A Variable Frequency Angle-Based Repetitive Control for Torque Ripple Reduction in PMSMs

M.Tang, A.Gaeta†, A.Formentini*,P.Zanchetta**

* *The University of Nottingham, University Park, Nottingham, UK, eexmt10@nottingham.ac.uk*
† *reDrives s.r.l., via Tommaso Fazello 5, Lentini (SR), Italy, alberto.gaeta@redrives.eu*

Keywords: deadbeat control, repetitive control, PMSM, torque ripple, variable frequency

Abstract

This paper presents a novel method for torque ripple reduction in PMSM drives at variable speed, using a combination of angle-based repetitive control and deadbeat current control. Based on the internal model principle, repetitive control is capable to reduce periodic torque ripple by generating a compensating action that consequently need to be synchronized with the original ripple. The time to angle, angle to time conversions for repetitive control, which improve both the stability and the performance when the sampling frequency is not integer multiple of the speed, are presented. A transient detection strategy is also developed to allow a stable torque ripple reduction even during speed and load changes.

1 Introduction

Torque ripple reduction in permanent magnet synchronous machines (PMSMs) has been studied for decades. Using traditional vector control, since the output torque of the machine can be expressed as a function of current, many of the existing torque ripple suppression methods reduce torque ripple by introducing a tailored compensating periodic current ripple [1-5]. In some cases, current references are generated by lookup tables computed from measurements [6]. In any case the bandwidth of the current loop should be high enough to track such current references, which may contain high frequency components.

Besides, since torque ripple is mainly periodic [7], repetitive control (RC) can be a very powerful solution for periodic torque ripple reduction or periodic current references tracking, depending on where the repetitive controller is located. For example, repetitive control can work in conjunction with normal PI controllers in current loops to remove errors between periodic current references and current feedback [8]. Alternatively, the repetitive control can be placed in the speed loop where the controller can use the speed error [9-10] or the torque error [7,11] as an input producing periodic current references. Among these two methods, the latter one may be preferred to avoid computing current lookup tables. It is also to be noted that the values of periodic current references relate not only to the rotor position, but also to the level of saturation and parameter variations of the machine. It is therefore difficult, time and resource consuming to derive the right lookup tables for the different expected working conditions.

Repetitive control has been applied successfully for torque ripple reduction in some existing works in scientific literature [7, 9-11]. RC is effective on reducing any periodic torque ripple without necessarily knowing the machine parameters, no matter the ripple is caused by cogging torque, inverter nonlinearities [12], flux harmonics, current offset, current scaling error, or mechanical unbalance.

However, the knowledge of the frequency of torque ripple is necessary for a time-domain RC to maximize disturbance rejection. The time-domain RC would need to realign its memory length to the new period of the repetitive torque ripple each time the speed is changed. For this reason, the memory content should be reset and a new learning process should be started since the phase of the periodic disturbance is not maintained among speed changes.

Considering that PMSMs are very often used as variable speed drives and the torque ripple is in fact periodic with respect to rotor position, [10] presents an angle-based RC which works effectively within a limited variable speed range of 50 rpm to 70 rpm.

In the angle domain, the frequency of the previously described torque disturbance is fixed and the controller memory does not need to be resized when the speed is changed; the phase of torque disturbance due to cogging torque and flux harmonics within a full electrical rotation are marginally affected by the speed and by saturation conditions. Therefore the RC should learn only the amplitude variation of the torque disturbance as a consequence of a speed/load change and parameter variations.

The latter consideration can be extended to torque disturbances due inverter non-linearities, current offset or scaling error if the current angle is kept constant as usually happens in case of surface PMSMs when controlled at zero direct current to minimise losses.

However, none of these papers considered that conventional PI current control may need to be replaced by other high bandwidth current control where torque ripple reduction is demanded at high speed. Besides, the interaction between the speed loop PI controller and the repetitive controller should be considered for better performance. Also torque ripple reduction has been proved effective only for a reduced range of low speed.

Moreover, synchronization issues between the targeted system torque and the injected RC compensating action at largely variable speed, arising from the use of the speed feedback loop for torque ripple compensation, have never been addressed in any scientific work, according to authors' knowledge.

This paper proposes a novel combination of angle-based repetitive control and deadbeat current control (DBCC) [13] for torque ripple reduction in PMSMs, aimed for variable speed/load applications. Simulation tests have shown that this strategy significantly improves the synchronization of the generated RC compensation action at variable frequency, widely expands the frequency range of the torque ripple suppression, and improves performance not only at steady state, but also during transients. Therefore, high performance of the drive system can possibly be achieved without the extra cost for improving the manufacturing process, and even with worn shafts or unbalanced loads.

This paper is organised as follows. System models equations and diagrams of the proposed control are explained in detail in section 2. Simulation results are shown in section 3. The results and advantages of the proposed control are discussed in section 4. Finally, conclusions with discussion over the improvements introduced by the proposed approach are given in section 5.

2 Model and equations

The overall structure of the proposed system is depicted in figure 1, where the details of the proposed repetitive control are shown in figure 2. For a surface mounted PMSM, the output torque is considered proportional to only the q-axis current, while the d-axis current is set to zero.

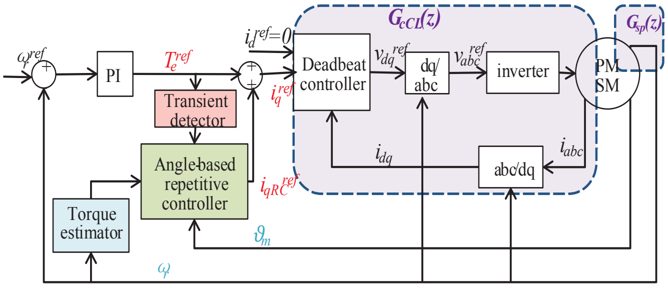
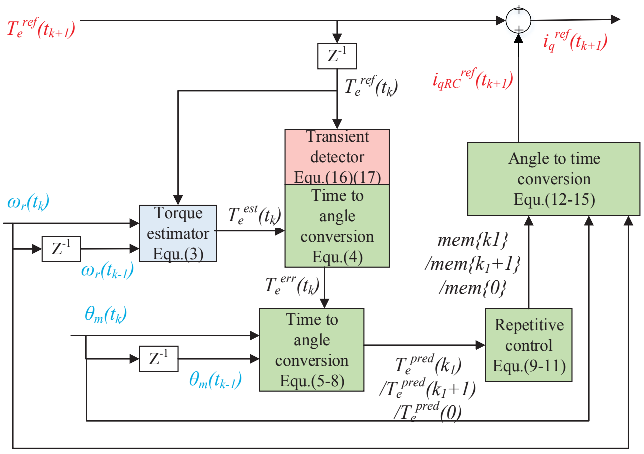
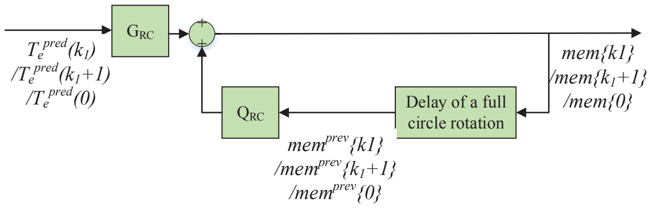


Figure 1: Block diagram for the proposed angle-based repetitive control and deadbeat current control



$T_e^{ref}(t_{k+1})$
Z^{-1}
$T_e^{ref}(t_k)$
+
+
$i_q^{ref}(t_{k+1})$
$i_{qRC}^{ref}(t_{k+1})$
Transient detector Equ.(16)(17)
Time to angle conversion Equ.(4)
Angle to time conversion Equ.(12-15)
$\omega_r(t_k)$
Torque estimator Equ.(3)
$T_e^{est}(t_k)$
Z^{-1}
$\omega_r(t_{k-1})$
$T_e^{err}(t_k)$
mem{k1} /mem{k_1+1} /mem{0}
$\theta_m(t_k)$
Time to angle conversion Equ.(5-8)
Repetitive control Equ.(9-11)
Z^{-1}
$\theta_m(t_{k-1})$
$T_e^{pred}(k_1)$ /$T_e^{pred}(k_1+1)$ /$T_e^{pred}(0)$

(a) Angle-based repetitive control in the control system



G_{RC}
+
+
$T_e^{pred}(k_1)$ /$T_e^{pred}(k_1+1)$ /$T_e^{pred}(0)$
mem{k1} /mem{k_1+1} /mem{0}
Q_{RC}
Delay of a full circle rotation
mem^{prev}\{k1\} /mem^{prev}\{k_1+1\} /mem^{prev}\{0\}

(b) Repetitive control (Equation (9-11))

Figure 2: Diagram of the angle-based repetitive control

2.1 Deadbeat current control loop

The deadbeat current control loop has been implemented as described in [6,13]. In closed loop it can be expressed by a delay of two sampling periods as in Equation (1) when operating within the voltage limitation determined by the inverter DC bus voltage.

$$G_{cCL}(z) = z^{-2} \tag{1}$$

2.2 Torque estimator

The discretized mechanical plant of the motor is described in Equation (2):

$$G_{sp}(z) = \frac{T_s}{Jz-(J-BT_s)} = \frac{\omega_r(z)}{T_e(z)-T_l} \tag{2}$$

Where, T_s is the sampling period, J is the motor and load inertia, B is friction factor, and T_l is the load torque.

The RC is connected with the speed loop PI using a "plug-in" structure. Assuming the pre-knowledge of the mechanical parameters (inertia J^{est} and friction factor B^{est}), the output torque is estimated and then used as an input for the RC. The output torque is estimated based on the reverse of Equation (2) as follow:

$$T_e^{est}(t_k) = \frac{J^{est}[\omega_r(t_k)-\omega_r(t_{k-1})]}{T_s} + B^{est}\omega_r(t_k) + T_e^{ref}(t_k) \tag{3}$$

Where, $T_e^{est}(t_k)$ is the estimated output torque at t_k, used as input for RC at t_{k+1}, $\omega_r(t_k)$ is the rotor angular speed at t_k; $\omega_r(t_{k-1})$is the rotor angular speed at t_{k-1}; the load torque is approximated using the value of the torque reference at t_k, $T_e^{ref}(t_k)$.

2.3 Time to angle conversion

Together with the estimated torque at t_k, $T_e^{est}(t_k)$, the torque reference $T_e^{ref}(t_k)$ generated by the speed loop PI controller at t_k, and the rotor mechanical position at t_k, $\theta_m(t_k)$, are all inputted to the repetitive controller at t_{k+1} which means at the beginning of the $(k+1)^{th}$ sampling period.

The torque error T_e^{err} at t_k is first calculated as in Equation (4):

$$T_e^{err}(t_k) = T_e^{ref}(t_k) - T_e^{est}(t_k) \tag{4}$$

The time to angle conversion is then executed. An array of length N, *mem*, is defined in the memory. The N values of the array *mem* are updated by the torque errors at only N chosen rotor angles out of the whole mechanical resolution of 2π and indicate the q-axis currents required for torque ripple reductions at the N chosen rotor locations. For example, $mem\{i\}$, the $(i+1)^{th}$ value in the array *mem*, (i is within $[0,N-1]$), is the q-axis current i_{qRC} required for reducing torque error when the rotor mechanical position is exactly $i*2\pi/N$. The rotor location index k_1 is defined as in Equation (5):

$$k_1 = \text{floor}\left(\frac{N\cdot\theta_m(t_k)}{2\pi}\right), k_1 \in [0, N-1] \tag{5}$$

Where, the value of N is chosen to be around the ratio between the sampling frequency f_s and the rated speed ω_r of the machine in Hz.

Since the rotor position cannot always be measured exactly at points coincident with integers multiple of $2\pi/N$, interpolations are necessarily used to predict the torque error at the closest chosen rotor locations ($\theta_m=0,\ 2\pi/N,\ 4\pi/N, \dots,\ 2\pi$). Linear interpolations are executed as in Equations (6)(7)(8) only if the value of k_1 is different from the one obtained at the previous sampling period.

In the case of positive rotation:

$$\begin{aligned} &T_e^{pred}(k_1) \\ &= T_e^{err}(t_{k-1}) + \frac{T_e^{err}(t_k) - T_e^{err}(t_{k-1})}{\theta_m(t_k) - \theta_m(t_{k-1})}\cdot[\theta_m(k_1) - \theta_m(t_{k-1})] \end{aligned} \tag{6}$$

In the case of negative rotation and $k_1 \neq (N-1)$:

$$\begin{aligned} &T_e^{pred}(k_1+1) = T_e^{err}(t_{k-1}) \\ &+ \frac{T_e^{err}(t_k) - T_e^{err}(t_{k-1})}{\theta_m(t_k) - \theta_m(t_{k-1})}\cdot[\theta_m(k_1) + 2\pi/N - \theta_m(t_{k-1})] \end{aligned} \tag{7}$$

In the case of negative rotation and $k_1=(N-1)$:

$$\begin{aligned} &T_e^{pred}(0) = T_e^{err}(t_{k-1}) \\ &+ \frac{T_e^{err}(t_k) - T_e^{err}(t_{k-1})}{\theta_m(t_k) - \theta_m(t_{k-1})}\cdot[\theta_m(k_1) + 2\pi/N - \theta_m(t_{k-1})] \end{aligned} \tag{8}$$

Where, $T_e^{pred}(k_1)$ is the torque error if the rotor mechanical angular position equals to $\theta_m(k_1)=k_1(2\pi)/N$, $T_e^{err}(t_k)$ and $T_e^{err}(t_{k-1})$ are the torque errors at t_k and t_{k-1}, $\theta_m(t_k)$ and $\theta_m(t_{k-1})$ are the rotor mechanical angular position at t_k and t_{k-1}. When the direction of the rotor movement is opposite to the positive direction of the encoder, thus negative rotation, $T_e^{pred}(k_1+1)$ is calculated instead of $T_e^{pred}(k_1)$. This is because the interpolation is more accurate estimating the torque error at $\theta_m(k_1+1)=2\pi(k_1+1)/N$, which is in between the current position $\theta_m(t_k)$ and the position at the previous period $\theta_m(t_{k-1})$, than estimating the torque error at $\theta_m(k_1)=k_1(2\pi)/N$, which has not been reached.

2.4 Repetitive control

After converting the torque error at the time t_k into the torque error at position $\theta_m(k_1)$ *or* $\theta_m(k_1+1)$ *or* $\theta_m(0)$, the memory array can now be updated according to the repetitive control Equations (9)(10)(11) (Figure 2(b)). Again, only when the rotor location index k_1 is different from the one at the previous sampling period, the memory array would need to be updated.

For positive rotation:

$$mem\{k_1\} = mem^{prev}\{k_1\}\cdot Q_{RC} + T_e^{pred}(k_1)\cdot G_{RC} \tag{9}$$

For negative rotation and $k_1 \neq (N-1)$:

$$mem\{k_1+1\} = mem^{prev}\{k_1+1\}\cdot Q_{RC} + T_e^{pred}(k_1+1)\cdot G_{RC} \tag{10}$$

For negative rotation and $k_1=(N-1)$:

$$mem\{0\} = mem^{prev}\{0\}\cdot Q_{RC} + T_e^{pred}(0)\cdot G_{RC} \tag{11}$$

Where, Q_{RC} is the forgetting factor of RC, G_{RC} is the gain of RC, $mem\{k_1\}$ is the $(k_1+1)^{th}$ value in the memory array *mem*, which indicates q-axis current required for compensating the torque ripple at rotor position $k_1*2\pi/N$, $mem^{prev}\{k_1\}$ is the previous value of $mem\{k_1\}$ obtained at the time of one full circle rotation ago.

2.5 Angle to time conversion

As shown in figure 2(a), the output of the repetitive controller after the angle to time conversion acts as an additional q-axis current reference, i_{qRC}^{ref}. Because of the delay of the current loop ($2T_s$), the current reference outputted at t_{k+1}, $i_{qRC}^{ref}(t_{k+1})$, will be achieved at t_{k+3}. The future rotor position at t_{k+3} is predicted and converted to be within $[0,\ 2\pi]$ as in Equation (12):

$$\theta_m^{fut}(t_{k+3}) = \text{mod}\left[\{\theta_m(t_k) + \omega_r(t_k)*3T_s\}, 2\pi\right] \tag{12}$$

A Future rotor location index k_2 is defined as in Equation (13):

$$k_2 = \text{floor}\left(\frac{N\cdot\theta_m^{fut}(t_{k+3})}{2\pi}\right), k_2 \in [0, N-1] \tag{13}$$

To compensate the overall delay of $3T_s$ ($2T_s$ for the current loop and $1T_s$ for processing time), the q-axis current reference generated by RC at t_{k+1}, $i_{qRC}^{ref}(t_{k+1})$, should be chosen from the memory array according to the future rotor mechanical position at t_{k+3}, $\theta_m(t_{k+3})$. Again, linear interpolations are used for estimating the value between $mem\{k_2\}$ and $mem\{k_2+1\}$, as in Equations (14)(15):

When $k_2 \neq (N-1)$:

$$\begin{aligned} &i_{qRC}^{ref}(t_{k+1}) = mem\{k_2\} \\ &+ \frac{N\cdot\left(mem\{k_2+1\} - mem\{k_2\}\right)}{2\pi}\cdot\left[\theta_m^{fut}(t_{k+3}) - \frac{k_2\cdot 2\pi}{N}\right] \end{aligned} \tag{14}$$

When $k_2=(N-1)$:

$$\begin{aligned} &i_{qRC}^{ref}(t_{k+1}) = mem\{k_2\} \\ &+ \frac{N\cdot\left(mem\{0\} - mem\{k_2\}\right)}{2\pi}\cdot\left[\theta_m^{fut}(t_{k+3}) - \frac{k_2\cdot 2\pi}{N}\right] \end{aligned} \tag{15}$$

2.6 Transient detector

The angle-based RC allows maintaining torque ripple compensation even during speed and load changes. In fact the learning process, after a transient, only needs to be updated with the small amplitude changes in the torque ripple due to the changed conditions (saturation of the machine for example) since the frequency and phase of the torque ripple are fixed with rotor position. However the learning process has to be temporarily disabled during each transient, to avoid the RC may output an excessive compensating reference current as a consequence of the non-existent large torque ripple estimated during each sudden change in speed. Therefore, a transient detection routine is designed to detect speed/load changes and disable the learning process accordingly.

In details, a speed transient will be detected once the change of the torque reference T_e^{ref} generated by the speed loop PI controller, ΔT_e^{ref}, is greater than a chosen threshold $\Delta T_e^{sudden\text{-}max}$ as in Equation (16):

Once $\Delta T_e^{ref}(t_k) = \left[T_e^{ref}(t_k) - T_e^{ref}(t_{k-1})\right] > \Delta T_e^{sudden-max}$:

$$T_e^{ref}(t_k) = 0 \qquad (16)$$

In addition, a load changes will be detected if the sum of ΔT_e^{ref}, ΔT_e^{sum} is larger than a chosen threshold $\Delta T_e^{sum\text{-}max}$ as in Equation (17):

Once $\Delta T_e^{sum}(t_k) = \left[\Delta T_e^{sum}(t_{k-1}) + \Delta T_e^{ref}(t_k)\right] > \Delta T_e^{sum-max}$:

$$T_e^{ref}(t_k) = 0 \qquad (17)$$

In both speed and load changes, the torque error T_e^{err} as defined in 2.3 is set to zero to block the learning procedure.

3 Simulation results

A PMSM model has been built in MATLAB/Simulink for simulating the effectiveness of the proposed method. It is worth noting that, in order to accurately simulate the non-sinusoidal flux distribution and cogging torque of PMSM in Simulink, two lookup tables, derived from a finite element (FE) model of the same motor, are used for the magnetic flux, and for the cogging torque. Parameters for RC are as Table 1:

Rated speed of motor ω_r	3000rpm=50Hz
Length of memory array N	300=15k/50
Sampling frequency f_s	15kHz
Sampling period T_s	66.7µs
Gain of RC G_{RC}	0.3
Forgetting factor of RC Q_{RC}	0.999
Threshold for detecting speed changes $\Delta T_e^{sudden\text{-}max}$	0.3Nm
Threshold for detecting load changes $\Delta T_e^{sum\text{-}max}$	0.3Nm

Table 1: Parameters of RC in simulation

Figure 3 shows a successful implementation of the deadbeat current control. The speed of the motor is kept at 100rpm and the q-axis current reference i_q^{ref} steps from 6A to 8A, the real q-axis current reaches the reference after 133µs ($=2T_s$) as designed.

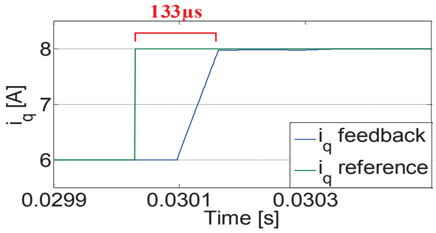


Figure 3 Performance of deadbeat current control

The resulting torque and speed waveforms with and without interpolation in the time to angle conversion are shown in Figure 4, where a speed change from 0 to 2995rpm (=49.917Hz) is applied at 0s and the load torque is set to be 5Nm. The speed 2995rpm is chosen since the sampling frequency 15kHz is then not integer multiple (300.5) of the speed. The torque and speed waveforms with and without the proposed transient detection strategy under speed/load changes are shown in Figure.5, where a speed change from 1500 to 3000rpm is applied at 1s and a load change from 2Nm to 5Nm is applied at 2s.

4 Discussion

Generally, the effectiveness of RC depends on whether the repetitive controller can learn or predict the torque ripple at different rotor locations precisely, and whether the compensation torque can be synthesized at exactly the desired rotor positions. Four innovative aspects of the proposed control method are discussed in the following together with the results in part 3.

4.1 Compensation of the current loop delays by position prediction

Current reference cannot be achieved instantaneously, but a delay is introduced as a consequence of the digital control implementation and depending on the particular type of current control used. The main advantage of DBCC compare to PI is that the delay of DBCC is fixed to be $2T_s$ in normal operations (no physical limits are hit) as shown in figure3. Therefore, the rotor position after the $2T_s$ delay can be easily predicted as Equation(12).

4.2 Compensation of the mechanical system delays by torque estimation

The low-pass filter effect of the mechanical system causes a phase shift between the rotor speed and the output torque of the machine. For a low-pass filter, high frequency signals are delayed more than the low frequency signals, so synchronization of the output of RC to target ripple reduction is more challenging at high speed. Since the torque is made up of components at different frequencies, the delay is difficult to compensate in RC. Alternatively, in the proposed control topology, the delay is compensated by the torque estimator as in Equation (3).

4.3 Improvements of RC learning accuracy

RC compensates the torque ripple based on the torque error it learns during previous operations. Therefore, the more accurate the torque errors are learned or recorded, the better the performance. Due to the floor function (Equation (5)), without the proposed interpolations, the compensation current saved in each memory location k_1 (or the torque error T_e^{err} used for calculating it) corresponds to a rotor position in the range $[k_1*2\pi/N, (k_1+1)*2\pi/N]$. With the worst case happening, when the sampling frequency and the rotor speed are not synchronised and the measured rotor position slowly shifts with time, the RC will generate a wrong compensation action, which in turn may increase the torque ripple, as shown in figure 4 when the sampling frequency is not integer multiple of the motor speed. Without enlarging the memory or increasing the sampling frequency, simply with interpolations, the performance of RC is improved.

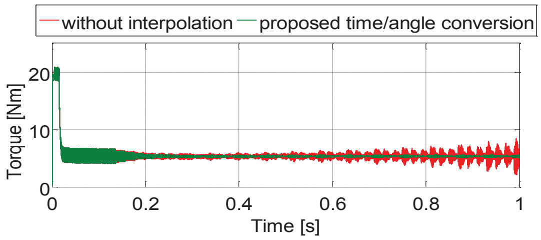
4.4 Rejection of oscillations during speed/load transients

In order to minimize interferences between the speed PI controller and the RC and minimize convergence time, the latter is connected in a plug-in configuration in the proposed control topology instead of a parallel implementation as it was done in [7,9-11].

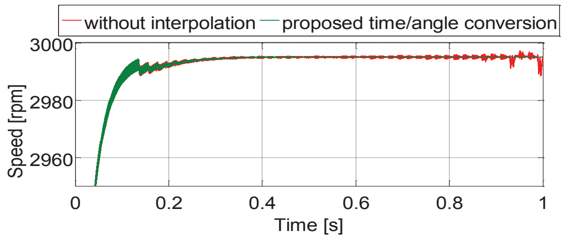
However, still, the interaction between speed PI and RC may cause oscillations during speed/load transients as shown in figure 5(green), where RC is kept on after being activated at 0.2s. Meanwhile, by properly disable only the learning process, the blue curve proves that the proposed transient detection strategy can reduce the oscillations while maintaining the reduction of torque ripple during the speed/load transients.

5 Conclusions

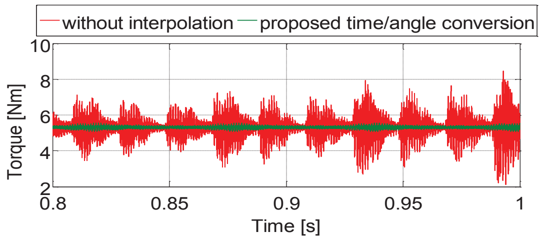
A novel combination of angle-based repetitive control (RC) and deadbeat current control (DBCC) is presented in this paper, aiming to effectively reducing periodic torque ripple under widely variable speed and even during speed/load transients. The DBCC, angle-based RC, torque estimator and the topology are chosen by considering the compensation of current control delays and the low-pass filter effect of mechanical system. Interpolations algorithms and a transient detection strategy are proposed to improve the accuracy of RC. The effectiveness of the proposed control topology is proven by simulation tests.



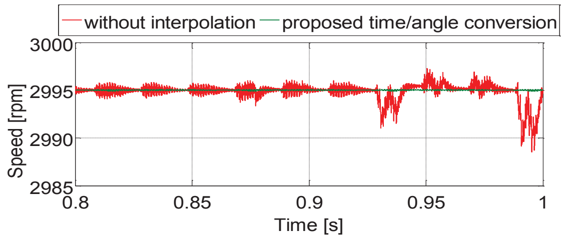
(a)Torque



(b)Speed

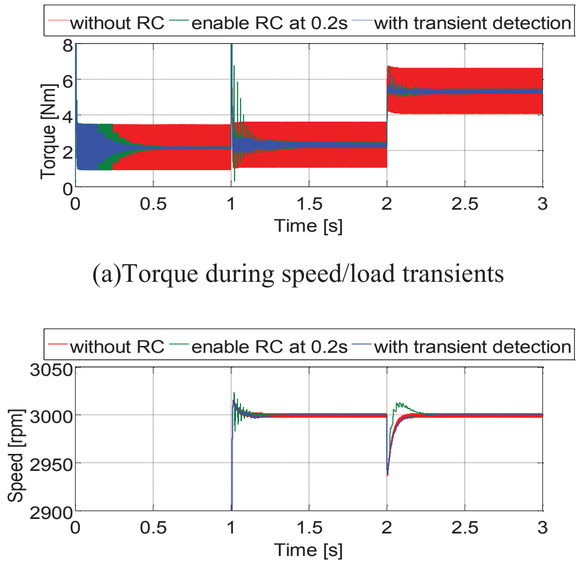


(c)Zoom in of (a)



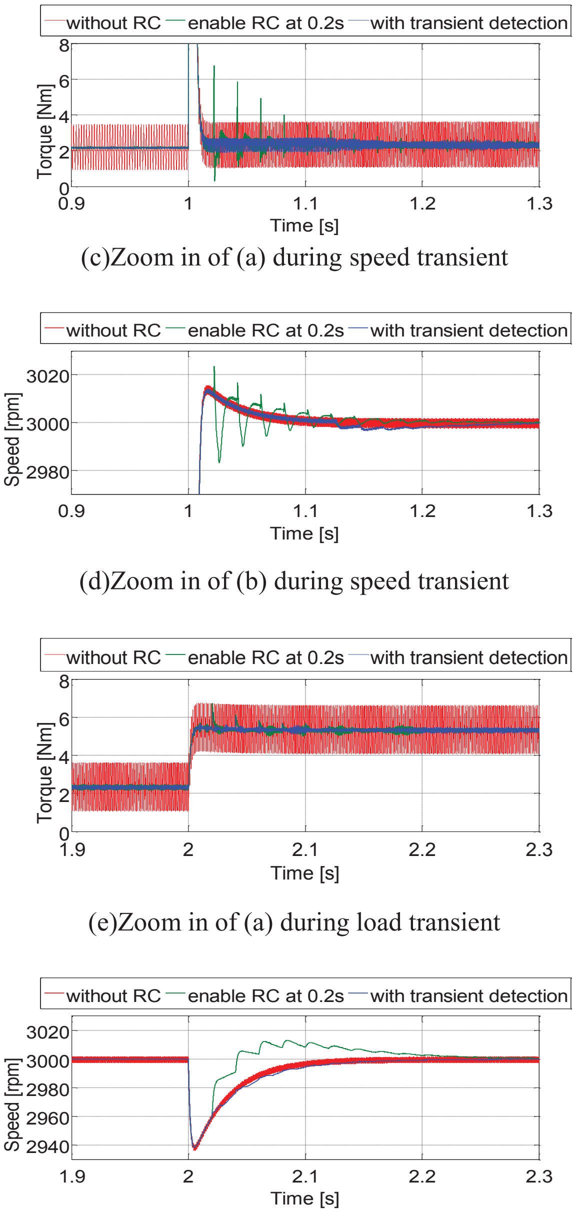
(d)Zoom in of (b)

Figure 4: The performance of the proposed time/angle conversion



(a)Torque during speed/load transients

(b)Speed during speed/load transients



(c)Zoom in of (a) during speed transient

(d)Zoom in of (b) during speed transient

(e)Zoom in of (a) during load transient

(f)Zoom in of (b) during load transient

Figure 5: The performance of the proposed transient detector

Acknowledgements

The authors wish to acknowledge the financial and technical support provided by the Shenzhen Best Motion Technology Limited, China.

References

[1] M Carpita, D Colombo, and A Monti, "A Generalised Approach to Torque Ripple Compensation In Permanent Magnet Machine Control," in *Industrial Electronics, 2000. ISIE 2000. Proceedings of the 2000 IEEE International Symposium on*, **vol.2**, pp. 396 – 400, (2000) .

[2] B.-J. Brunsbach, G. Henneberger, and T. Klepsch, "Compensation of torque ripple," in *Electrical Machines and Drives, 1993. Sixth International Conference on (Conf. Publ. No. 376)*, pp. 588 - 593, (1993).

[3] E. Favre, L. Cardoletti, and M. Jufer, "Permanent-magnet synchronous motors: a comprehensive approach to cogging torque suppression," *Industry Applications, IEEE Transactions on*, **vol. 29**, no. 6, pp. 1141 - 1149 , (1993).

[4] B H Lam, S K Panda, and J X Xu, "Torque Ripple Minimization in PM Synchronous Motors - An Iterative Learning Control Approach," in *Power Electronics and Drive Systems, 1999. PEDS '99. Proceedings of the IEEE 1999 International Conference on*, *vol. 1*, pp. 144 - 149, (1999).

[5] Lusu Guo and Leila Parsa, "Torque ripple reduction of the modular Interior Permanent Magnet machines using optimum current profiling technique," in *Electric Machines and Drives Conference, 2009. IEMDC '09. IEEE International*, pp. 1094 - 1099, (2009).

[6] J Holtz and L Springob, "Identification and compensation of torque ripple in high-precision permanent magnet motor drives," *Industrial Electronics, IEEE Transactions oon*, **vol. 43**, no. 2, pp. 309 - 320, (1996).

[7] Weizhe Qian, Sanjib K Panda, and Jian-Xin Xu, "Torque Ripple Minimization in PM Synchronous Motors Using Iterative Learning Control," *Power Electronics, IEEE Transactions on*, **vol. 19**, pp. 272 - 279, (2004).

[8] P. Mattavelli, L. Tubiana, and M. Zigliotto, "Torque-ripple reduction in PM synchronous motor drives using repetitive current control," *Power Electronics, IEEE Transactions on*, **vol. 20**, no. 6, pp. 1423 - 1431, (2005).

[9] Li Wenshan, Zhang Jian, and Li Yong, "A simpler and more efficient iterative learning controller for PMSM torque ripple reduction," in *Electrical Machines and Systems (ICEMS), 2013 International Conference on*, pp. 1231 – 1235, (2013).

[10] Y. Yuan, F. Auger, L. Loron, and S. Moisy, "Torque ripple reduction in Permanent Magnet Synchronous Machines using angle-based iterative learning control," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, pp. 2518 - 2523, (2012).

[11] Zheng Enrang, China Xi"an, and Qiao Wei, "A ILC scheme and passive filter for PMSM speed control to minimize pulating torque," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on (Volume:5)*, pp. V5-492 - V5-496, (2010).

[12] B. Lazhar, "On the compensation of dead time and zero-current crossing for a PWM-inverter-controlled AC servo drive," *IEEE Trans. Ind. Electron.*, **vol. 51**, no. 5, pp. 1113-1117, (2004).

[13] Lothar Springob and Joachim Holtz, "High-Bandwidth Current Control for Torque-Ripple Compensation in PM Synchronous Machines," *Industrial Electronics, IEEE Transactions on*, **vol. 45**, no. 5, pp. 713 - 721, (1998).