Tuning of PI regulators in distributed control system for an electric vehicle

Abstract. In this paper a drive system for an urban electric vehicle is presented. The electric propulsion system consists of two in-wheel outer-rotor permanent magnet synchronous motors (PMSMs) fed by two three-level inverters. A proposed control system has been designed as a distributed one. The outer speed loop invokes a master digital signal controller (DSC) whereas the inner current loops are closed using two slave microcontrollers. An effective analytical tuning of such a control system requires accurate identification of delays presented in all the paths, including the controller area network (CAN) bus used here to intercommunicate all the DSCs, the delays inherently related to a digital (sampled) nature of the system and the delay caused by a pulse width modulator. A step-by-step description of the analytical tuning of the controllers is provided. The tuning procedure respects all the identified delays and is validated experimentally in a non-mobile laboratory setup.


Keywords: electric vehicle, Naslin polynomial method, optimal modulus criterion, system delays

Słowa kluczowe: pojazd elektryczny, metoda wielomianu Naslina, kryterium modułowego optimum, opóźnienia systemowe

Introduction

In the last years, the electric vehicles (EVs) have been investigated as an alternative form of transportation. They are seen as a way to reduce pollution emissions and noise from automobiles in the centre of the cities. Moreover, they have several advantages over internal combustion engine vehicles. The most important one is the possibility of applying a full torque at low speed, which gives opportunity to eliminate the clutch and gearbox. Another feature is easy feasibility of the independent drive of each wheel.

Figure 1 shows topology of the powertrain of the designed electric vehicle (ECO-Car). The vehicle drivetrain system consists of two in-wheel outer-rotor permanent magnet synchronous motors (PMSMs) mounted in the rear wheels [1, 2]. The PMSMs are fed by two three-level DC/AC inverters. The topology of inverters has been chosen to improve an efficiency of energy conversion and a quality of the output voltage waveform. The vehicle has been equipped with a hybrid energy storage system, containing LiFePO₄ cells and ultracapacitor stack. The ultracapacitor stack support electrochemical battery during acceleration and regenerative braking. The electrochemical battery and ultracapacitor stack are connected to the drive inverters through interleaved DC/DC converters [3].

In the ECO-Car, due to using the direct-drive a control system has been designed as a distributed one. The control system includes three digital signal controllers (DSCs). One DSC operates as a master controller, the other DSCs operate as slaves. The current controllers of each motor have been implemented on the slave DSCs. The algorithm of the speed control is executed by master controller. The controller area network (CAN) bus is used here to intercommunicate all the DSCs [4-9]. It is the standard protocol in automotive applications related to on-board communication systems [10, 11]. The master and slave controllers exchange information about actual speed of each motor and reference currents. A schematic representation of the distributed control system is shown in Fig. 2.

This paper deals with the design of speed and current controllers for PMSM drive. The PI current controller has been tuned based on optimal modulus criterion [12]. For comparative purposes, two methods the Naslin polynomial [13] and the Matlab’s systune function [14] have been used to select the PI speed controller parameters. The system delays are taken into account in the calculation of controller gains. The tuning process of PI controllers has been verified on a test rig.

Model of the drive system

The dynamic model of the PMSM in an orthogonal dq coordinate system can be described as:

\[ u_d = R_d i_d + L_d \frac{di_d}{dt} - p\omega_m L_i q, \]

\[ u_q = R_q i_q + L_q \frac{di_q}{dt} + p\omega_m L_i d + p\omega_m \varphi_f, \]

where: \( u_d, u_q \) – stator voltages in d and q axes, \( i_d, i_q \) – stator currents in d and q axes, \( R_d, R_q \) – stator resistance, \( L_d, L_q \) – stator inductances in d and q axes, \( \varphi_f \) – permanent magnetic flux linkage, \( p \) – the number of pole pairs, \( \omega_m \) – rotor angular speed.

The electromagnetic torque is given by:

\[ T_e = -\frac{3}{2} p\varphi_f (L_d - L_q)i_d i_q. \]
For a surface mounted PMSM, we have \( L_d = L_q \), thus torque equation is
\[
T_e = \frac{3}{2} p \phi f I_q.
\]
In order to eliminate the coupled term in (1) and (2), new variables
\[
u_{dc} = -p m L_q I_q,
\]
\[
u_{qc} = p m L_q I_q + p m \phi f
\]
are subtracted from those equations. Considering that, the symmetric PMSM has been used for the drive system, the transfer functions for the two currents in \( d \) and \( q \) axes are the same
\[
G_{RL} = \frac{R_s}{1 + s T_{RL}} = K_{RL}.
\]
The mechanical equation describing behaviour of PMSM motor is
\[
d\omega_m = \frac{1}{J} (T_e - r F_R),
\]
where: \( r \) – wheel radius, \( F_R \) – resistance force. The resultant moment of inertia \( J \) is a sum of the moments of inertia from motor \( J_m \), wheel \( J_w \) and the one associated with the vehicle \( J_v \).
\[
J = J_m + J_w + J_v.
\]
The motor is supplied from voltage source inverter (VSI) controlled by using PWM method. VSI is approximated by the first order transfer function
\[
G_{PWM} = \frac{K_{PWM}}{1 + s T_{PWM}},
\]
where: \( K_{PWM} \) – VSI gain, \( T_{PWM} \) – delay caused by a pulse width modulator. The reference signal of PWM is updated once per period. Therefore, it has been assumed that delay \( T_{PWM} \) is equal a half of the PWM period [15, 16].

The delays occurred in control system are caused by filtering of speed \( T_{max} \), CAN bus \( T_{CAN} \), as well as the calculating algorithms of the slave DSCs \( T_{di} \) and the master DSC \( T_{ds} \). Each of the delays have been approximated as first order transfer function [15, 16, 17].

A discrete PI controller with anti-windup shown in Figure 3 has been adopted in the simulation and the experimental studies.

\[
G_c = \frac{K_{PWM} K_{RL} K_{mc}}{(1 + s T_{di})(1 + s T_{PWM})(1 + s T_{RL})},
\]
where: \( K_{mc} \) – measurement gain of the current.

In the plant, the dominant time constant is given by
\[
T_c = T_{RL} = \frac{L_q}{R_s}
\]
and sum of small time constants is as follows
\[
T_{\Sigma} = T_{di} + T_{PWM}.
\]

To design PI current controller, the plant transfer function (11) can be approximated by
\[
G_{co} = \frac{K_c}{(1 + s T_{c})(1 + s T_{c})},
\]
where: \( K_c \) – gain of the plant.
The gain of the plant is equal to
\[
K_c = \frac{K_{PWM} K_{mc}}{R_s}.
\]
Parameters of PI controller are obtained by using the modulus optimum method for the transfer function (14). This method ensures that the magnitude of the closed loop transfer function is almost equal to 1 in the widest possible frequency range. The block diagram of the current control loop is shown in Figure 4.

The proportional $K_{pc}$ and integral $K_{ic}$ gain of the PI current controller is obtained from:

\[ K_{pc} = \frac{T_c}{2K_{iC}}, \]
\[ K_{ic} = \frac{1}{2K_{iC}}. \]

**Speed controller**

In order to determine gains of PI speed controller, the transfer function of closed loop current control should be calculated by substituting equations (12-17) and is equal to:

\[ G_{cc} = \frac{1}{2T_{mc}^2s^2 + 2T_{mc}s + 1}, \]

and assuming that $2T_{mc}^2 << 1$:

\[ G_{cc} \approx \frac{1}{1 + 2T_{mc}s}. \]

The transfer function for the plant shown in Figure 5 is given by:

\[ G_v = \frac{K_{m0}1.5p\phi_f}{s(1+sT_{d2})(1+s2T_{CAN})(1+s2T_{mc})(1+sT_{m0})}, \]

where $K_{m0}$ - measurement gain of the speed.

The sum of small time constants is expressed by

\[ T_{mc} = 2T_{CAN} + T_{d2} + T_{m0} + 2(T_{d1} + T_{PWM}) \]

The gain of the plant is equal to

\[ K_v = \frac{K_{m0}1.5p\phi_f}{K_{mc}J}. \]

The transfer function of plant (20) can be approximated with the transfer function

\[ G_{vo} = \frac{K_v}{s(1+sT_{mc})}. \]

The proportional gain and integral gain can be calculated by using the Naslin polynomial method:

\[ K_{pv} = \frac{1}{aK_vT_{mc}}. \]

where value of $a$ factor affects rise time and overshoot in the process control.

The speed controller designed by Naslin polynomial method is compared with the controller tuned by the Matlab's `systune` function. The `TuningGoal.StepRejection` class [18] is used to specify a desired response for a step disturbance. The pseudocode with the `systune` function is described in Appendix. The parameters of the drive with PMSM and values of delays are presented in Table 1.

### Table 1. The parameters of PMSM and the values of delays

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_s$</td>
<td>1.1 Ω</td>
</tr>
<tr>
<td>$L_{q}$, $L_d$</td>
<td>15.57 mH</td>
</tr>
<tr>
<td>$J$</td>
<td>0.0201 kgm²</td>
</tr>
<tr>
<td>$p$</td>
<td>4</td>
</tr>
<tr>
<td>$T_{PWM}$</td>
<td>50 µs</td>
</tr>
<tr>
<td>$T_{d1}$</td>
<td>100 µs</td>
</tr>
<tr>
<td>$T_{d2}$</td>
<td>100 µs</td>
</tr>
<tr>
<td>$T_{CAN}$</td>
<td>2000 µs</td>
</tr>
<tr>
<td>$T_{m0}$</td>
<td>2500 µs</td>
</tr>
</tbody>
</table>

Values of the speed and current controller parameters used in simulation and experiment are shown in Table 2.

### Table 2. The parameters of PI controllers

<table>
<thead>
<tr>
<th></th>
<th>PI current controller</th>
<th>PI speed controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_v$</td>
<td>4.1</td>
<td>$a = 2$</td>
</tr>
<tr>
<td>$K_p$</td>
<td>14.2</td>
<td>$a = 3$</td>
</tr>
<tr>
<td>$K_i$</td>
<td>9.4</td>
<td>$a = 4$</td>
</tr>
<tr>
<td>$K_{ic}$</td>
<td>7.1</td>
<td>systune</td>
</tr>
<tr>
<td>$K_{ic}$</td>
<td>63.9</td>
<td></td>
</tr>
<tr>
<td>$K_{ic}$</td>
<td>507.0</td>
<td></td>
</tr>
</tbody>
</table>

### Simulation and experimental results

The drive system with the PMSM, inverter and the designed controllers have been tested in Matlab/Plecs environment.

The experiments have been carried out by using a non-mobile laboratory setup (Fig. 6). The drive was equipped with the VSIs and the digital signal controllers TMS320F3335.
Figure 7 shows the $i_q$ current response for the step change in the reference current from 0 to 10 A. The tuned current controller provides the response without overshoot and the settling time of about 0.05 s. The relatively slow dynamics of the response is caused by the employed anti-windup scheme (Fig. 3). The experimental result shown in Figure 8 corresponds with the simulation result in Figure 7.

![Fig. 7. Simulation result for current response](image)

Fig. 7. Simulation result for current response

![Fig. 8. Experimental result for current response](image)

Fig. 8. Experimental result for current response

Figures 9 and 10 depict the speed response of the control system for the step change in the reference speed from 0 to 50 rad/s. The step of load from 0 to 100% occurs in 0.7 s and returns to 0 in 1.4 s. In the simulation and experimental studies speed controller was examined for various values of $\alpha$ factor.

Figures 11 and 12 show the speed response of the control system with speed controller tuned by `systune` function. The speed is with almost zero overshoot and satisfying control performance.

![Fig. 9. Simulation results for speed response](image)

Fig. 9. Simulation results for speed response

![Fig. 10. Experimental results for speed response](image)

Fig. 10. Experimental results for speed response

![Fig. 11. Simulation result for speed response with PI controller tuned by `systune` function](image)

Fig. 11. Simulation result for speed response with PI controller tuned by `systune` function

![Fig. 12. Experimental result for speed response with PI controller tuned by `systune` function](image)

Fig. 12. Experimental result for speed response with PI controller tuned by `systune` function

**Conclusions**

This paper presents design approach of the controllers for the distributed control system of EV. A mathematical model of drive system, including all delays occurring in the control loops, has been developed. The model has been used to synthesize the speed and current control loops of the drive. The method of modulus optimum was used to derive expressions for the parameters of PI current controller. The PI speed controller was tuned by Naslin polynomial method as well as the `systune` Matlab’s function. Presented results clearly indicate that including the communication delay is essential to proper tuning of speed controller. The simulation and experimental results validate the correctness of the designed controllers.

**Acknowledgment:** This study was financed from founds for statutory activity of Electric Faculty of Warsaw

---

PRZEGŁĄD ELEKTROTECHNICZNY, ISSN 0033-2097, R. 91 NR 9/2015

293
University of Technology. The funds have been provided under the grant titled "Sterowanie dwusilnikowym ukладem napędowym dla miejskiego pojazdu elektrycznego" (Control of a two-motor drive system for an urban electric vehicle). The experiments were performed in the ECO-Mobility project laboratory setup (WND-POIG.01.03.01-14-154/09, 2009-2013).

Appendix

M-script for the PI speed controller tuning:

```matlab
ST0 = slTuner('PI_Speed','PI Controller');
addPoint(ST0,'load','wm');
addPoint(ST0,'wm');
Req = TuningGoal.StepRejection('load','wm,0.06,0.15,0.5);
Options = systuneOptions('RandomStart',10);
[ST1,Soft,~,Info] = ST0.systune([ Req ],Options);
```

PI speed – the simulation model.
PI Controller – name of PI speed controller.
load – name of disturbance input.
wm – speed measurement.
Req – the requirement sets a minimum standard for rejecting step disturbances.

[ST1,Soft,~,Info] - the systune output.

REFERENCES

Authors: mgr inż. Piotr Rumniak, prof. dr hab. inż. Lech M. Grzesiak, dr inż. Bartłomiej Ufnalski, Institute of Control and Industrial Electronics, Faculty of Electrical Engineering, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warszawa, Poland, email: piotr.rumniak@ee.pw.edu.pl