2-2015

Speed Error Mitigation for a DSP-Based Resolver-to-Digital Converter Using Auto-Tuning Filters

Nezar Abou Qamar
Southern Illinois University Carbondale, nezar@siu.edu

Constantine J. Hatziadoniu
Southern Illinois University Carbondale, hatz@siu.edu

Haibo Wang
Southern Illinois University Carbondale, zhwang@siu.edu

The paper is published in the IEEE Transactions on Industrial Electronics, Vol 62-2, Feb 2015, pp 1134-9.

Speed Error Mitigation for a DSP-Based Resolver-to-Digital Converter Using Auto-Tuning Filters

***Abstract*—Modern resolver-to-digital converters (RDC) are typically implemented using DSP techniques to reduce hardware footprint and enhanced system accuracy. However, in such implementations, both resolver sensor and ADC channel unbalances introduce significant errors particularly in the speed output of the tracking loop. The frequency spectrum of the output error is variable depending on the resolver mechanical velocity. This paper presents the design of an auto-tuning output filter based on the interpolation of pre-computed filters for a DSP-based RDC with a type-II tracking loop. A fourth-order peak and a second-order high pass filter are designed and tested for an experimental RDC. The experimental results demonstrate significant reduction of the peak-to-peak error in the estimated speed.**

***Indexing Terms*—Resolver sensor, adaptive filter design, speed control, ADC imbalances.**

I. Introduction

Resolvers are rotating transformers providing a robust and cost-effective means for angular position measurement and speed tracking in various applications, such as industrial drives, brake systems, robots, machine tools, radars and avionics [1]–[3]. Many industrial applications using asynchronous drives (ASD) still rely on encoders and resolvers because of their ability to reject common mode noise typically caused by PWM converters employed in ASD drives [4]. A resolver-to-digital converter (RDC) is a non-linear observer required to convert the analogue signals of the resolver into digital signals for the position and speed estimations. Modern RDC are implemented in DSP microcontrollers resulting in significant hardware reduction [1], [4], [5]. Closed loop implementations of RDC typically use a type-II tracking loop (such as in [2], [5], [6]). They are more advantageous than open-loop implementations (such as in [1], [4], [7], [8]), since they provide speed estimation along with position estimation and exhibit robust stability properties.

A major challenge in the RDC design is the elimination of resolver signal errors introduced by resolver quadrature and coil imbalances, amplifier imbalances, ADC offsets, ambient and quantization noise [3], [5]-[14]. These errors appear in the speed output of the RDC as oscillations of a frequency that is an integer multiple of the resolver mechanical frequency of rotation (e.g. a harmonic) and of a magnitude that is proportional to the resolver speed [12]-[14]. Calibration methods to deal with these errors based on the least squares as in [3] are more suitable for open loop systems. Self-tuning methods applied in angle tracking observers such as the one discussed in [9] assume a predefined mathematical expression of the resolver signals, where the magnitude and angle imbalances as well as the signal DC offset are used as the parameters of the signal equation. These parameters are, then, estimated online and the estimated values are used subsequently to estimate the resolver position and speed. The works in [5], [10] and [11] deal successfully with the 2^{nd} harmonic output oscillations introduced by quadrature, coil imbalance and amplifier gain errors. The technique employs two synchronously rotating frames (SRF) based on the Park's transformation for the positive and negative sequences. A low-pass filter (LPF), necessary for filtering the excitation frequency, is placed at the speed output, outside the tracking loop; the filter is required to have a low enough pass band in order that at low resolver speeds it effectively filters the oscillations on the RDC speed output due to previously

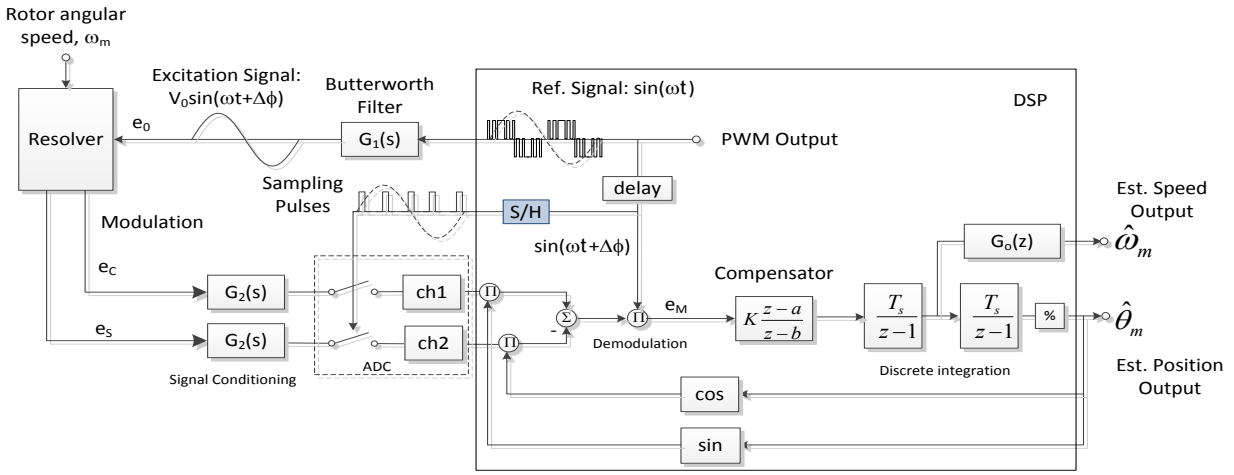

Fig. 1. System Model

mentioned resolver errors. In systems where the RDC speed output is used as a feedback for motor control, the LPF can introduce a significant phase lag, limiting the stability of the control loop. In addition, the algorithm deals only with the 2nd harmonic: the effect of higher order harmonics in the RDC error, e_M, while it is limited in the position output, $\hat{\theta}_m$, due to the two integrations between e_M and $\hat{\theta}_m$, it may become significant in the speed output, $\hat{\omega}_m$, due to only one integration between e_M and $\hat{\omega}_m$ (Fig. 1). This is especially so at high RDC speeds, since the magnitude of the error harmonics is proportional to the resolver speed [12]-[14]. This, therefore, may require additional pairs of SRF at higher than the 2nd order harmonic frequency, which will add a significant computational overhead due to the complex transformation computations required by each rotating frame.

In this paper, we present a DSP-based RDC implementation incorporating a type-II tracking loop. Novel auto-tuning filters are proposed to mitigate the resolver signal error on the speed estimation without adding phase-lag into the control loop. The filter design is independent of any particular mathematical representation of the resolver signals. The implementation of the proposed filters on practical DSP's requires only minimal computation time and memory size. Simulation and experimental results show that the developed RDC exhibits a stable and fast response with good precision for the estimated speed.

II. Design of the Auto-Tuning Filters

The RDC Tracking Loop. The representation of the DSP-resolver system is shown in Fig. 1, [6]. A sinusoidal-PWM signal is generated by the DSP microcontroller and filtered by a third order Butterworth LPF. The resulting sinusoidal signal is used to drive the resolver output. The resolver output signals are conditioned and level-shifted by G_2 before they are fed to the ADC channels where they are sampled at a rate of multiple times of the excitation frequency. The operations of sampling and demodulation are synchronized with the PWM reference signal after a fixed phase shift to match the delay introduced by the external circuits.

The mathematical model of the resolver is discussed in [11]. With reference to Fig. 1, the quadrature voltages of an *ideal* resolver are:

$$e_{C,S} = V\ sin(\omega t)\ cos\left(\theta_m - k\frac{\pi}{2}\right) \quad (1)$$

Where: e_C, e_S are determined by $k = 0\ and\ 1$ respectively, V is a constant that is determined by the resolver construction and the excitation magnitude, $\sin(\omega t)$ is the excitation signal, and θ_m is the mechanical position of the resolver, $\dot{\theta}_m = \omega_m$. With reference to the same figure, the demodulated control error signal e_M is given in (2) (normalized by $\frac{V}{2}$) [11]:

$$e_M = sin\left(\theta_m - \hat{\theta}_m\right)[1 - cos(2\omega t)] \quad (2)$$

If $\theta_m - \hat{\theta}_m$ is very small (a condition met near the lock state), (2) is approximated as:

$$e_M \approx \left(\theta_m - \hat{\theta}_m\right) - \left(\theta_m - \hat{\theta}_m\right) cos(2\omega t) \qquad (3)$$

With reference to Fig. 1, the control error, e_M, is filtered through a lead compensator and two discrete integrators. The lead compensator improves the loop stability. The compensator is designed based on the Nelder-Mead simplex method [15], which is available in MATLAB for non-linear systems [16], [17].

The Resolver Signal Errors. Coil imbalance, amplifier gain imbalance and dc offsets are the most significant sources of resolver signal error which may introduce estimation errors in the RDC output [1], [5], [6], [12]-[14]. We describe these errors onto the resolver signals in (1), adding a constant voltage, v_{dc}, to both e_S, e_C representing the dc offset, and the term, $v_g \sin(\omega t)\cos(\theta_m)$, on e_C representing coil and amplifier gain imbalances (called the gain error), where v_g is a constant depending on the degree of imbalance [13], [14]. The resolver signals modified by the above terms are given in (4) and (5):

$$e_S = V\, sin(\omega t)\, sin(\theta_m) + v_{dc} \qquad (4)$$

$$e_C = V\, sin(\omega t)\, cos(\theta_m) + v_{dc} + v_g\, sin(\omega t)\, cos(\theta_m) \qquad (5)$$

Approaching the steady state, $\theta_m \approx \omega_m t$, and the normalized control error, e_M, becomes:

$$e_M = \theta_m - \hat{\theta}_m + \frac{\sqrt{2}v_{dc}}{V} sin(\omega t)\, cos\left(\omega_m t + \frac{\pi}{4}\right) + \frac{v_g}{4V}(1 - cos(2\omega t))\, sin(2\omega_m t) \qquad (6)$$

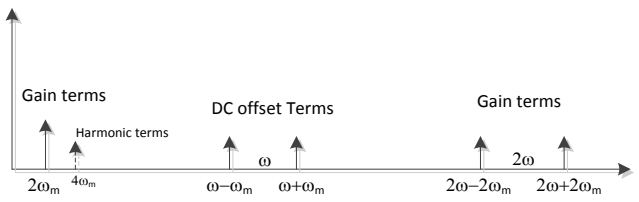

Fig. 2. Spectrum of the oscillating error terms.

The spectrum of the control error from (6) is shown in Fig. 2. The high frequency terms can be easily filtered from the output by the LPF. The low frequency terms, however, can introduce significant oscillations especially on the estimated speed. Low frequency harmonic terms of the order of $4\omega_m, 6\omega_m, \ldots$ with a lower magnitudes may also be present due to the quadrature imbalance. To deal with the low frequency terms a self-tuning filter is proposed for G_o.

Design of the Speed Output Filter, G_o. With reference to Fig. 1, the two integrators in the forward loop provide sufficient damping to the high frequency oscillatory terms in the speed and position responses introduced by the dc offset and gain error. However they do not have the same effect on the low frequency oscillatory terms introduced by the gain error, especially for the speed response. The speed output filter G_o is designed to deal with the remaining oscillatory terms. Since these oscillations are harmonics of ω_m, seen in Fig. 2, a filtration scheme with auto-tuning properties should be most effective. In this work, two different design approaches are presented resulting in (a) an auto-tuning peak filter; and (b) an auto-tuning high pass filter. The filter designs are presented next.

(a) *Auto-Tuning Peak Filter.* The auto-tuning peak filter is shown in Fig. 3. The idea is to utilize this filter to estimate the instantaneous amplitude of the speed harmonic ($\hat{h}$), and then subtract $\hat{h}$ from the unfiltered speed estimation ($\bar{\omega}_m$) to eliminate the harmonic.

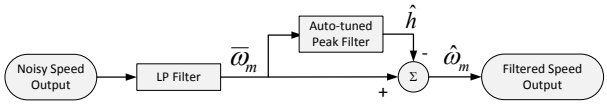

Fig. 3. Auto-tuning peak filter scheme.

The unfiltered speed estimation ($\bar{\omega}_m$) can be represented as shown in (7), where the second term is the oscillations introduced by the *n*-th harmonic By computing the filter center frequency, f_{center}, according to (8), the output of the filter, ($\hat{h}$), is then given by (9), where, the error term, e_{tuning}, is present due to the mismatch between f_{center} of the filter and the frequency of the *n*-th harmonic as a result of the

second term in (8). A perfect cancellation can be only achieved if the auto-tuning peak filter has a frequency response with 0^o phase shift and 0-dB magnitude at the frequency of the harmonic to be eliminated. Since this is practically impossible, it is therefore necessary to design the auto-tuning filter with a sufficient bandwidth to tolerate the error in the estimated harmonic frequency, such that e_{tuning} in (9) is reduced to a very small value.

$$\bar{\omega}_m = \omega_m + A\sin(n\omega_m t) \tag{7}$$

$$f_{center} = n \cdot \bar{\omega}_m = n\omega_m + nA\sin(n\omega_m t) \tag{8}$$

$$\hat{h} = A\sin(n\omega_m t) + e_{tuning} \tag{9}$$

In the discrete implementation of the filter, the maximum step size for updating f_{center} should be limited. A large step size for f_{center} may introduce oscillations in the filter response since the filter transfer function parameters will also change rapidly [18]. Alternatively, instead of defining a maximum step size, f_{center} can be tuned by passing (8) to a moving average or a low-pass filter. This will limit the step size and also attenuate the harmonic term in (8), which, in turn, will significantly reduce e_{tuning}.

For computational efficiency it is recommended to design offline a number of fixed peak filters for different values of f_{center} in the range of the RDC operation. These filters are then used in real time to derive a new peak filter with the new center frequency computed from (8) by interpolating between the poles of the fixed filters. For example, assume that two peak filters of 4th order have been designed with transfer function H_1 and H_2 given in (10) for $i = 1$ and 2 respectively. The filters have the same bandwidth and their center frequencies are respectively $f_{c1}$and f_{c2}. The interpolated filter transfer function, H_3, is given in (10) for $i = 3$ and its poles are given in (11) and (12), where the center frequency f_{c3} is given by (8).

$$H_i = K_i \frac{(z^{-2}-2z^{-1}+1)^2}{\prod_{j=1,2}[(z^{-1}-p_{ij})(z^{-1}-p_{ij}^*)]}, \text{ i=1,2,3} \tag{10}$$

Where: $p_{ij} = r_{ij}e^{i\alpha_{ij}}$. The interpolated filter parameters are given by:

$$r_{3j} = r_{1j} + (r_{2j} - r_{1j}) \cdot \frac{f_{c3}-f_{c1}}{f_{c2}-f_{c1}} \tag{11}$$

$$\alpha_{3j} = \alpha_{1j} + (\alpha_{2j} - \alpha_{1j}) \cdot \frac{f_{c3}-f_{c1}}{f_{c2}-f_{c1}} \tag{12}$$

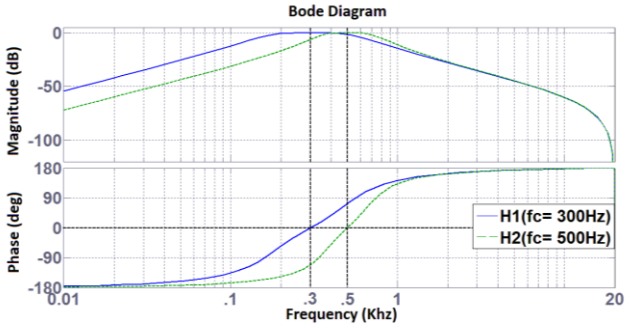

Fig. 4. Offline designed filters H_1 and H_2

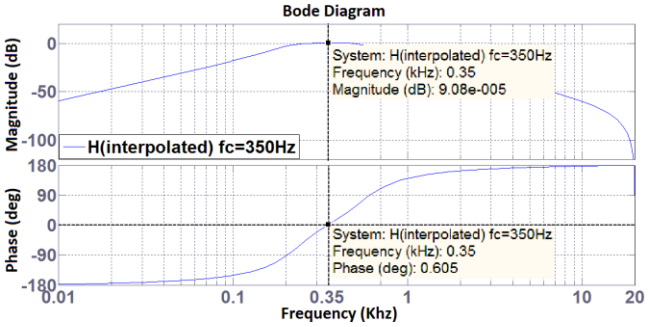

Fig. 5. Interpolate peak filter H_3 with a $f_{c3} = 350Hz$

To illustrate the accuracy of the procedure used for interpolation, Fig. 4 shows the magnitude and phase of two filers H_1 and H_2 with a f_{c1} and f_{c2} equal to 300 Hz and 500 Hz respectively. In Fig. 5, H_1 and H_2 have been used to interpolate a new filter H_3 with f_{c3} equal to 350 Hz based on (10)-(12). Note the accuracy of the interpolated filter in Fig. 5, having nearly 0 dB magnitude and 0.6° phase at the tune frequency. With this scheme, an additional auto-tuning peak filter is needed for each additional harmonic to be eliminated. An LPF with a high cut-off frequency may be added as in Fig. 3 to filter high frequency disturbances such as white and impulsive noise introduced respectively from the ADC quantization errors and ground currents from adjacent operating equipment. Having a high cut-off frequency, this LPF will not introduce a significant phase lag for control purposes. Furthermore, the LPF is necessary when the resolver carrier frequency is not significantly larger than the tracking loop bandwidth. In this case, the magnitude of the demodulation terms

is significant and the single integration before the speed output is not sufficient to eliminate them from the speed response.

(b) Auto-Tuning High Pass Filter. An auto-tuning high-pass filter of 2nd order is shown in Fig. 6. The filter pass band frequency f_{pass} [18] is increased proportionally to the resolver speed through the tuning function given in (13). This function uses the estimated resolver speed before filtration as the only tuning parameter and ensures inclusion of the harmonics in the pass band of the filter, within which the phase shift is approximately zero. Similar to the previous design, the estimated harmonic amplitude is subtracted from the unfiltered speed estimation to eliminate the harmonic.

$$f_{pass} = \begin{cases} 250\ Hz & ,\ \bar{f}_m < 50Hz \\ 250 + \left(\bar{f}_m - 50\right) \cdot 2\ Hz & ,\ \bar{f}_m > 50Hz \end{cases} \quad (13)$$

Given the offline-designed filters H_1 and H_2 with a transfer function given by (14) for $i = 1$ and 2 respectively, the interpolated filter H_3 has a transfer function given by (14) for $i = 3$ and its parameters are given by (15)-(17).

$$H_{\mathrm{i}} = K_{\mathrm{i}} \frac{z^{-2} - 2z^{-1} + 1}{(z^{-1} - p_{\mathrm{i}})(z^{-1} - p_{\mathrm{i}}^{*})}, i = 1,2,3 \quad (14)$$

Where: $p_{\mathrm{i}} = r_{\mathrm{i}} e^{i\propto_{\mathrm{i}}}$, and

$$r_3 = r_1 + (r_2 - r_1) \cdot \frac{f_{pass} - f_{pass1}}{f_{pass2} - f_{pass1}} \quad (15)$$

$$\alpha_3 = \alpha_1 + (\alpha_2 - \alpha_1) \cdot \frac{f_{pass} - f_{pass1}}{f_{pass2} - f_{pass1}} \quad (16)$$

$$K_3 = K_1 + (K_2 - K_1) \cdot \frac{f_{pass} - f_{pass1}}{f_{pass2} - f_{pass1}} \quad (17)$$

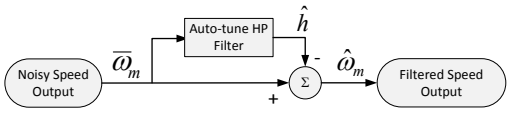

Fig. 6. Auto-tune high-pass filter scheme.

III. Experimental Results

The hardware set-up is shown in Fig. 7. The RDC was implemented on a 32-bit DSP microcontroller operating with a 150-MHz clock. With reference to Fig. 1, the resolver excitation is derived from the PWM module of the DSP device. For this purpose a two-level switching waveform of 42 switching operations per 10-kHz cycle was employed. The switching pattern in the waveform is a PWM signal designed using the method of selective harmonic elimination in order to eliminate up to the 19th harmonic [19] so that no low order harmonics be present in the resolver output. The DSP microcontroller used in this study allows to directly load memory data to the registers of the PWM module via direct memory access (DMA). This has been exploited to achieve 0 CPU load for the generation of the PWM pulses. The PWM switching times are stored (in terms of clock cycles) in the memory. The PWM qualifier is initiated to trigger a DMA transfer after a comparison event is true. Therefore, after each PWM switching operation, the DMA is triggered to update the comparator register to the next switching event from the stored switching times. With the DSP clocked at 150 MHz and using an ADC sampling and loop evaluation rate of 40 kHz, the maximum CPU load for the entire RDC algorithm without an output filter is 9.8%.

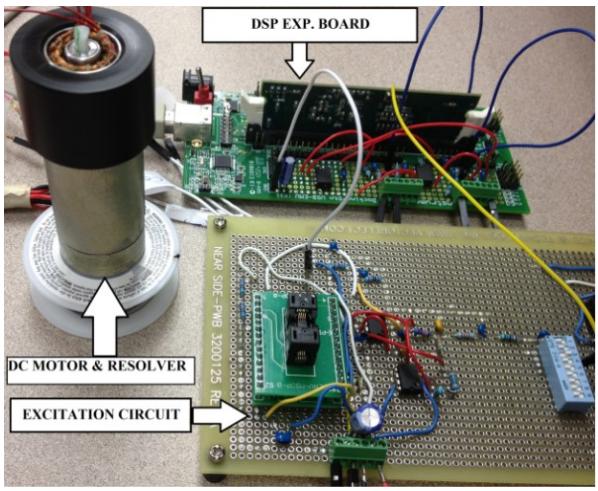

Fig. 7. Evaluation prototype.

Unfiltered Output: Initially the RDC is tested using a fixed LPF for G_o without auto-tuning properties for the purpose of filtering high frequency noise. Subsequently, the RDC response to a step position input was derived: the motor position was varied manually from -120° to $+120^{\circ}$ with steps of 30°. At each step, the RDC was engaged for the estimation. The position output responses to each step are shown in Fig. 8. The estimation error varies at different positions with the maximum error of $\pm 0.1°$ occurring at the position of about 90^0.

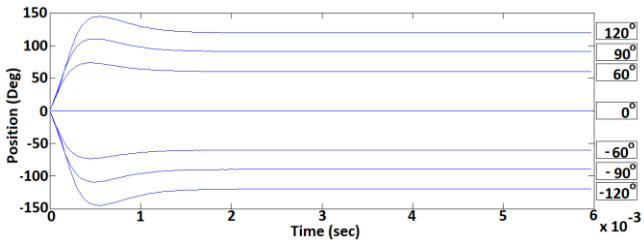

Fig. 8. Position response at -120^0 through 120^0.

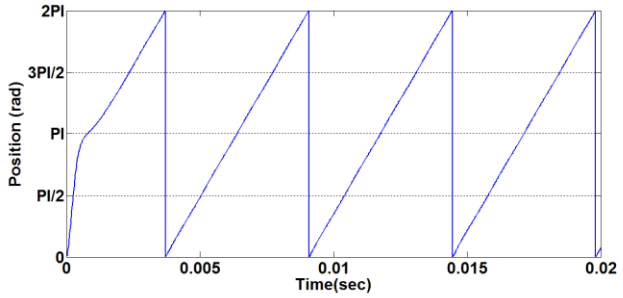

Fig. 9. Position response at speed 1920 rpm

Next, the RDC is engaged while the motor running at 10920 rpm. Fig. 9 shows the position output response and Fig. 10 shows the speed output response. The overshoot is relatively high due to the difference between the initial position estimation of the RDC, which is assumed zero and the actual resolver position. The oscillation in the speed output is due to resolver signal error and has peak-to-peak value of ±4.5% of the steady state speed. Finally, the dynamic response of the RDC was evaluated by engaging the RDC at the onset of a motor constant acceleration from zero to the maximum of 10920 rpm. The speed output response is shown in Fig. 11. The same steady state error is observed.

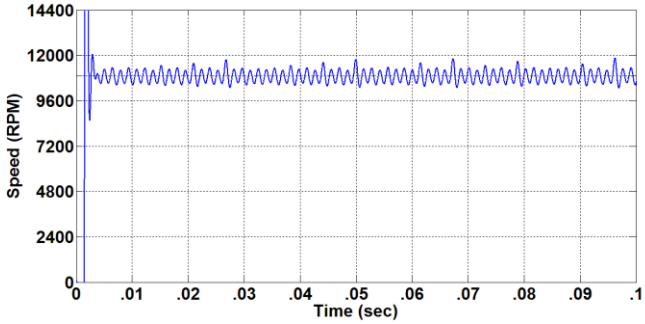

Fig. 10. Measured speed response at 10920 rpm.

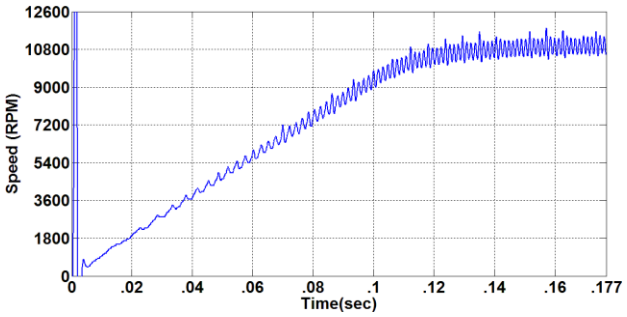

Fig.11 . Dynamic speed response under motor acceleration.

(a) *Response with an Auto-Tuning Peak Filter.* Now, a fourth-order auto-tuning peak filter with a pass-band of 200 Hz is designed for the speed output filter, $G_0(z)$. After engaging $G_0(z)$, the peak-to-peak error in speed estimation has dropped from ±4.5% to ±0.75% as shown in Fig. 12 and 13. With the DSP microcontroller clocked at 150 MHz, the maximum CPU load for the entire RDC algorithm is 24.2%, which includes 14% for the output filter, $G_0(z)$.

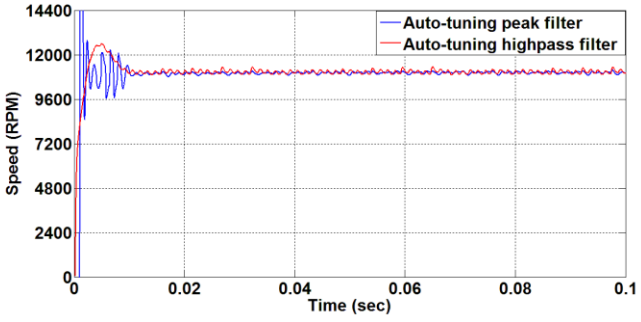

Fig. 12. Step speed at 10920 rpm: Auto-tuning peak filter.

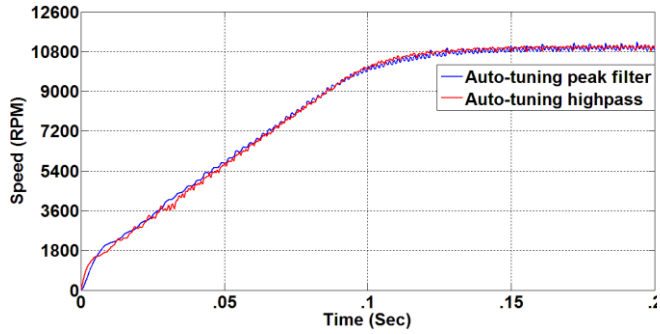

Fig. 13. Tracking of the DC motor speed during motor acceleration using an auto-tuning peak filter.

(b) Response with an Auto-Tuning High Pass Filter. A second order auto-tuned HP filter was designed for $G_0(z)$ based on (14)-(17) in order to deal with the multiple output harmonics noticeable in the speed output. The filter gain of its pass band is $A_{pass} = 0.001\ dB$. The step speed response and the response under motor acceleration are shown in Fig. 12 and 13 respectively. The peak-to-peak error is ±1%.

This filter has lesser damping performance than the peak filter, however, it only adds 7.9% to the CPU load. In addition, a single HP filter can be designed to deal with multiple harmonics, making this filter an attractive alternative solution, if CPU load is critical. Finally using look-up tables with the interpolated parameter values in (15)-(17), the CPU load can be further decreased to 4%. Table 1 compares the overall RDC performance under each filter type.

Table 1: Auto-Tuning Peak and HP Filter Comparison

Performance Parameter	Auto-tuning HP filter	Auto-tuning peak filter
Total CPU Load	17.7% (7.9% for the filter)	24.2% (14% for the filter)
Response Speed ($t_{steady-state}$)	12ms	9ms
Number of filters	1	Number of filtered harmonics

IV. Conclusion and Discussion

The paper presents an RDC design using auto-tuning filters at the speed output to damp harmonic oscillations due to resolver signal errors. The design method is independent of any signal model or parameter estimation of the resolver. Of the two filter types presented in the paper, the auto-tuning peak filter requires more CPU load but it is very effective around a narrow frequency band. Therefore, multiple peak filters are needed, if multiple frequency bands are present in the speed output. The auto-tuning high-pass filter is a simpler alternative design resulting in slightly less damping performance but a lower CPU load and an extended frequency range.

Finally, the above auto-tuning filters can also be used to filter the position estimation. However, due to the negligible position estimation error, in contrast with the significant error in speed, a position filter was not implemented in this work.

V. Acknowledgements

The work was supported by a grant from the NSF I\UCR Center of Embedded Systems at SIUC.

VI. References

[1] S. Sarma, V. K. Agrawal, and S. Udupa, "Software-Based Resolver-to-Digital Conversion Using a DSP," *Ind. Electron. IEEE Trans. On*, vol. 55, no. 1, pp. 371–379, Jan. 2008.

[2] R. Hoseinnezhad, "Position sensing in brake-by-wire callipers using resolvers," *Veh. Technol. IEEE Trans. On*, vol. 55, no. 3, pp. 924–932, May 2006.

[3] R. Hoseinnezhad, A. Bab-Hadiashar, and P. Harding, "Calibration of Resolver Sensors in Electromechanical Braking Systems: A Modified Recursive Weighted Least-Squares Approach," *Ind. Electron. IEEE Trans. On*, vol. 54, no. 2, pp. 1052–1060, Apr. 2007.

[4] D. A. Khaburi, "Software-Based Resolver-to-Digital Converter for DSP-Based Drives Using an Improved Angle-Tracking Observer," *Instrum. Meas. IEEE Trans. On*, vol. 61, no. 4, pp. 922–929, Apr. 2012.

[5] J. Bergas-Jané, C. Ferrater-Simón, G. Gross, R. Ramírez-Pisco, S. Galceran-Arellano, and J. Rull-Duran, "High-Accuracy All-Digital Resolver-to-Digital Conversion," *Ind. Electron. IEEE Trans. On*, vol. 59, no. 1, pp. 326–333, Jan. 2012.

[6] Seon-Hwan Hwang, Hyun-Jin Kim, Jang-Mok Kim, Liming Liu, and Hui Li, "Compensation of Amplitude Imbalance and Imperfect Quadrature in Resolver Signals for PMSM Drives," *Ind. Appl. IEEE Trans. On*, vol. 47, no. 1, pp. 134–143, Feb. 2011.

[7] L. Ben-Brahim, M. Benammar, and M. A. Alhamadi, "A Resolver Angle Estimator Based on Its Excitation

Signal," *Ind. Electron. IEEE Trans. On*, vol. 56, no. 2, pp. 574–580, Feb. 2009.
[8] M. Benammar, L. Ben-Brahim, and M. A. Alhamadi, "A high precision resolver-to-DC converter," *Instrum. Meas. IEEE Trans. On*, vol. 54, no. 6, pp. 2289–2296, Dec. 2005.
[9] J. Faber, "Self-calibration and noise reduction of resolver sensor in servo drive application," in *ELECTRO, 2012*, 2012, pp. 174–178.
[10] P. Rodriguez, J. Pou, J. Bergas, J. I. Candela, R. P. Burgos, and D. Boroyevich, "Decoupled Double Synchronous Reference Frame PLL for Power Converters Control," *Power Electron. IEEE Trans. On*, vol. 22, no. 2, pp. 584–592, Mar. 2007.
[11] G. Gross, M. Teixido, A. Sudria, and J. Bergas, "All-digital resolver-to-digital conversion," *Power Electron. Appl. 2005 Eur. Conf. On*, p. 8 pp.–P.8, 2005.
[12] C. Attaianese and G. Tomasso, "Position Measurement in Industrial Drives by Means of Low-Cost Resolver-to-Digital Converter," *Instrum. Meas. IEEE Trans. On*, vol. 56, no. 6, pp. 2155–2159, Dec. 2007.
[13] D. C. Hanselman, "Resolver signal requirements for high accuracy resolver-to-digital conversion," *Ind. Electron. IEEE Trans. On*, vol. 37, no. 6, pp. 556–561, Dec. 1990.
[14] Hanselman, D.C., "Techniques for improving resolver-to-digital conversion accuracy," *Industrial Electronics, IEEE Transactions on* , vol.38, no.6, pp.501,504, Dec 1991
[15] J.A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.
[16] D. Xue, Y. Q. Chen, and D. P. Atherton, "PID Controller Design," in *Linear Feedback Control*, Philadelphia, PA, 2007, pp. 216–221.
[17] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, 1998.
[18] J.G. Proakis and D.G. Manolakis, 'Desing of Digital filters' in 'Digital Signal Processing: principles, algorithms and applications', third edition, Prentice Hall, 1996, pp619-620.
[19] H. S. Patel and R. G. Hoft, "Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverter: Part I-Harmonic Elimination," *IEEE Trans. Ind. Appl.*, vol. IA-9, pp. 310-317, May 1973.

VII. Biographies