Complex Signal Processing for Coriolis Mass Flow Metering in Two-Phase Flow

Ming Li, Manus Henry, University of Oxford.

Corresponding Author: Manus Henry

Department of Engineering Science

Parks Road, Oxford OX1 3PJ.

manus.henry@eng.ox.ac.uk

Abstract

This paper presents new complex signal processing methods, Complex Bandpass Filtering (CBF), Complex Notch Filtering (CNF) and their combination (CBF-CNF), applied to the Coriolis Mass Flowmeter (CMFM). CBF, CNF and CBF-CNF can be utilized to suppress the negative frequency component of each sensor signal in order to produce the corresponding analytic form with reduced tracking delay. Further processing of the analytic form yields the amplitude, frequency, phase and phase difference of the sensor signals. In comparison with previously published methods, the new techniques offer short delay, high noise suppression, high accuracy and low computational cost. A reduced delay is useful in CMFM signal processing especially for maintaining flowtube oscillation in two/multi-phase flow conditions. The central frequency and the frequency range for the CBF, CNF and CBF-CNF methods are selectable so that they can be customized for different flowtube designs.

Keywords: Complex signal processing, complex bandpass filter, parameter estimation, Coriolis mass flow meter

1. Introduction

The Coriolis Mass Flow Meter (CMFM) provides direct and accurate measurement of the mass flow and (usually) the density of a single-phase fluid. This capability has led to its widespread industrial application over several decades [1]. A CMFM has two primary components: the flowtube, with typically two velocity sensors, and at least one driver to induce flowtube oscillation; and the transmitter, an embedded system for signal processing, measurement and control. Recently, CMFM applications have been extended to two-phase, gas/liquid mixtures [2]–[4]. The signal processing requirements for CMFM are challenging, especially in two-phase conditions [5], [6], entailing the tracking of amplitude, frequency, phase and phase difference simultaneously for the essentially sinusoidal signals.

For example, Fig. 1 compares a sensor signal when monitoring the flow of a single phase liquid with another sensor signal when a moderate level of gas (15% gas volume fraction) is included in the liquid flow to create a two-phase mixture. In the lower graph the flowtube damping is constantly changing, as are the mixture flow rate and density. These changes result in much greater variation in frequency, amplitude and phase difference in the corresponding CMFM sensor signals than for single phase flow (upper plot). Accordingly, signal processing for CMFM in two-phase conditions is more challenging than for a single phase fluid.

One aspect of CMFM operation less frequently discussed in the literature is the time delay between sensor input and driver output. Since a primary requirement for CMFM operation is to maintain flowtube oscillation, usually at a fixed amplitude [7], a digitally synthesized driver signal is generated to be in a fixed phase relation with a sensor signal (or often the sum – and hence mid-phase – of the two sensor signals), depending on the sensor type. For velocity sensors the required phase offset is 180° (or 0°), but for acceleration sensor the required phase offset is 270° (or 90°) [8]. In this process, time delay plays an important role in determining the quality of the amplitude control. With single phase flow, the time delay in the flow tube control system is

less important as the sensor signals are inherently more stable. But with two-phase flow, as illustrated in Fig. 1, time delay in the flowtube control system may become a more significant factor in regulating the rapidly changing oscillation. A further problem caused by two-phase flow is the rise in mechanical damping on the flowtube. As shown by comparing the signal amplitudes in the upper and lower plots of Fig. 1, higher damping often results in a lower amplitude of oscillation, as the maximum drive energy is usually limited. With lower and varying amplitude, and higher levels of background noise, the Signal-to-Noise Ratio (SNR) is significantly reduced in two-phase flow, and so noise suppression is also important.

With a digital drive, time lags occur between the sensor and driver signals due to Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC) operations, signal processing and phase synchronization. Fig. 7 in [9] shows a typical timing diagram for the drive generation process, where a large proportion of the delay is due to the measurement process itself, i.e. the extraction of frequency, phase and amplitude information from the sensor signal. Minimizing this delay will improve the responsiveness of the amplitude control, and indeed the flow measurement itself, an increasingly important issue in a number of applications [10], [11].

Current methods for CMFM signal processing include the Hilbert Transform (HT) [12]–[15], dual quadrature demodulation [16], [17], digital correlation [18], and a combination of the Adaptive Notch Filter (ANF) for frequency tracking together with the Discrete-Time Fourier Transform (DTFT) for amplitude and phase calculation [19]–[21]. These methods work well under single-phase, steady flow conditions since all the parameters are relatively stable. But there has been little previous discussion in the literature about dynamic response and performance in two-phase flow conditions [5]. Prism signal processing [22] is a new technique based on recursive FIR filtering: it has been applied in Coriolis metering to monitor automobile fuel injection pulses as short as 1 ms [10], thus demonstrating the potential for fast dynamic response. However, Prism signal processing has not yet been applied to the two-phase flow problem.

In summary, accurately tracking CMFM sensor signals, when subject to time-varying amplitude and frequency, and while minimizing measurement delay, are important challenges for the next generation of two-phase flow capable CMFM.

Given these requirements, a technique based on complex signal processing, the Complex Bandpass Filter (CBF), was presented by the current authors in [6]. In this paper, we expand this approach to describe the Complex Notch Filter (CNF) and the Complex Bandpass Filter with Complex Notch Filter (CBF-CNF) as new techniques with application to CMFM. The key idea is to suppress the negative frequency of the sensor signal in order to generate the corresponding analytic form, from which each required parameter value can be calculated. Furthermore, given their natural bandpass property, the new techniques need not require pre-filtering, thus reducing time delay and introducing an ability to suppress noise.

In section 2, the complex signal processing is described for each of the CBF, CNF and CBF-CNF methods. In section 3, the new techniques are compared with current methods. Simulation results with quantified error performances are presented. In Section 4, the complexity of each algorithm is given. Finally, in section 5, the findings are summarized.

2. Complex Signal Processing

2.1. Complex Bandpass Filter

The CBF is derived from a conventional low-pass filter design (for example an IIR elliptic filter) through the application of a complex shift factor $e^{j\theta}$ to the filter coefficients. Define the original IIR filter transfer function as:

$$H_r(z) = \frac{\sum_{m=0}^{P} b_m z^{-m}}{\sum_{n=0}^{Q} a_n z^{-n}} \tag{1}$$

where P and Q are the numerator and denominator order, and b_m and a_n are the filter coefficients. If $Q = 0$, the filter is Finite Impulse Response (FIR); otherwise, it is Infinite Impulse Response (IIR).

Applying the complex shift factor $e^{j\theta}$ to $H_r(z)$:

$$H_r(z) \xrightarrow{z^{-1}=e^{j\theta}z^{-1}} H_c(z) = \frac{\sum_{m=0}^{P} b_m e^{j\theta m} z^{-m}}{\sum_{n=0}^{Q} a_n e^{j\theta n} z^{-n}} \tag{2}$$

The numerator and denominator orders P and Q are unchanged, while the filter coefficients become $b_m e^{j\theta m}$ and $a_n e^{j\theta n}$.

The complex shift rotates the zeros and poles of the original filter by an angle θ in the z-plane in the anti-clockwise direction. Fig. 2 shows the resulting zero and pole rotation for an exemplary 5^{th} order elliptic filter.

The rotation only changes the angle of the poles and zeros in the z-domain, while the radius and relative positions remain the same. Accordingly, the effect on the filter properties is to induce a shift of θ radians/sample in the magnitude response, as shown in Fig. 3, converting the original low pass filter into a bandpass filter.

When a real signal is passed through the CBF, the negative frequency component is filtered out, resulting in a single-sided analytic form. The filtering process in the frequency domain is shown in Fig. 4, where the original input is a real double-sided sinusoidal signal with frequency ω_0.

After obtaining the analytic form of the filtered signal, the calculation of amplitude, frequency, phase are similar to those used in the Hilbert transform method [12]. For CFM signal processing, where two sensor signals must be tracked, the phase difference is readily calculated from the phases of the individual signals. Assume the CFM sensor signals take the form:

$$\begin{aligned} x_1 &= A_1 \sin(\omega t + \phi/2) \\ x_2 &= A_2 \sin(\omega t - \phi/2) \end{aligned} \tag{3}$$

where A_1 and A_2 are the amplitudes of the two sensor signals, ω is their common frequency, and ϕ is the phase difference between them. Passing the signals through identical CBF filters, complex analytic signals are obtained as follows:

$$\begin{aligned} x_{1a} &= A_1[\cos(\omega t + \phi/2) + i\sin(\omega t + \phi/2)]/2 = A_1 e^{j(\omega t+\phi/2)}/2 \\ x_{2a} &= A_2[\cos(\omega t - \phi/2) + i\sin(\omega t - \phi/2)]/2 = A_2 e^{j(\omega t-\phi/2)}/2 \end{aligned} \tag{4}$$

The amplitudes can be obtained by taking conjugate terms:

$$x_{1a} \times \overline{x_{2a}} = A_1 e^{j(\omega t+\phi/2)} \times A_2 e^{-j(\omega t-\phi/2)}/4 = A_1 \times A_2 e^{j\phi}/4 \tag{5}$$

For brevity we assume here that the two input signals have the same amplitude ($A_1 = A_2 = A$); this is a reasonable first approximation for the CMFM, and further computational steps can be taken to account for any sensor imbalance. Equation (5) is thus simplified to:

$$x_{1a} \times \overline{x_{2a}} = A^2 e^{j\phi}/4 \tag{6}$$

The phase difference may now be calculated using:

$$\phi = \arg(x_{1a} \times \overline{x_{2a}}) \tag{7}$$

while the amplitude can be obtained using:

$$A_1 = A_2 = 2|x_{1a}| = 2|x_{2a}| \tag{8}$$

The frequency may be derived from the change of phase over a number of samples. For example, consider adjacent samples from the first sensor signal:

$$\begin{aligned} \overline{x_{1a}(\mathrm{n}-1)} \times x_{1a}(\mathrm{n}) &= A e^{-j(\omega t_{n-1}+\phi/2)} \times A e^{j(\omega t_n+\phi/2)}/4 \\ &= A^2 e^{j(\omega t_n - \omega t_{n-1})}/4 \end{aligned} \tag{9}$$

followed by:

$$\omega t_n - \omega t_{n-1} = \arg(\overline{x_{1a}(\mathrm{n}-1)} \times x_{1a}(\mathrm{n})) \tag{10}$$

Converting into Hz:

$$f = \frac{(\omega t_n - \omega t_{n-1}) \times f_s}{2\pi} \tag{11}$$

where f_s is the sampling frequency. Using adjacent samples to calculate frequency provides a short response time but is susceptible to noise; comparing the phase shift over a larger number of samples is equally possible, leading to a tradeoff between measurement noise and response time.

By selecting the 'shift' frequency and passband of the CBF, the range of frequencies that may be tracked can be matched to the resonant frequency range of the corresponding flowtube.

2.2. Complex Notch Filter

Another way to generate the analytic form of a real sinusoidal signal is to apply a Complex Notch Filter (CNF) to filter out the negative frequency component. CNF is advantageous in that the tracking delay may be close to zero at the retained positive frequency. The CNF is designed such that the notch frequency is located at the negative frequency of the real sensor signal. For a CFM, where a range of frequencies may be output from a flowtube, a bandstop filter can be created by shifting a high-pass filter to the negative side of the frequency domain. In this case, the transfer function of the CNF follows equation (2) but the shift angle becomes -θ which means zeros and poles rotates clock-wise z-plane. The transfer function becomes:

$$H_r(z) \xrightarrow{z^{-1}=e^{-j\theta}z^{-1}} H_c(z) = \frac{\sum_{m=0}^{P} b_m e^{-j\theta m} z^{-m}}{\sum_{n=0}^{Q} a_n e^{-j\theta n} z^{-n}} \tag{12}$$

Fig. 5 shows the zeros and poles from a 5th-order elliptic high-pass filter rotated to form a CNF with notch at ω_c = -0.1 rad/s.

In Fig. 6, the CNF's group delay at the corresponding positive frequencies (around 0.1 rad/s) is shown to be approximately 2 samples. Generally, it would be expected that the CNF tracking delay should be significantly reduced compared to that of the CBF. A comparison via simulation is given in Section 3.4 below.

CNF may be extended to form a comb filter to remove harmonic noise, e.g. a mains frequency (50 or 60 Hz) along with its harmonics. By adding notches at a series of harmonics, the CNF can be turned into a comb structure while still retaining a small delay at the positive tracking frequency.

One disadvantage of CNF is that it provides limited noise reduction compared with the bandpass characteristic of the CBF. This occurs because only the notched negative frequency is attenuated, while all other noise components are passed through into the sinusoid parameter tracking calculations.

2.3. Complex Bandpass Filter with Complex Notch Filter

While CBF has good noise suppression but a relatively large tracking delay, CNF is sensitive to noise but has a short tracking delay. This suggests it may be possible to achieve a trade-off between noise performance and tracking delay for complex signal processing by cascading CBF and CNF together (CBF-CNF). In this combined approach (Fig. 7), the design of the CBF section need not be as constrained in terms of passband, roll-off, and stop band attenuation, because the CNF ensures good rejection of the negative frequency component. This method has the potential to reduce tracking time delay and so reduce errors during dynamic change. Here we combined a 3^{rd}-order CBF with a 4^{th}-order CNF to form a CBF-CNF for use in simulations.

3. Simulation

In order to evaluate performance, the CBF and CBF-CNF techniques are compared with established sinusoidal tracking methods using one simulation of continuous two-phase flow conditions, and a second simulation of empty-full batching.

3.1. Current Methods

Two current techniques are used for comparison purposes: ANF for frequency tracking combined with the DTFT for amplitude and phase calculation [19]–[21], which is denoted as DTFT (ANF) here; and the Hilbert Transform (HT) [12]–[15]. The sampling frequency f_s is 2 kHz, which is sufficiently higher than the simulated CMFM vibrating frequency at around 100 Hz.

For DTFT (ANF), we follow the technique described in [13] which uses the Steiglitz-McBride ANF (SMM-ANF) structure. The transfer function is

$$H(z) = \frac{\widehat{A}_n(z^{-1})}{\widehat{A}_n(\rho z^{-1})} = \prod_{k=1}^{m} \frac{1 + \widehat{\alpha}_k(n) z^{-1} + z^{-2}}{1 + \rho \widehat{\alpha}_k(n) z^{-1} + \rho^2 z^{-2}} \tag{13}$$

where *m is* the track number i.e. the number of peak frequencies to be tracked. Since the CMFM signal has a single dominant frequency, *m*=1. ρ is the pole contraction factor which determines the bandwidth of the ANF, given by:

$$BW = 2\cos^{-1}\left(\frac{2\rho}{1+\rho^2}\right) \tag{14}$$

The weight coefficient $\widehat{\alpha}_k(n) = -2\cos\widehat{\omega}_k(n)$, with $\widehat{\omega}_k(n)$ the notch frequency estimate of the input frequency, is adjusted by a Recursive Mean Square (RMS) algorithm:

$$\begin{aligned}
&\hat{\alpha}_k(n+1) = \hat{\alpha}_k(n) - P(n)\psi(n)e_s(n) \\
&\psi(n) = \frac{\partial e_s(n)}{\partial \hat{\alpha}_k(n)} = \frac{y(n-1) - \rho e_s(n-1)}{1 + \rho\hat{\alpha}_k(n)z^{-1} + \rho^2 z^{-2}} \\
&P(n) = \frac{(1-\lambda)P(n-1)}{\lambda + \psi^2(n)P(n-1)}
\end{aligned} \tag{15}$$

where $\psi(n)$ is the gradient function, $P(n)$ represents the covariance parameters, λ is the forgetting factor, and $e_s(n)$ represents the output of ANF.

When implementing the SMM-ANF algorithm in the simulation described below, the initial values are:

$$\rho = 0.9, P(1) = P(2) = 100, \lambda = 0.9 \tag{16}$$

After obtaining the frequency, the amplitude and phase difference can be calculated using a recursive DTFT:

$$\begin{aligned}
DFT_R(n,\omega_k) = &\left[x(n+N/2-1) - x(n-N/2-1)\right](-1)^k e^{\frac{j2\pi\omega_k}{N}} \\
&+ DFT_R(n-1,k)e^{\frac{j2\pi\omega_k}{N}}
\end{aligned} \tag{17}$$

where *x(n)* is the input signal function, *N* is the window length, *k* is the sampling point ranging from $0, 1, \cdots, N/2\text{-}1$ and ω_k is the estimated frequency from ANF at k^{th} sampling point.

For the HT, we follow the method presented in [12] using Parks-McClellan FIR filter design method to form a 49^{th} -order FIR filter. Passing the input signal through the HT generates an imaginary signal orthogonal to the original real input signal, and so the analytic form is obtained.

3.2. Two-phase flow simulation

In order to simulate the sensor signals arising in two-phase flow conditions, a Random Walk Model (RWM) was proposed by Tu *et al.* [13]. However, the technique places no limits on the instantaneous change of the parameter values. In a real CMFM, the rate of change of each parameter is physically limited due to mechanical inertia, limited fluid velocities and so on. Here we introduce filtering of the parameter changes in order to create a more realistic simulation – the Modified Random Walk Model (MRWM). We define parameters as follows:

$$
\begin{aligned}
y_1(n) &= A(n)\sin[\omega(n)n/f_s + \phi(n)/2] + \sigma_{e1}\cdot e_1(n) \\
y_2(n) &= A(n)\sin[\omega(n)n/f_s - \phi(n)/2] + \sigma_{e2}\cdot e_2(n) \\
A(n) &= \frac{(A_f(n) - \min(A_f(n)))\times(A_{\max} - A_{\min})}{\max(A_f(n)) - \min(A_f(n))} + A_{\min} \\
\omega(n) &= \frac{(\omega_f(n) - \min(\omega_f(n)))\times(\omega_{\max} - \omega_{\min})}{\max(\omega_f(n)) - \min(\omega_f(n))} + \omega_{\min} \\
\phi(n) &= \frac{(\phi_f(n) - \min(\phi_f(n)))\times(\phi_{\max} - \phi_{\min})}{\max(\phi_f(n)) - \min(\phi_f(n))} + \phi_{\min} \\
A_f(n) &= h_A(n) * e_A(n) \\
\omega_f(n) &= h_\omega(n) * e_\omega(n) \\
\phi_f(n) &= h_\phi(n) * e_\phi(n)
\end{aligned}
\tag{18}
$$

where $e_1(n)$ and $e_2(n)$ are un-correlated white noise sequences, $e_A(n)$, $e_\omega(n)$ and $e_\phi(n)$ are uniformly distributed random noise signals in the interval (-1,1), and σ_{e1}, σ_{e2} are gain factors for the input noise. $A_{\max}$, $A_{\min}$ are the upper and lower limits for the time-varying amplitude, and where $\omega_{\max}$, $\omega_{\min}$, $\phi_{\max}$ and $\phi_{\min}$ are the corresponding limits for frequency and phase difference respectively. $h_A(n)$, $h_\omega(n)$ and $h_\phi(n)$ are low-pass filters to limit the rate of change of parameter values and the '*' operator is the time domain convolution process for filtering.

The bound-limited amplitude, frequency and phase difference values are generated via uniform random noise processes and passed through a low-pass filter to constrain the rate of change. Table I shows parameter values for MRWM used for simulation in this paper. Then, based on the filtered parameter values, the two sensor signals are generated. Fig. 8 compares the output of

MRWM with that of RWM. In Fig. 8, both the RWM and MRWM sequences range over the desired parameter values in a random fashion. However, MRWM excludes high frequency variations which are physically unrealistic, but which will influence the error statistics and hence the performance evaluation of the measurement techniques.

3.3. Simulation Results

Based on signals generated using MRWM, simulations of CBF and CBF-CNF have been carried out alongside the DTFT (ANF) and HT methods. Fig. 9 shows typical sensor data supplied to each of the tracking algorithms over 1 s period; this is similar to the example of real two-phase flow sensor data shown in Fig. 1.

As discussed above, the CNF used in isolation is sensitive to noise, and so it is not included in this simulation study. Accordingly, the CBF, CBF-CNF, DTFT (ANF) and HT methods are tested using the time-varying input shown in Fig. 9. In further simulations, white noise with standard deviation σ is added to the sensor signals. With $\sigma = 5$ mV and a time varying amplitude the average SNR is approximately 20 dB.

Figs. 10-12 show the methods' tracking performance for noise-free conditions while Figs. 13-15 show the corresponding performance (in terms of residual error) with the addition of noise. From the figures, CBF and CBF-CNF outperform DTFT (ANF) and the Hilbert method, having a smaller tracking delay and a better dynamic response. CBF-CNF also has the smallest tracking delay and shows better tracking in the noise-free case. For the noisy case, since CBF is designed to have deeper stopband attenuation, the tracking result is smoother especially for the frequency parameter.

3.4. Quantification of error performance

To evaluate each method's performance numerically, the Root Mean Squared Error (RMSE) is applied, calculated using:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}(i) - Y(i))^2} \tag{19}$$

where $Y(i)$ and $\hat{Y}(i)$ are the true and estimated values. Table 2 compares the performance of each method for both the noise free and σ = 5 mV simulations.

From Table 2, CBF-CNF performs best in the noise-free simulation and in the σ = 5 mV simulation except for frequency tracking. This is because the frequency calculation is susceptible to high noise and requires deeper stop-band attenuation to track frequency well. However the CBF technique outperforms all the current methods, particularly with added noise.

3.5. Batching From Empty Simulation

One particularly challenging condition for maintaining good measurement tracking and flowtube control occurs during a rapid transition from empty to full or from full to empty. This condition typically arises in a batching operation, where the Coriolis meter starts and ends empty, and is required to report the total flow of liquid in the batch [10]. Such a transition, with the associated changes in mass flow, density and flowtube damping, is likely to result in simultaneous changes in frequency, amplitude and phase difference on the two sensor signals, over a period as short as 0.5 s.

All three new techniques have been tested alongside the DTFT (ANF) and HT methods in a start of batch simulation. Here an empty-to-full flowtube filling process is simulated via otherwise noise-free sensor signals where the common frequency drops linearly from 100 Hz down to 85 Hz, the amplitudes drops linearly from 0.3 V down to 0.05 V and the phase difference increases

linearly from 0° up to 4°, all simultaneously over 0.5 s. The simulation sample rate is 2 kHz. Fig. 16 shows the parameter values used for the sensor signals during the empty-full simulation.

Figs 17-19 show the residual errors for each method, while Table 3 shows the RMSE error and the approximate tracking time delay. From these results, CBF, CNF and CBF-CNF all perform better than current methods with reduced error and/or tracking delay.

4. Complexity

Further key aspects of algorithm assessment include computational complexity and data storage requirements, as CMFM measurement algorithms need to be implemented in embedded systems and to run in real-time. Table 4 compares the current and new techniques in terms of the static memory size required for buffering data and storing variables, and the number of additions (or subtractions) and multiplications (or divisions) required to process each new sample. The assessment of complexity includes the full calculation from pre-filtering through to the estimation of frequency, amplitude and phase difference. It excludes however the arctangent calculation used once per sample that is common to all the methods listed.

From Table 4, it can be seen that the CBF, CNF and CBF-CNF techniques have relatively low computational requirements. This will support the use of the newly developed techniques in real-time, low cost, implementations.

5. Conclusion

This paper has described three complex bandpass filtering methods and has applied them to CMFM signal processing. The CBF can be derived from a simple low-pass filter with a selectable central frequency and bandwidth. The calculation is simple and computational cost is

small. Due to the nature of bandpass filtering, the CBF can not only track amplitude, frequency and phase difference at the same time, but it also applies noise suppression, which is increasingly important in CMFM applications. Additionally, CBF combined with CNF has been used to reduce tracking delay and improve accuracy. Simulation studies suggest the tracking performance of these two methods is generally superior to that of the DTFT (ANF) and Hilbert transform techniques. The computational cost and size is also less than that of the existing techniques.

A future publication will describe experimental results in which the real-time measurement and control performance of the CBF and CBF-CNF algorithms have been compared with that of a commercially applied algorithm over a range of two-phase flow conditions. These results will further demonstrate that CBF and CBF-CNF offer a good signal processing solution to meet future CMFM signal processing requirements.

References

[1] T. Wang and R. Baker, "Coriolis flowmeters: a review of developments over the past 20 years, and an assessment of the state of the art and likely future directions," *Flow Meas. Instrum.*, vol. 40, pp. 99–123, Dec. 2014.

[2] M. Henry, M. Tombs, and F. Zhou, "Field experience of well testing using multiphase Coriolis metering," *Flow Meas. Instrum.*, vol. 52, pp. 121–136, Dec. 2016.

[3] M. Tombs, F. Zhou, and M. Henry, "Two-phase coriolis mass flow metering with high viscosity oil," *Flow Meas. Instrum.*, vol. 59, pp. 23–27, Mar. 2018.

[4] Q.-L. Hou, K.-J. Xu, M. Fang, Y. Shi, B.-B. Tao, and R.-W. Jiang, "Gas–Liquid Two-Phase Flow Correction Method for Digital CMF," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 10, pp. 2396–2404, Oct. 2014.

[5] M. Li and M. P. Henry, “Signal Processing Methods for Coriolis Mass Flow Metering in Two-Phase Flow Conditions,” *2016 IEEE Int. Conf. Ind. Technol.*, 2016, pp. 690–695.

[6] M. Li and M. Henry, “Complex bandpass filtering for Coriolis mass flow meter signal processing,” *IECON 2016 - 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, 2016, pp. 4952–4957.

[7] S. Wang, D. Zheng, S. Fan, P. Wang, and W. Wang, “Experimental study on the closed-loop control system of Coriolis mass flowmeter for oil-water two-phase flow measurement,” *2010 IEEE Symp. Ind. Electron. Appl.*, 2010, pp. 61–65.

[8] J. Kutin, A. Smrečnik, and I. Bajsić, “Phase-locking control of the Coriolis meter’s resonance frequency based on virtual instrumentation,” *Sensors Actuators A Phys.*, vol. 104, no. 1, pp. 86–93, Mar. 2003.

[9] M. Zamora and M. P. Henry, “An FPGA Implementation of a Digital Coriolis Mass Flow Metering Drive System,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 7, pp. 2820–2831, Jul. 2008.

[10] F. Leach, S. Karout, F. Zhou, M. Tombs, M. Davy, and M. Henry, “Fast Coriolis mass flow metering for monitoring diesel fuel injection,” *Flow Meas. Instrum.*, vol. 58, pp. 1–5, Dec. 2017.

[11] C. Clark, R. Cheesewright, and S. Wang, “Prediction of the Dynamic Performance of Fast Response Coriolis Meter Systems,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 95–99, Jan. 2008.

[12] G. R. Duffill, S. M. Jones, and A. T. Patten, “Meter electronics and methods for determining a liquid flow fraction in a gas flow material,” U.S. Patent 7974792, July 05, 2011.

[13] Y. Tu, H. Yang, H. Zhang, and X. Liu, “CMF Signal Processing Method Based on Feedback Corrected ANF and Hilbert Transformation,” *Meas. Sci. Rev.*, vol. 14, no. 1, pp. 41–47, 2014.

[14] A. Uehara, T. Hashizume, T. Wakui, A. Yoshino, A. Kadoguchi, and N. Miyaji, “Diagnosis of aerated flow at water line with Coriolis flowmeter using Hilbert transform,” *Proc. SICE Annu. Conf. 2010*, 2010, pp. 1682–1687.

[15] W. Lin and J.-L. Zhao, “A Novel Method Based on Hilbert Transform for Signal Processing of Coriolis Mass Flowmeter,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 02, p. 1858001, Feb. 2018.

[16] A. Svete, J. Kutin, G. Bobovnik, and I. Bajsić, “Theoretical and experimental investigations of flow pulsation effects in Coriolis mass flowmeters,” *J. Sound Vib.*, vol. 352, pp. 30–45, Sep. 2015.

[17] L. van de Ridder, W. B. J. Hakvoort, J. van Dijk, J. C. Lötters, and A. de Boer, “Quantification of the influence of external vibrations on the measurement error of a Coriolis mass-flow meter,” *Flow Meas. Instrum.*, vol. 40, pp. 39–49, Dec. 2014.

[18] J. Ruoff, W. Gauchel, and H. Kück, “Advances in Signal Acquisition and Signal Processing of Coriolis Flow Meters,” *Procedia Eng.*, vol. 87, pp. 1585–1588, Jan. 2014.

[19] T. Shen, Y. Tu, and H. Zhang, “A novel time varying signal processing method for Coriolis mass flowmeter,” *Rev. Sci. Instrum.*, vol. 85, no. 6, p. 065116, Jun. 2014.

[20] D. Feng, S. Fan, and D. Zheng, “A time-varying signal processing method for Coriolis mass flowmeter based on adaptive filter,” *Trans. Inst. Meas. Control*, vol. 40, no. 1, pp. 261–268, Jan. 2018.

[21] H. Kitami and H. Shimada, “Signal processing method, signal processing apparatus, and

coriolis flowmeter," U.S. Patent 8676517, March 18, 2014.

[22] M. Henry, F. Leach, M. Davy, O. Bushuev, M. Tombs, F. Zhou, and S. Karout, "The Prism: Efficient Signal Processing for the Internet of Things," *IEEE Ind. Electron. Mag.*, vol. 11, no. 4, pp. 22–32, Dec. 2017.

Tables

Parameters	Values
Sampling frequency (f_s)	2 kHz
Low-pass filter cut-off frequencies (f_A^s, f_ω^s, ω_ϕ^s)	6 Hz
Range for amplitude	A_{min} = 0.05 V, A_{max} = 0.3 V
Range for frequency	f_{min} = 85 Hz, f_{max} = 100 Hz
Range for phase difference	ϕ_{min} = 0°, ϕ_{max} = 4°

Table 1: Parameter Values for MRWM

RMSE in Noise Free Experiment				
Method and Parameter	DTFT (AANF)	Hilbert	CBF	CBF-CNF
Amplitude (V)	9.580e-4	6.786e-4	4.574e-4	2.983e-4
Frequency (Hz)	1.365e+0	5.294e-1	3.562e-1	2.052e-1
Phase Diff (°)	2.458e-1	1.404e-1	9.551e-2	6.076e-2
RMSE in σ = 5 mV Experiment				
Method and Parameter	DTFT (AANF)	Hilbert	CBF	CBF-CNF
Amplitude (V)	2.512e-2	2.903e-3	1.123e-3	1.031e-3
Frequency (Hz)	1.567e+0	2.280e+1	8.991e-1	1.185e+0
Phase Diff (°)	3.286e-1	1.944e-1	1.077e-1	7.756e-2

Table 2: Tracking Performance for Two-Phase Simulation

Method and Parameter	DTFT (AANF)	Hilbert	CBF	CNF	CBF-CNF
Amplitude (V)	2.64e-5	2.12e-5	9.28e-6	1.09e-6	2.02e-6
Frequency (Hz)	7.41e-1	6.68e-2	2.82e-3	3.35e-3	3.47e-3
Phase Diff (°)	9.87e-2	5.36e-3	2.48e-3	3.14e-4	5.30e-4
Tracking Delay (ms)	15	12.5	8.75	3.125	3.75

Table 3: Tracking Performance for Empty-Full Batching Simulation

Method	DTFT (AANF)	Hilbert	CBF	CNF	CBF-CNF
Additions	21	99	21	21	25
Multiplications	53	108	26	26	30
Static Storage (bytes)	432	800	320	320	448

Table 4: Algorithm Complexity

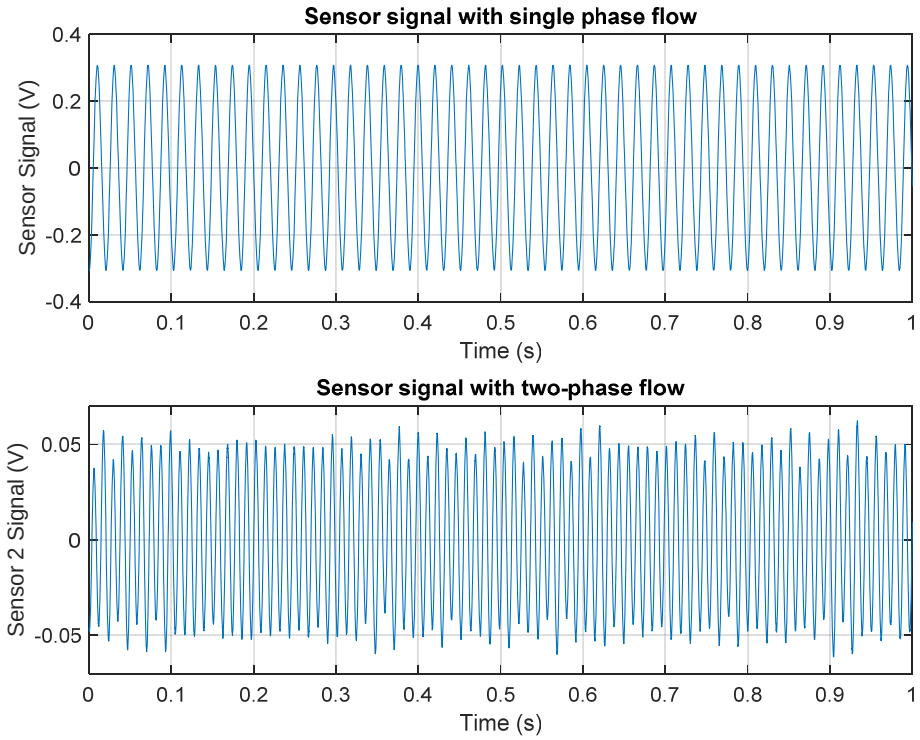

Figure 1. Coriolis Mass Flow meter sensor signals with single phase flow

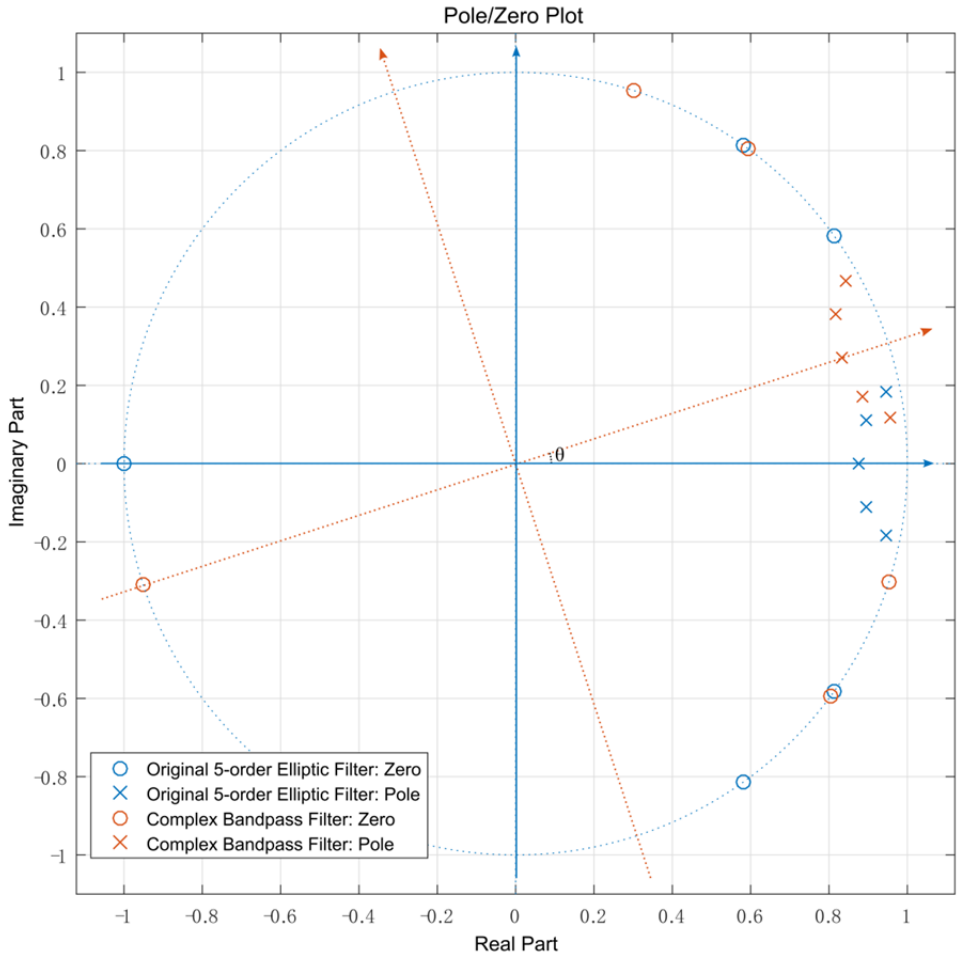

Figure 2. Zero and pole rotation for Complex Bandpass Filter

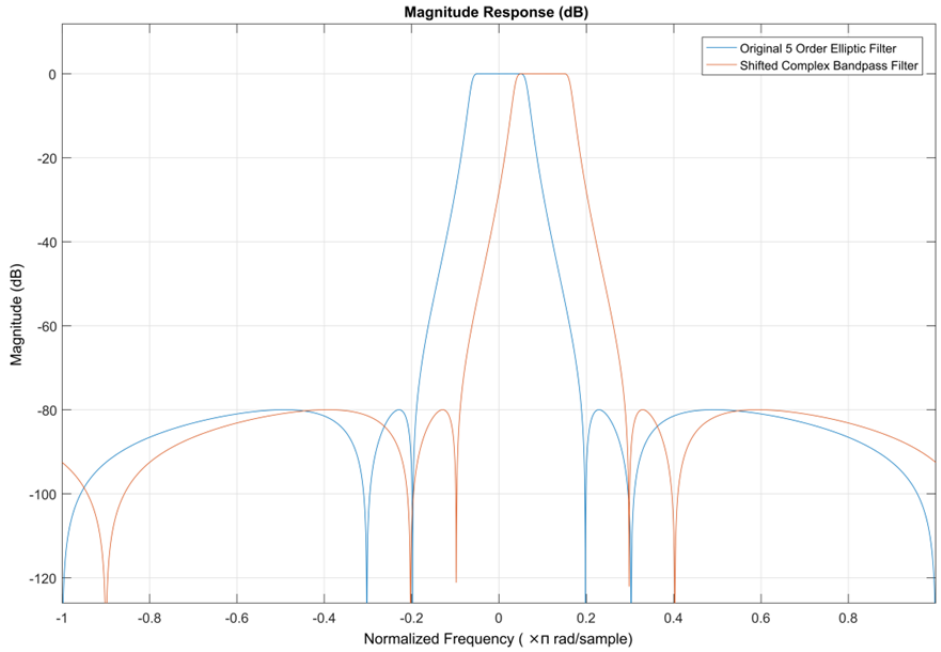

Figure 3. Magnitude Response of Elliptic filter and Complex Bandpass Filter

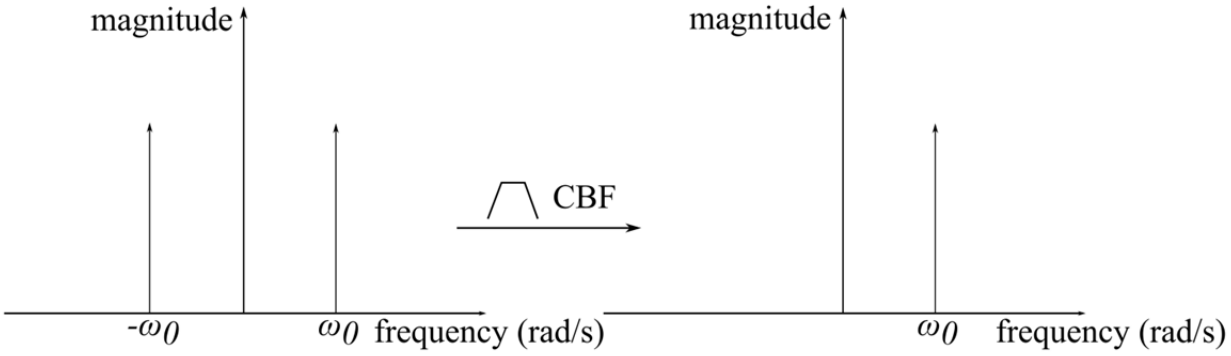

Figure 4. Filtering process for Complex Bandpass Filter

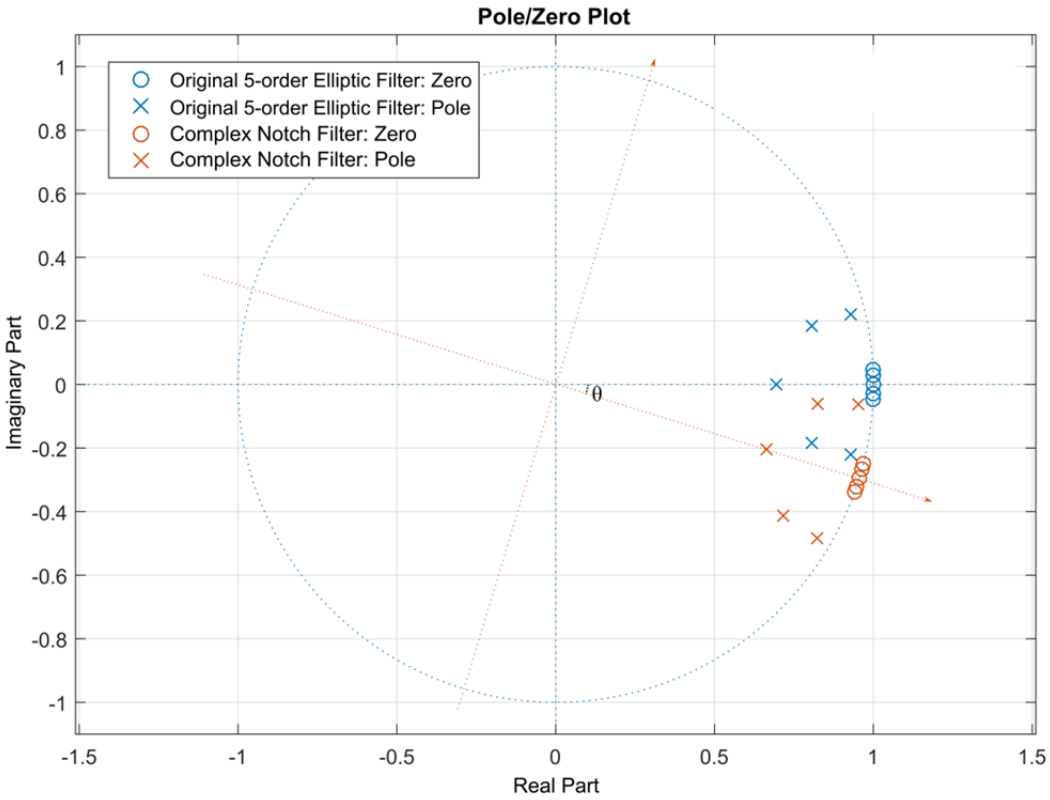

Figure 5. Zero and pole rotation for Complex Notch Filter

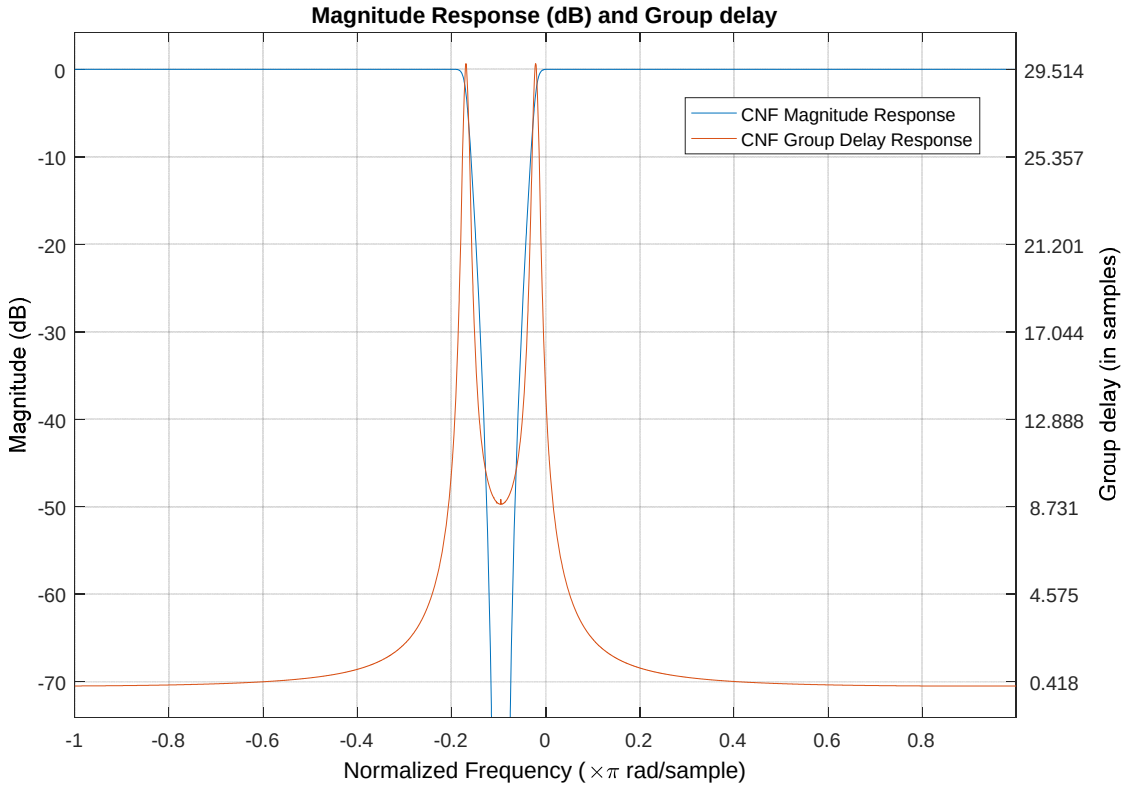

Figure 6. Group delay of 5th order Complex Notch Filter

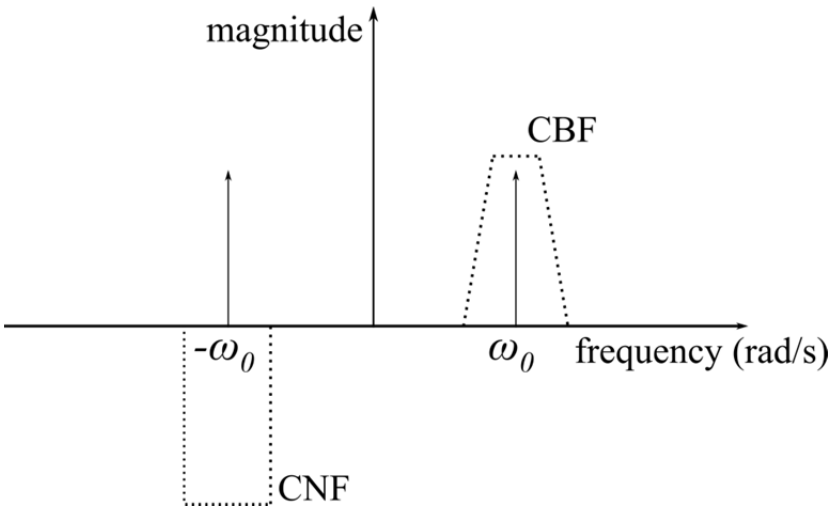

Figure 7. Filtering process for CBF-CNF

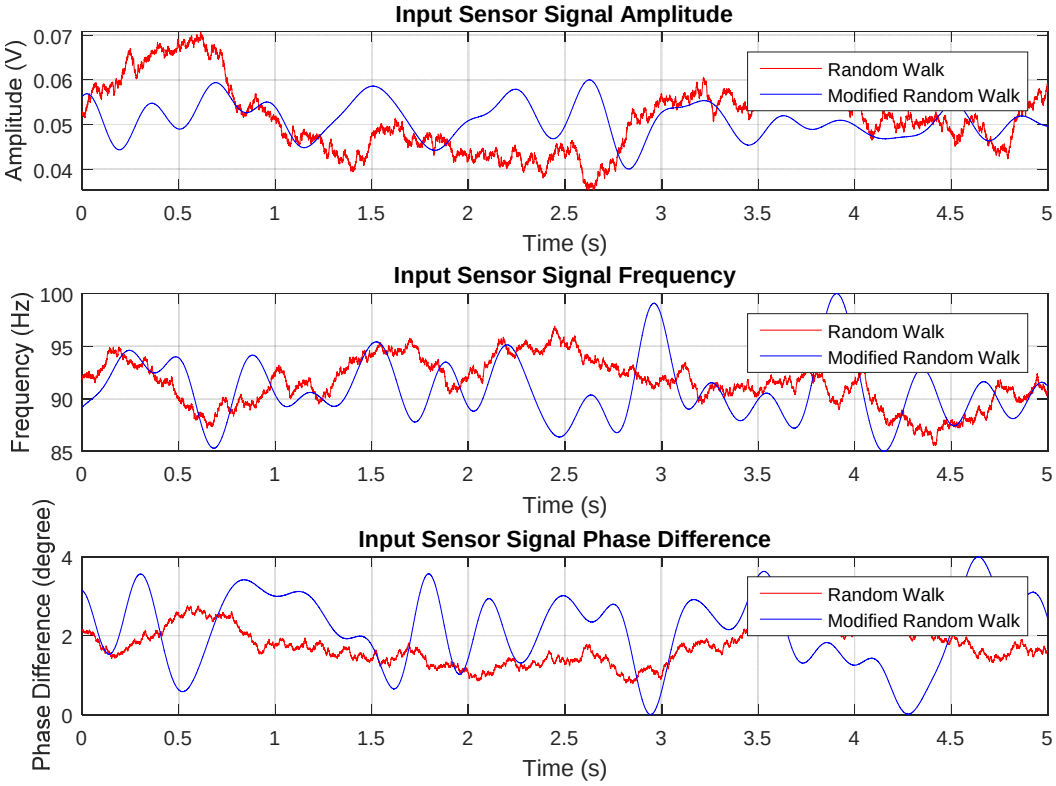

Figure 8. Comparison between RWM and MRWM techniques for generating parameter values

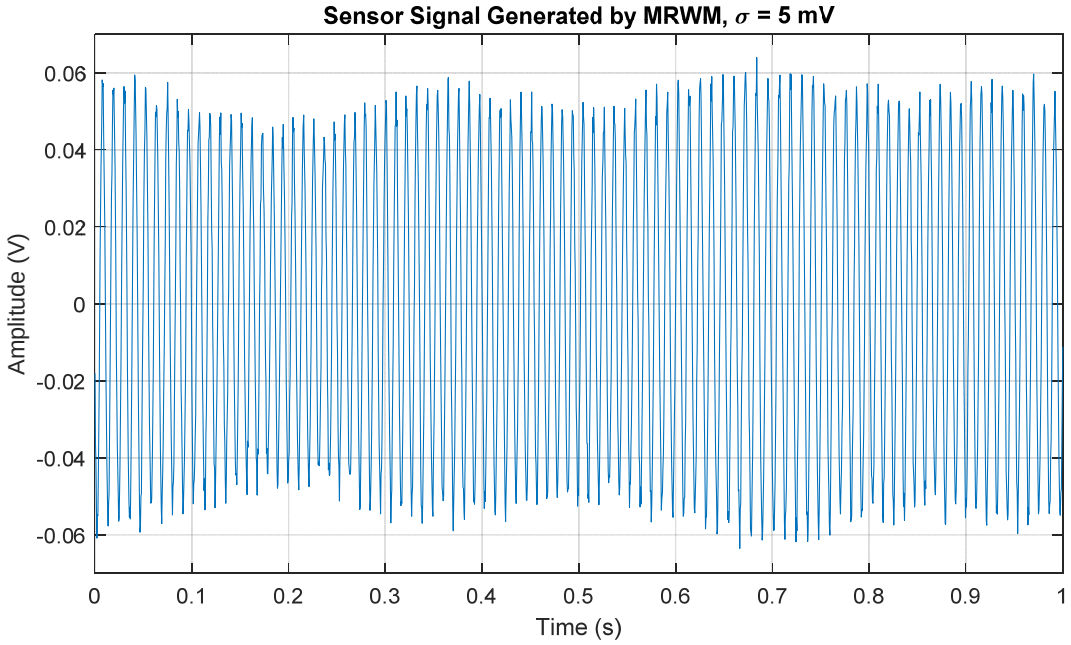

Figure 9. Sensor signals generated by MRWM

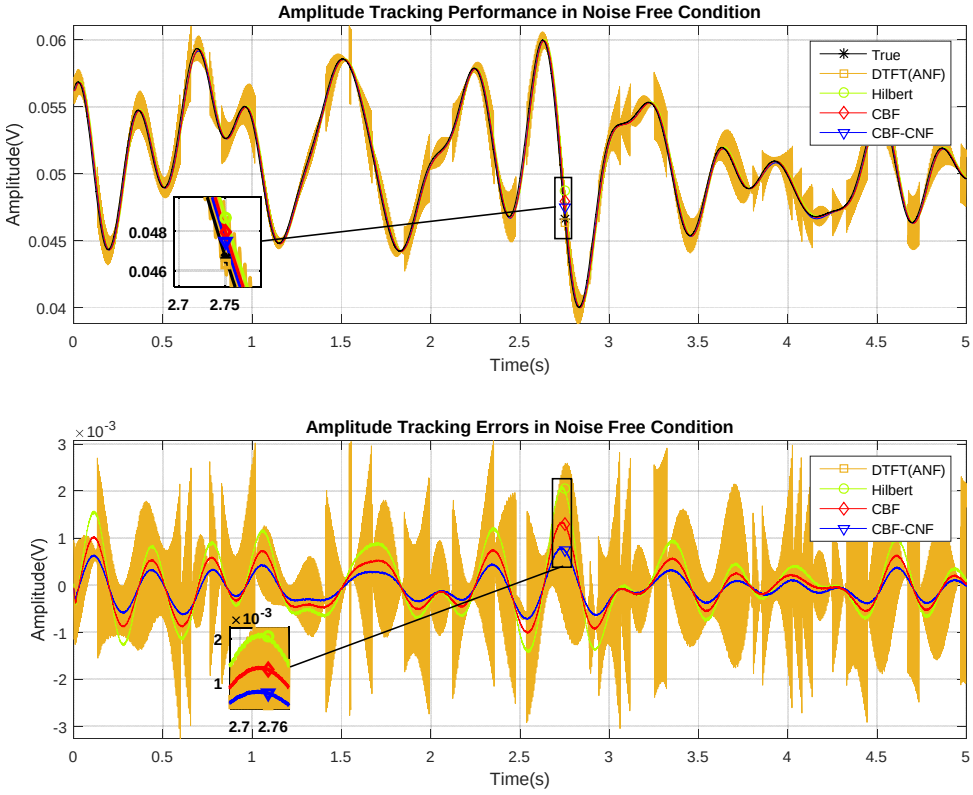

Figure 10. Amplitude tracking performance with noise free two-phase simulation

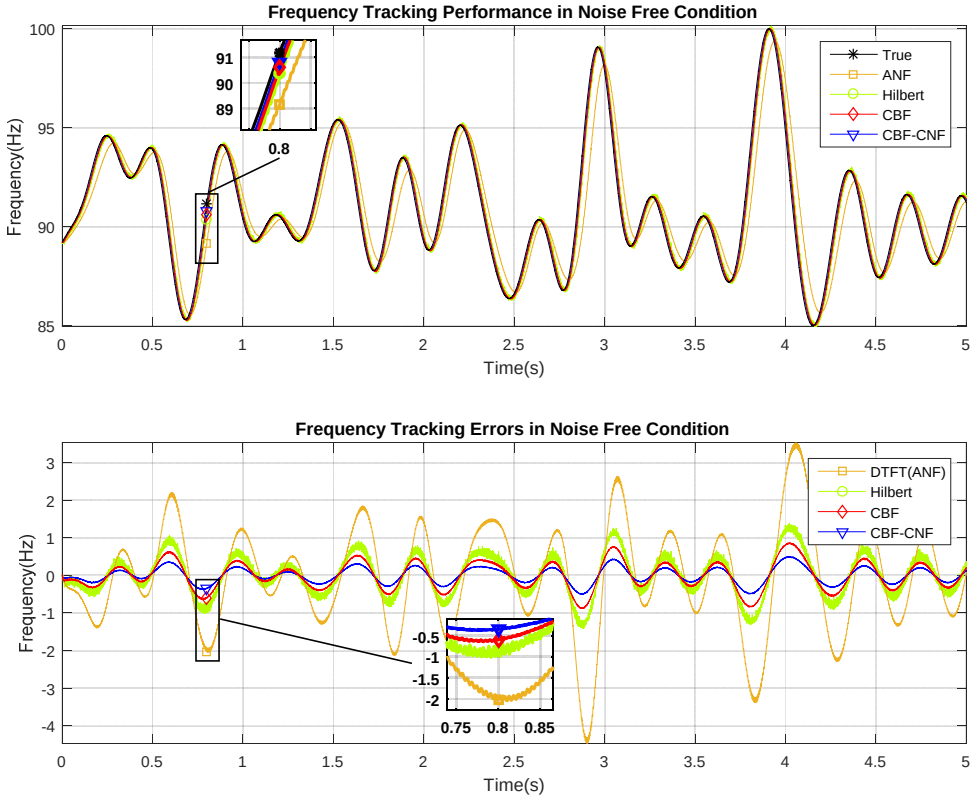

Figure 11. Frequency tracking performance with noise free two-phase simulation

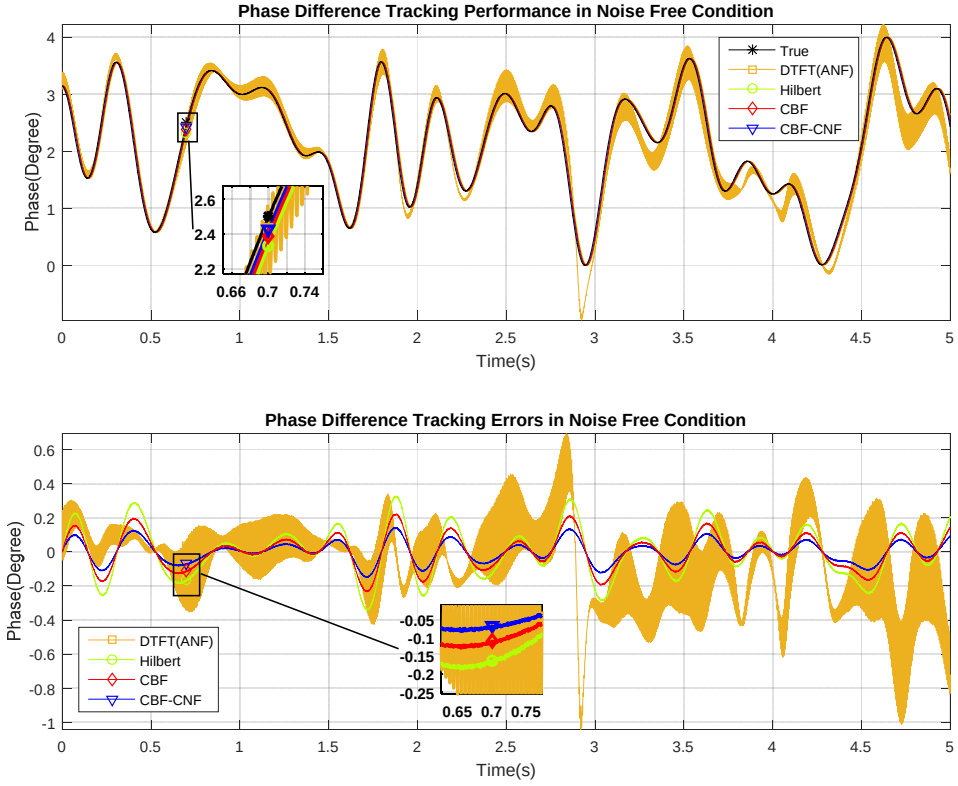

Figure 12. Phase difference tracking performance with noise free two-phase simulation

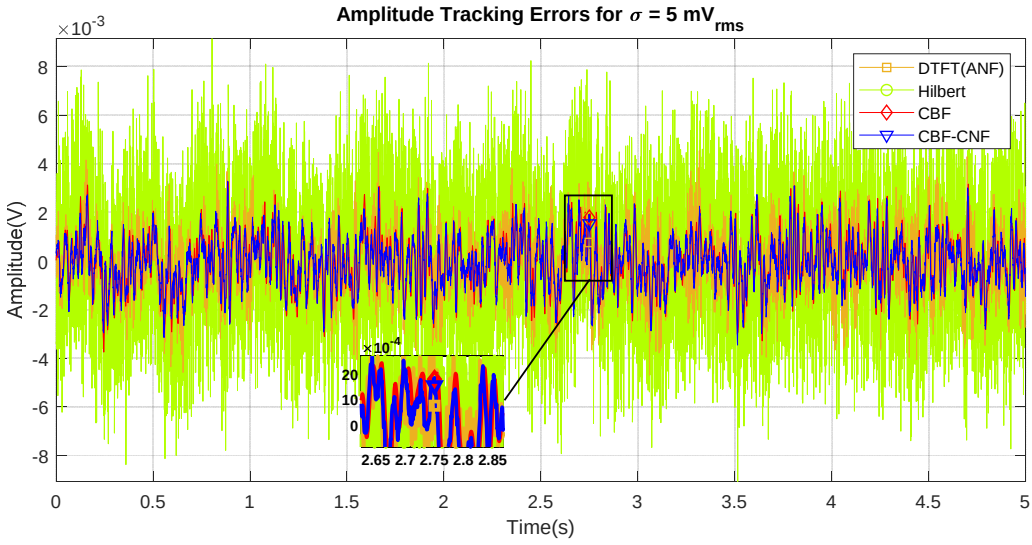

Figure 13. Amplitude tracking performance with σ = 5 mV

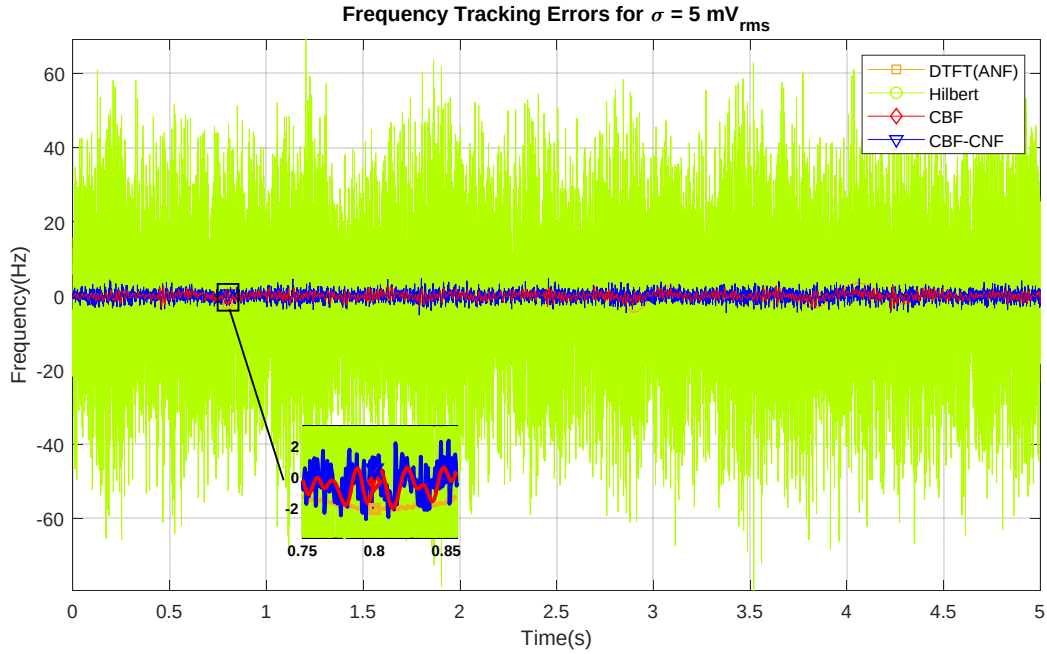

Figure 14. Frequency tracking performance with σ = 5 mV

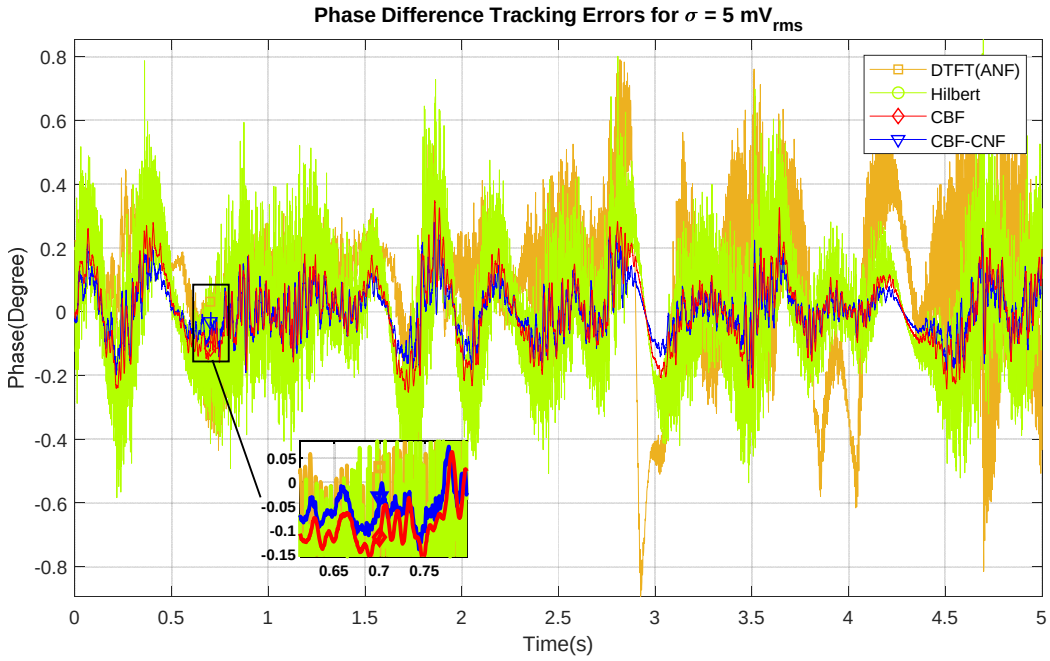

Figure 15. Phase difference tracking performance with σ = 5 mV

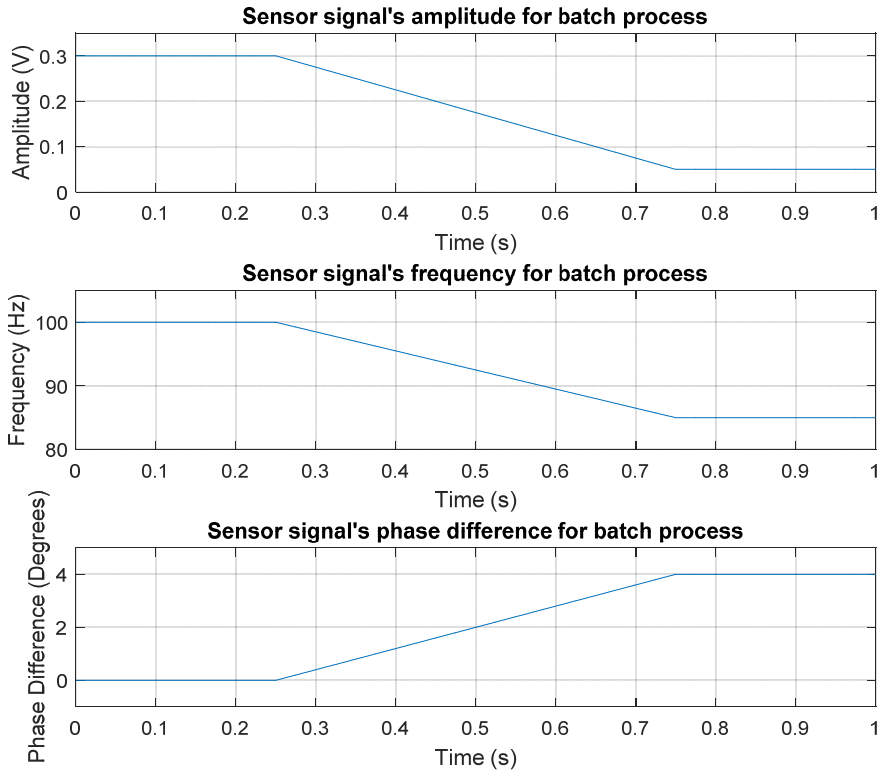

Figure 16. Sensor signal parameters for empty-full simulation

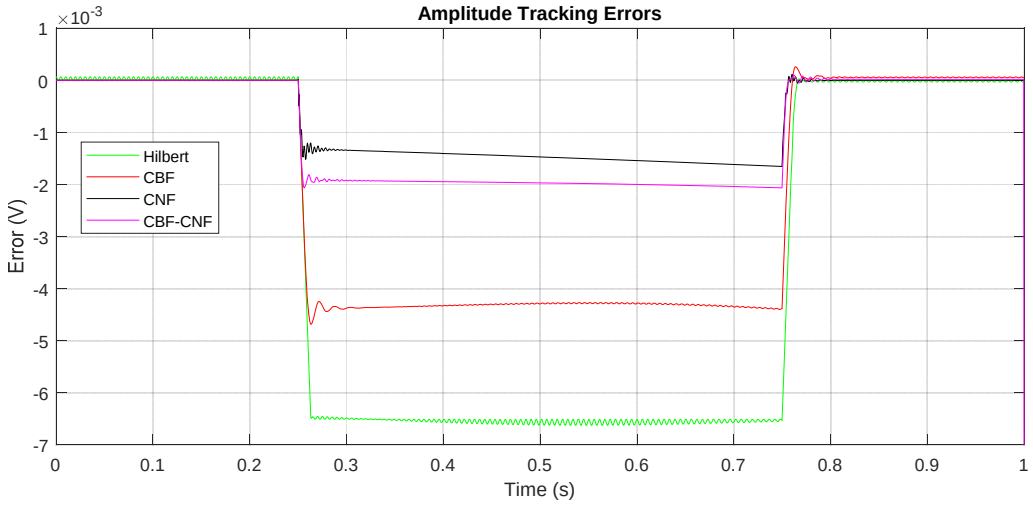

Figure 17. Amplitude tracking performance for empty-full simulation

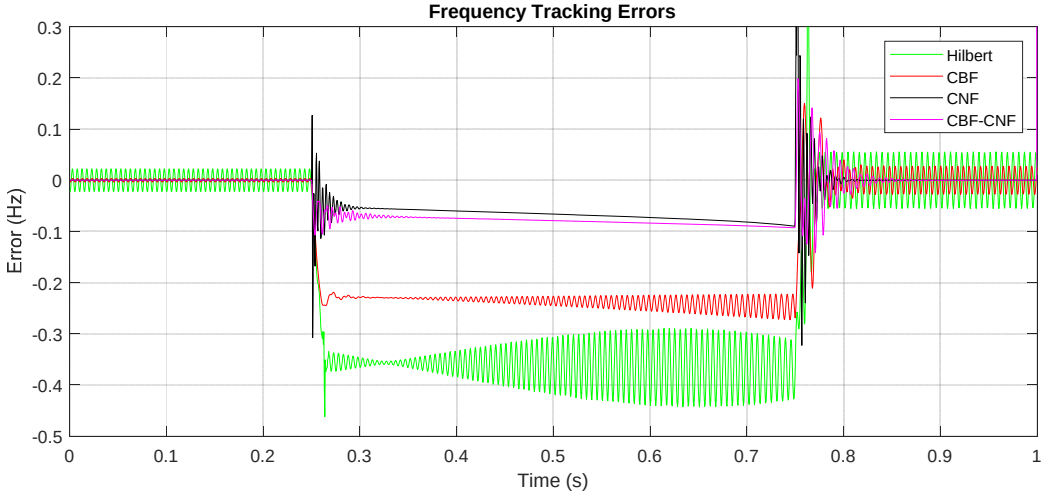

Figure 18. Frequency tracking performance for empty-full simulation

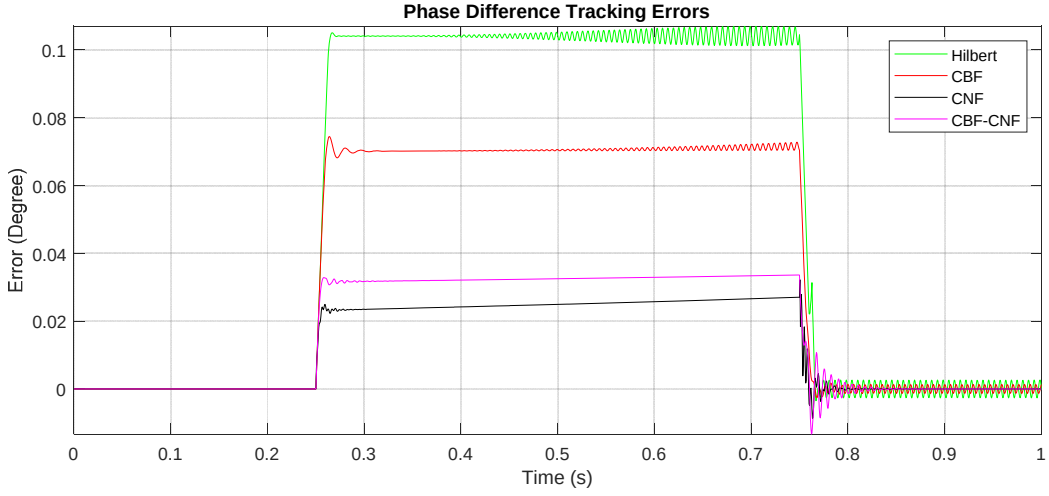

Figure 19. Phase Difference tracking performance for empty-full simulation