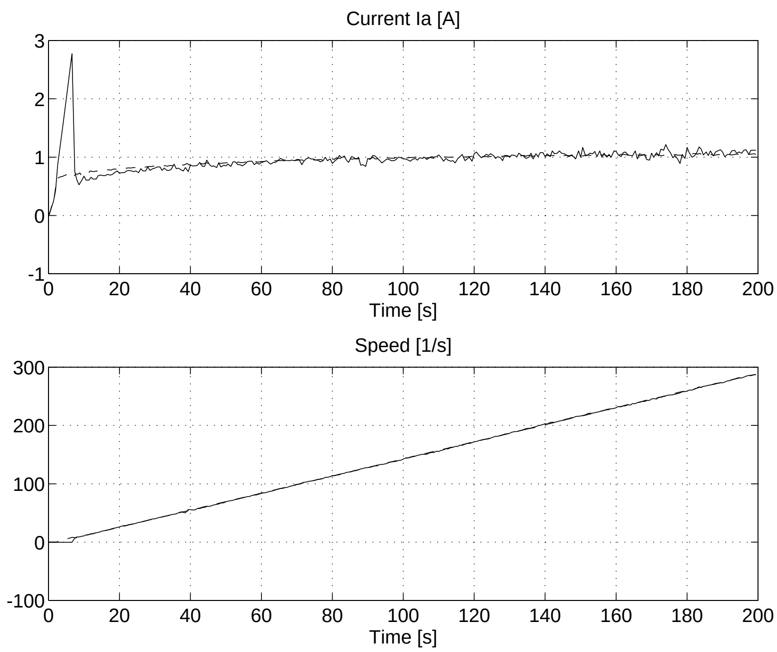

Fig. 17. Static data @ M_l = 0.2 Nm; __ measurements, -- model

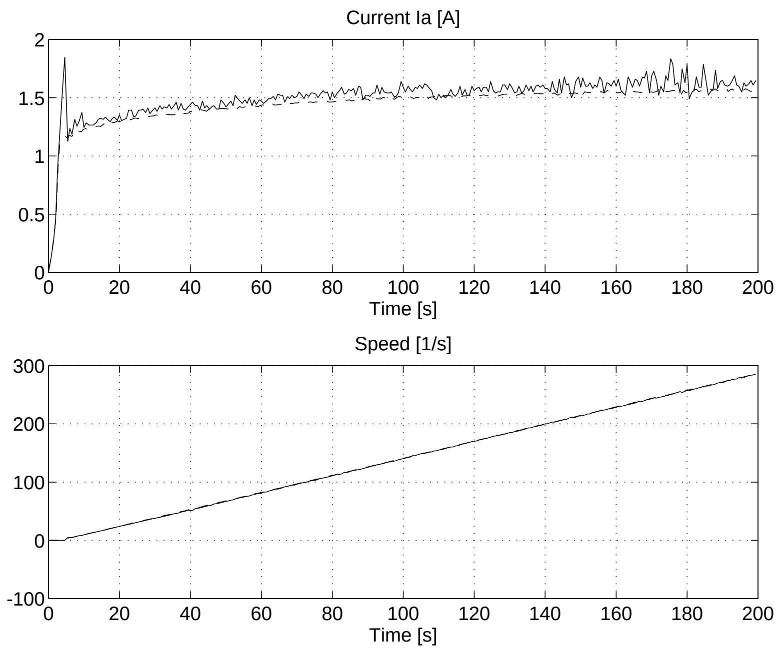

Fig. 18. Static data @ M_l = 0.4 Nm; __ measurements, -- model

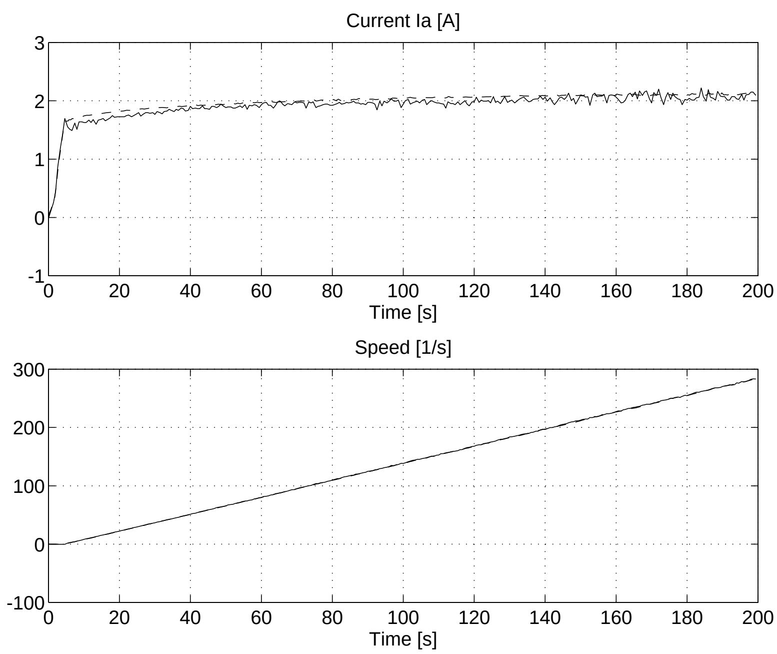

Fig. 19. Static data @ M_l = 0.6 Nm; __ measurements, -- model

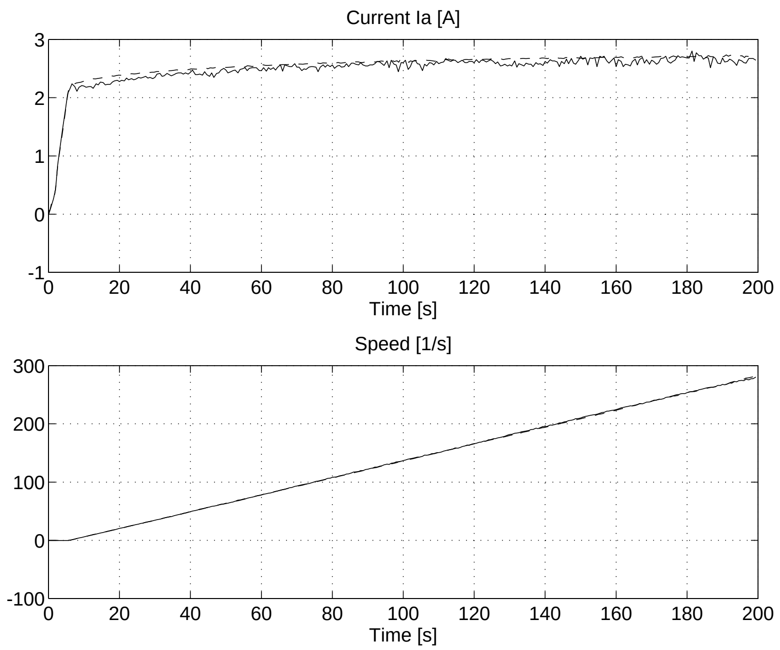

Fig. 20. Static data @ M_l = 0.8 Nm; __ measurements, -- model

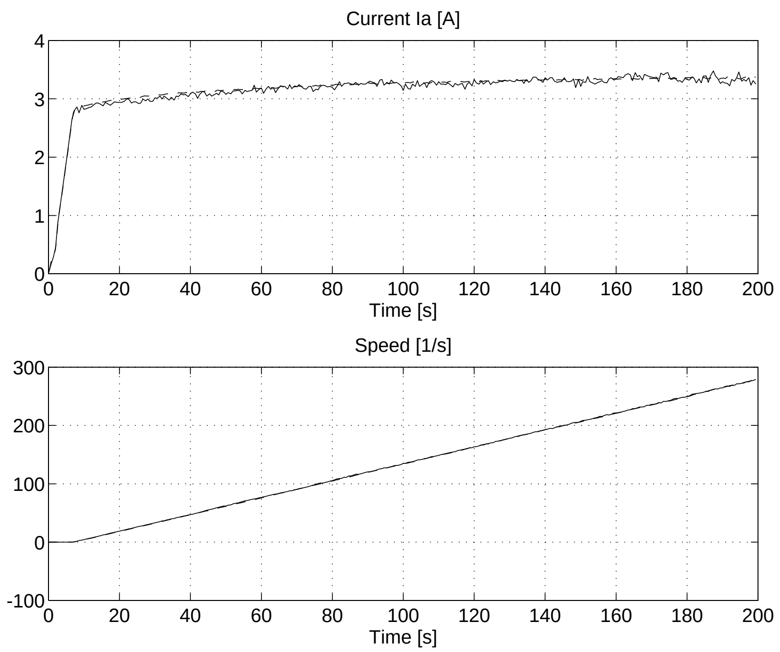

Fig. 21. Static data @ M_l = 1.0 Nm; __ measurements, -- model

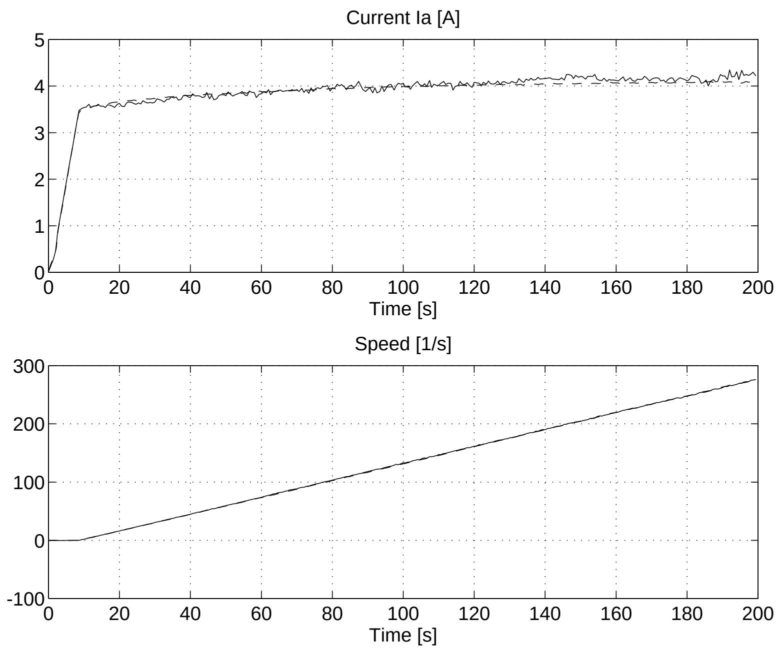

Fig. 22. Static data @ M_l = 1.2 Nm; __ measurements, -- model

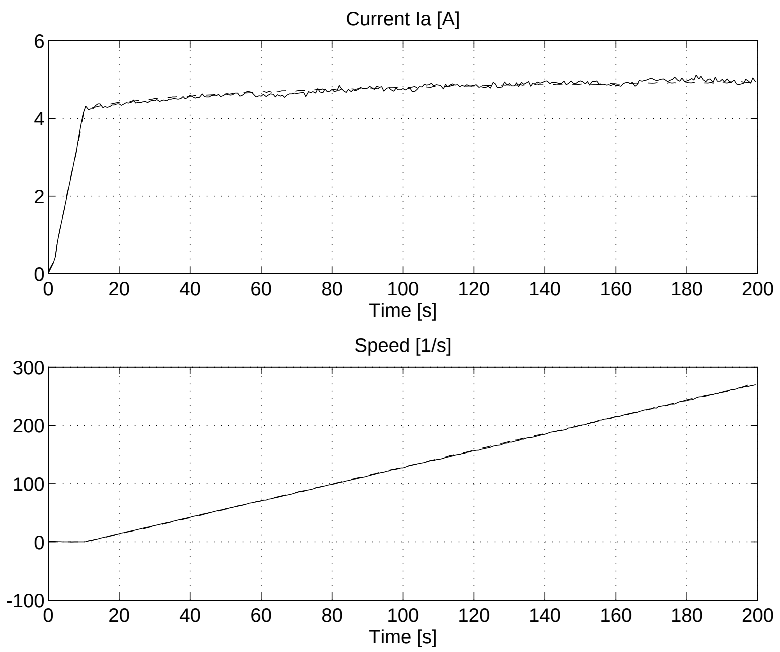

Fig. 23. Static data @ M_l = 1.4 Nm; __ measurements, -- model

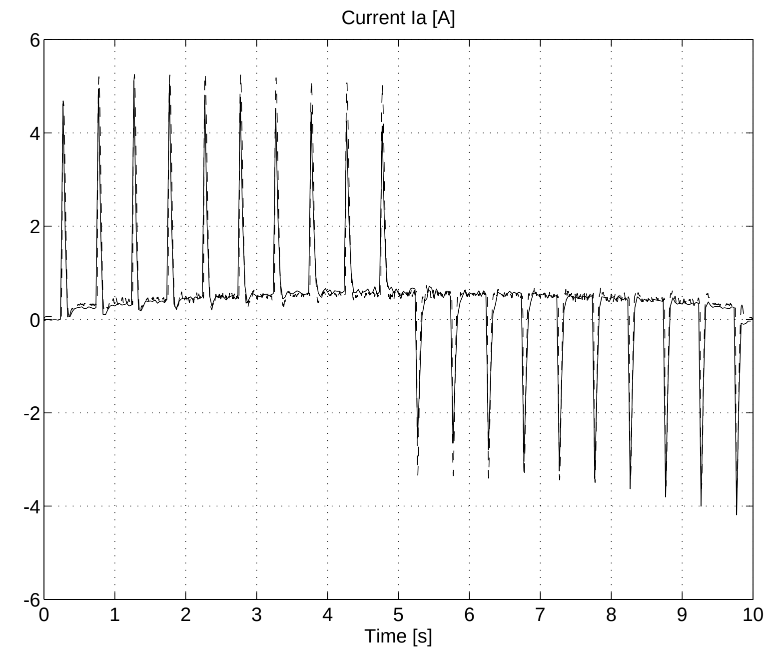

Fig. 24. Dynamic data - I_a @ M_l = 0 Nm; __ measurements, -- model

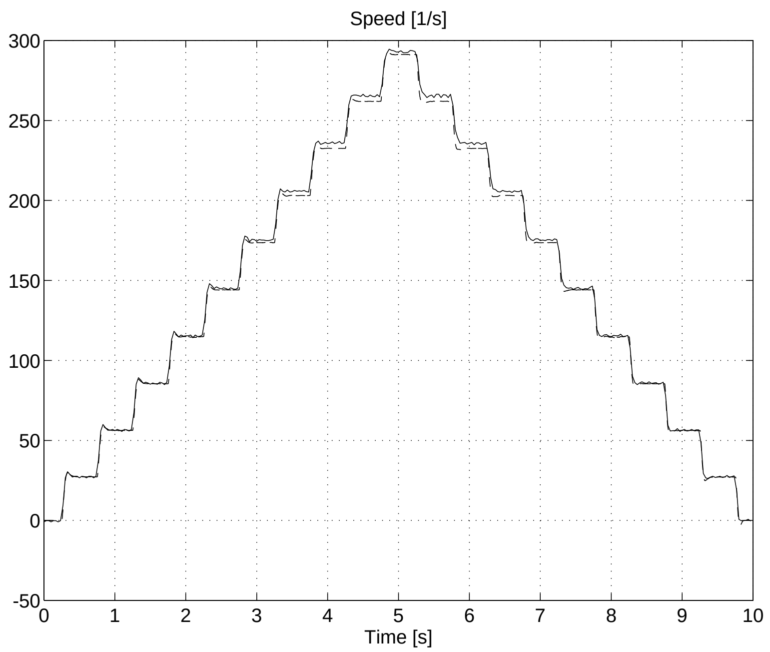

Fig. 25. Dynamic data - ω@ M_l = 0 Nm; __ measurements, -- model

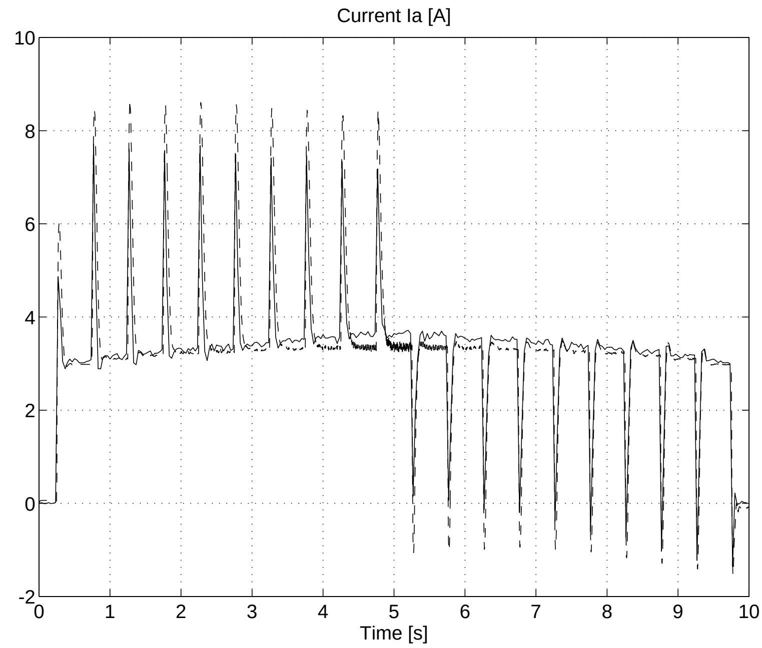

Fig. 26. Dynamic data - I_a @ M_l = 1.0 Nm; __ measurements, -- model

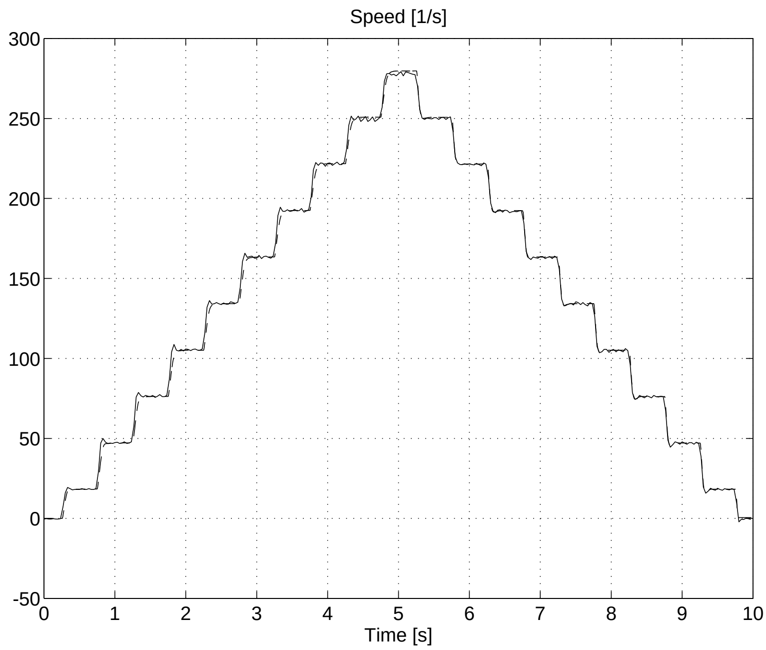

Fig. 27. Dynamic data - ω @ M_l = 1.0 Nm; __ measurements, -- model

3. Conclusions

Simulation results show that the mathematical model fits a real DC motor quite well. Results are very good when simulating static behaviour. Dynamic results are not as good as static, so more attention should be paid to build-up better dynamic model[3].

[3] The problem is in fact that the static Ia-Ua function was modeled with the quadratic function instead of the square/root function, which gives *almost perfect static and dynamic matching of the model and the measurements*. The reason is that during transition of speed, we are moving quite beyond the static identified region, where quadratic characteristic does not describe well the real situation. [This remark was written after the report was already finished]

4. Literature

[1] Höfling, "*Fault detection of a D.C. motor using grey box modelling and continuous-time parity equations*", Electrotechnical and Computer Science Conference ERK'94, Portorož, Slovenia, Vol. A, pp. 175-178, 1994.

[2] Juričić, *"Grey box identification for diagnostic purposes: An example of a DC motor*", Internal report, J. Stefan Institute, Ljubljana, Slovenia, 1994.

5. Appendix

Program REDUC.M:

```
% res = reduc(input,number);
% function reduc reduces a number of points in vector (matrix)
%
% input - input vector or matrix
% number - number of desired legth of vector (matrix)

function res = reduc(input,number);

a = size(input);
ratio = a(1)/number;
vect = [1:ratio:a(1)];
res = input(vect,:);
```

Program DETIA1.M:

```
ia = [motpc00(:,4) motpc02(:,4) motpc04(:,4) motpc06(:,4) motpc08(:,4) motpc10(:,4) motpc12(:,4) motpc14(:,4)];
ua = [motpc00(:,5) motpc02(:,5) motpc04(:,5) motpc06(:,5) motpc08(:,5) motpc10(:,5) motpc12(:,5) motpc14(:,5)];
w = [motpc00(:,6) motpc02(:,6) motpc04(:,6) motpc06(:,6) motpc08(:,6) motpc10(:,6) motpc12(:,6) motpc14(:,6)];

k = fmins('detia',[-0.34212 1.4723 9.6016e-4],[1 1e-6 1e-10 0 0 0 0 0 0 0 0 0 0 800],[],t,ia,ua,w);

ian=reduc(ia,300);
uan=reduc(ua,300);
wn=reduc(w,300);

plot(uan,ian,uan,(uan+k(1)*wn)./(k(2)+k(3)*wn),'--');
title('Ia [A]')
grid
xlabel('Ua [V]')
gtext('Ml = 0 Nm')
gtext('Ml = 1.4 Nm')
axis([0 120 0 6])
```

Program DETIA.M:

```
% function a=detia(x,t,ia,ua,w);
%
% function detia is a criterion function to determine armature current
% of DC motor.

function a=detia(x,t,ia,ua,w);

b = (ua+x(1)*w)./(x(2)+x(3)*w);

c = (b-ia).*(b-ia);
a = mean(mean(c));
```

Program DETMF1.M:

```
t=iaoff(40:170,1)-iaoff(40,1);
w=iaoff(40:170,6);
dw=w-w(1);
wint=0.02*cumsum(w);
wint=wint-wint(1);
```

```
sqw=sqrt(w);
wsqint=0.02*cumsum(sqw);
wsqint=wsqint-wsqint(1);
w2=w.*w;
w2int=0.02*cumsum(w2);
w2int=w2int-w2int(1);

k = fmins('detmf',[0.15 0 0],[1 1e-6 1e-10 0 0 0 0 0 0 0 0 0 0 500],[],dw,t,wint,w2int,wsqint);

plot(t,w,t,(-k(1)*t-k(2)*wint-k(3)*wsqint)/1.932e-3+w(1),'--');
title('Speed [1/s]')
grid
xlabel('Time [s]')
```

Program DETMF.M:

```
% function a=detmf(x,ynorm,t,wint,w2int,wsqint);
%
% function detmf is criterion function to find friction Mf of DC motor

function a=detmf(x,ynorm,t,wint,w2int,wsqint);

b = 1.932e-3*ynorm+x(1)*t+x(2)*wint+x(3)*wsqint
b = b.*b;
a = mean(b);
```

Program DETMEM1.M:

```
t = motpc00(55:1000,1);
w = [motpc00(55:1000,6) motpc02(55:1000,6) motpc04(55:1000,6) motpc06(55:1000,6) motpc08(55:1000,6)
motpc10(55:1000,6) motpc12(55:1000,6) motpc14(55:1000,6)];
ia = [motpc00(55:1000,4) motpc02(55:1000,4) motpc04(55:1000,4) motpc06(55:1000,4) motpc08(55:1000,4)
motpc10(55:1000,4) motpc12(55:1000,4) motpc14(55:1000,4)];
ua = [motpc00(55:1000,5) motpc02(55:1000,5) motpc04(55:1000,5) motpc06(55:1000,5) motpc08(55:1000,5)
motpc10(55:1000,5) motpc12(55:1000,5) motpc14(55:1000,5)];
ml = 1.0*ones(size(w));
ml = [0.0*ml(:,1) 0.2*ml(:,1) 0.4*ml(:,1) 0.6*ml(:,1) 0.8*ml(:,1) 1.0*ml(:,1) 1.2*ml(:,1) 1.4*ml(:,1)];

k = fmins('detmem',[0.36 0],[1 1e-6 1e-10 0 0 0 0 0 0 0 0 0 0 300],[],w,ia,ml,ua);

y1 = 1.932e-3*288/200+5.3041e-2-5.116e-4*w+1.968e-2*sign(w).*sqrt(abs(w))+ml;
y2 = k(1)*ia+k(2)*ia.*ia;

y1 = reduc(y1,300);
y2 = reduc(y2,300);

plot(reduc(ua,300),y1,reduc(ua,300),y2,'--')

title('M [Nm]')
grid
xlabel('Ua [V]')
gtext('Ml = 0 Nm')
gtext('Ml = 1.4 Nm')
```

Program DETMEM.M:

```
% function a=detmem(x,w,ia,ml,ua);
%
% function detmem is a criterion function to find a value of electomagnetic
% moment Mem of a DC motor
```

```
function a=detmem(x,w,ia,ml,ua);

b = 1.932e-3*288/200+5.3041e-2-5.116e-4*w+1.968e-2*sign(w).*sqrt(abs(w))+ml-x(1)*ia-x(2)*ia.*ia;
b = b.*b;
a = mean(mean(b));
```

Program DETLA1.M:

```
t = motvar00(100:200,1)-motvar00(100,1);
ia = motvar00(100:200,4);
ua = motvar00(100:200,5);
w = motvar00(100:200,6);

global ias;

k = fmins('detla',[6.963e-3],[1 1e-6 1e-10 0 0 0 0 0 0 0 0 0 0 300],[],ia,ua,w);

plot(t,ia,t,ias,'--')
```

Program DETLA.M:

```
% function a = detla(x,ia,ua,w);
%
% function detla is criterion function to find armature inductance
% of DC motor

function a = detla(x,ia,ua,w);

global ias;

ias=[];
d = size(ia);

for i = 1:d(1)
  if (i == 1)
    di = 2.5e-3*(ua(i)-0.34212*w(i));
    ias(i) = di/x(1);
  else
    di = 2.5e-3*(ua(i)-0.34212*w(i)-1.4723*ias(i-1)-9.6016e-4*w(i).*ias(i-1));
    ias(i) = ias(i-1)+di/x(1);
  end
end

b = ia-ias';
c = b.^2;

a = mean(mean(c));
```

Program DETJ1.M:

```
t = motvar00(100:200,1)-motvar00(100,1);
ia = motvar00(100:200,4);
ua = motvar00(100:200,5);
w = motvar00(100:200,6);
ml = 0.0*ones(size(t));

t = reduc(t,500);
ia = reduc(ia,500);
w = reduc(w,500);
ml = reduc(ml,500);
```

```
global ws;

k = fmins('detj',[J],[1 1e-6 1e-10 0 0 0 0 0 0 0 0 0 0 300],[],ia,ua,w,ml);

plot(t,w,t,ws,'--')
```

Program DETJ.M:

```
% function a = detj(x,ia,ua,w,ml);
%
% function detj is a criterion function what serves to find inertia
% constant of DC motor

function a = detj(x,ia,ua,w,ml);

global ws;

ws=[];
d = size(ia);

for i = 1:d(1)
  if (i == 1)
    ws(i) = 0;
  else
    mem = 0.44737*ia(i)-0.023142*ia(i)*ia(i);
    mminus = ml(i)+0.053011-5.116e-4*w(i-1)+0.01968*sign(w(i-1))*sqrt(abs(w(i-1)));
    if ((abs(ws) < 0.1) & (abs(mem) < abs(mminus)))
      dw = 0;
    else
      dw = 0.2*2.5e-3*(mem-mminus);
    end
    ws(i) = ws(i-1)+dw/x(1);
  end
end

b = w-ws';
c = b.^2;

a = mean(mean(c));
```