A COMPRESSIVE PHASE-LOCKED LOOP

Stephen R. Schnelle, J. P. Slavinsky, Petros T. Boufounos, Mark A. Davenport, Richard G. Baraniuk

1 Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005
2 Mitsubishi Electric Research Laboratories, Cambridge, MA 02139
3 Department of Statistics, Stanford University, Stanford, CA 94305

ABSTRACT
We develop a new method for tracking narrowband signals acquired via compressive sensing. The compressive sensing phase-locked loop (CS-PLL) enables one to track oscillating signals in very large bandwidths using sub-Nyquist sampling. A key feature of the approach is the fact that we perform the frequency tracking directly on the compressive measurements without ever recovering the signal. The CS-PLL has a wide variety of potential applications, including communications, phase tracking, and robust control.

Index Terms— Compressive sensing, phase-locked loop, FM demodulation

1. INTRODUCTION
Compressive sensing (CS) is a recently developed field within signal processing that enables the acquisition and recovery of sparse signals without loss of information at a sampling rate significantly below the Nyquist rate. CS uses a randomized measurement system, and typically recovers the signal via convex optimization or one of a fleet of greedy recovery algorithms. Several hardware architectures have applied this theory to wideband analog signals.

Unfortunately, sparse recovery algorithms are relatively computationally expensive. For streaming applications such as radio receivers, low computational complexity and real-time recovery is paramount. Furthermore, the finite-dimensional nature of existing recovery algorithms requires that streaming (infinite-length) signals must be processed in finite-length blocks, often introducing significant input-output delay and blocking artifacts at the boundaries. Thus, classical CS recovery algorithms are not appropriate for most real-time applications.

In this paper we develop a phase-locked loop (PLL) architecture that extracts phase and frequency information directly from compressive samples of modulated signals [1]. This has a variety of applications involving frequency and phase tracking, such as the demodulation of frequency modulated (FM) signals. Since the modulated signal is never fully recovered, the CS-PLL offers computational advantages over streaming CS recovery algorithms, such as [2], that would require the use of a conventional PLL on the recovered signal. Thanks to the implementation of other signal processing operations in the compressive domain, such as filtering and detection systems [3], the CS-PLL can be integrated smoothly into devices such as wideband compressive radio receivers, which can track and monitor a wide range of radio frequencies in real-time.

A special case of the CS-PLL includes PLLs that use random sampling, which have been explored in previous works. For instance, a simple FPGA implementation of a phase-locked loop for quantized additive random sampling (ARS) was supported with numerical simulations for varying loop filter parameters in a random sampling PLL for synchronization applications in [4]. This paper generalizes this approach to more generic CS sampling schemes.

In the next section we establish our notational lingo and provide the relevant background on PLLs, CS, and practical compressive samplers. Section 3 introduces the CS-PLL, while Section 4 provides an analysis of its key features. Section 5 provides experimental results of the system, while Section 6 concludes the paper.

2. BACKGROUND

2.1. Phase-locked loop (PLL)
The phase-locked loop (PLL) is a well-established method for tracking the frequency and phase of a signal \( x[n] \) using a feedback loop to continuously update an estimate of the parameters of the signal. Figure 1 shows a typical discrete-time real-valued PLL architecture. The phase detector and loop filter estimate the phase difference between \( x[n] \) and a reference signal \( u[n] \) by multiplying the two signals and low-pass-filtering the product to obtain

\[
\hat{\theta}[n] = \sum_k x[k] u[k] h[n - k],
\]

where \( h[n - k] \) is the impulse response of the low-pass filter. The phase estimate is used by an oscillator to produce the reference signal (with sinusoidal carrier)

\[
u[n] = \cos(\omega n + \hat{\theta}[n]).
\]

The PLL works by adjusting the estimate of \( \hat{\theta}[n] \) until \( x[n] \) and \( u[n] \) are approximately orthogonal.

A second-order loop enables the PLL to track both a signal’s phase and frequency, while higher-order systems can be useful for handling Doppler effects. A standard baseline model of a PLL has a loop filter described by the transfer function

\[
H(z) = C_2 \frac{z - \omega_n}{z - 1} + C_1 \frac{\omega_n}{z - 1},
\]

where \( C_1 = \omega_n^2 \), and \( C_2 = 2\zeta\omega_n \). For more information on PLLs, see [5].
We now introduce the CS-PLL, a new family of digital and mixed analog/digital PLLs based on CS. Recall that the phase estimate update calculation in the basic PLL is the (weighted) inner product between the Nyquist-rate samples $x[n]$ of the signal we wish to estimate/track and the estimated signal $u[n]$ generated by the oscillator. If both $x[n]$ and $u[n]$ can be represented by not only their Nyquist rate samples but also their (lower rate) compressive samples with $x[n]$ producing $y[m]$ and $u[n]$ producing $v[m]$, then the RIP guarantees that the inner product between their compressive samples $y[m]$ and $v[m]$ will be very close to the inner product between their Nyquist rate samples $x[n]$ and $y[n]$. Hence the angle is maintained between the vectors.

Leveraging this insight, we introduce two compressive samplers into the basic PLL system to create the CS-PLL shown in Figure 3. The first sampler acquires compressive samples $y[m]$ of the continuous-time input signal $x(t)$. (Note that the architecture could also easily accommodate a discrete-time input signal $x[n]$. For simplicity we focus in this paper only on the continuous-time case.) The second converts the oscillator’s Nyquist rate samples $u[n]$ into the compressive samples $v[m]$. The sampling operation in the loop—which is the discrete-time model for the compressive sampling used to acquire the signal—is causal and efficient to implement.

As with the classical PLL, the CS-PLL computes the phase estimate inner product

$$
\theta[m] = \sum_k y[k] \bar{v}[k] h[m-k],
$$

with index $m$ denoting the lower sampling rate and $h[n]$ mimicking the response of the higher rate filter in the Nyquist-rate PLL. A non-linear and/or time-varying filter could also be used here, for example if we had a signal with varying message bandwidth, just as in the traditional case.

### 4. ANALYSIS AND MODELLING OF THE CS-PLL

In this section, we analyze the properties of the CS-PLL and use a simple model to study its stability properties.

#### 4.1. Estimation

We first show that the CS-PLL is a maximum likelihood estimator of a modulated signal’s phase and frequency. To begin, suppose that the CS-PLL is frequency locked, and thus we are only trying to estimate the signal’s phase [5]. This is a safe assumption in typical PLL analysis: large frequency offsets prevent the loop from locking, whereas small offsets are handled with other structures such as second-order loops. Our signal of interest

$$
x(t) = \cos(\omega_c t + \theta)
$$
(with \( \omega_i \) the continuous time frequency), is compressively sampled and corrupted with additive white Gaussian noise (AWGN) \( w_i[m] \). Thus, we obtain
\[
y[m] = \sum_{k=-\infty}^{\infty} p_m[k] \cos(\omega k + \theta) + w_i[m]
\]
where \( p_m[k] \) denotes the pseudo-random coefficients for sample \( m \) (i.e., element \((m, k)\) of the sampling matrix \( \Phi \) corresponding to the compressive sampler) and \( \omega \) is the Nyquist-rate discrete-time frequency corresponding to \( \omega_i \). Although \( p_m[k] \) may be a pseudorandom sequence, the sequence is known to the system, and thus our sole source of randomness is the input noise \( w_i[m] \) over a set of \( M \) measurements. AWGN added to \( x(t) \) is simply scaled AWGN on \( y[m] \).

We now show that the CS-PLL can be viewed as a maximum likelihood (ML) estimator (and, assuming independent measurements, an MMSE estimator as well) by considering a signal with slowly varying phase over \( M \) measurements. The tracking error variance of the unbiased estimator \( \sum_{k=-\infty}^{\infty} p_m[k] \cos(\omega k + \tilde{\theta}) \) assuming each measurement is independent is
\[
\sigma^2 = \sum_{m=1}^{M} \left(y[m] - \sum_{k=-\infty}^{\infty} p_m[k] \cos(\omega k + \tilde{\theta})\right)^2
\]
where \( \tilde{\theta} \) represents the quantity we are estimating. Taking the derivative of \( \sigma^2 \) with respect to \( \tilde{\theta} \)
\[
\frac{\partial \sigma^2}{\partial \tilde{\theta}} = \sum_{m=1}^{M} \left(y[m] - \sum_{k=-\infty}^{\infty} p_m[k] \cos(\omega k + \tilde{\theta})\right) \times \sum_{k=-\infty}^{\infty} p_m[k] \sin(\omega k + \tilde{\theta})
\]
Two terms are present in each measurement \( m \): the correlation of \( y[m] \) with the compressive measurements of a 90-degree out-of-phase reference output and a offset term independent of the output.

The correlation term confirms that the compressive sampler model in the loop should match that used on the input. For the offset term, we note that if the \( p_m[k] \) terms are generated independently of each other with mean zero, then when we treat \( p_m[k] \) as random and consider the average case of \( M \) samples by finding the expected value over \( p_m[k] \). The second terms in the summation reduce to
\[
\sum_{k=-\infty}^{\infty} (p_m[k])^2 \cos(2(\omega k + \tilde{\theta}))
\]
and would be filtered out by the low-pass filter in the CS-PLL. Ignoring this offset simplifies our system’s complexity and corresponding analysis model and does not drastically affect performance. The CS-PLL naturally acts to achieve average case performance over time with its loop filters and adaptive nature.

**4.2. Linear model for the CS-PLL**

Next we develop a simplified linear model for the PLL and determine its transfer function and other characteristics. Given an RD with chipping sequence \( p_m[\frac{N}{M} m + k] \) and Nyquist rate samples \( x[k] = \sin(\omega k + \theta_i[k]) \) and \( u[l] = \cos(\omega l + \theta_j[l]) \) with \( \omega \) the discrete-time Nyquist frequency, the multiplier output

\[
\theta[n] = \sum_{k=0}^{L-1} X[k] u[r_k] + \theta_m[n] + \theta_i[n] + \theta_j[n]
\]

![Fig. 4. Linear sample-and-hold model for CS-PLL stability analysis using RDs.](image)

where \( X[k] \) consists of two terms \( \sum_{k=0}^{L-1} \sum_{l=0}^{M-1} p_m[r_k] x[r_k] p_m[r_l] u[r_l] \) and \( \sum_{k=0}^{L-1} \sum_{l=0}^{M-1} p_m[r_k] x[r_k] p_m[r_l] u[r_l] \) where \( L \) is the length of the RD band, and \( r_k = \frac{N}{M} m + k \). The first sum accumulates Nyquist-rate samples, and the second accumulates cross-term noise \( w_i[m] \).

Since \( w_i[m] \) is zero-mean and uncorrelated with the input and the feedback term, we can formulate a linear model for the phase as well. As before, our model includes an FIR filter with \( L \) random taps equal to \( (p_m[\frac{N}{M} m + k])^2 \) and downsampling by \( \frac{N}{M} \) following the phase detector. For a sampler that contains only a single RD, \( L = N/M \). This FIR filter model has sharp nulls at multiples of the aliasing frequency, indicating that we continue to remove narrow-band noise that CS is designed to prevent. The loop filter operates at the lower sampling rate and a sample and hold element is added to this input to the oscillator to return the sampling rate to the original high rate. The linearized model is shown in Fig. 4(b).

For the error signal, although solving this high-order polynomial analytically for bounds is difficult, we find heuristically that reducing \( C_1 \) and \( C_2 \) by the compression ratio \( N/M \) makes the system stable in the case of a single random demodulator. Assuming the loop filter is designed properly to ensure stability, a zero steady state phase error for an initial frequency offset will result, found by solving \( e_{ss} = \lim_{t \to \infty} (z - 1) (1 - H(z)) \). Because of the accumulate-and-dump component in this model, the loop order is \( L + \frac{N}{M} \), verifying the potential for increasing instability as the compression grows. If we were to interleave multiple random demodulators (with \( L \) a multiple of \( N/M \)), the stability of the system becomes a much greater concern.
mic compression factor of the performance of the conventional PLL corresponds to a logarithmic compression ratio of 20. The output of the CS-PLL closely resembles power cordless phone input signal of 25 dB input SNR with a compression factor of 0, shown at the left edge of the plots.

We quantify the CS-PLL’s performance in terms of its output SNR and compare with a more conventional Nyquist-rate PLL for various and compression factors averaged over 25 trials. Note that the performance of the conventional PLL corresponds to a logarithmic compression factor of $N/M$ of 0, shown at the left edge of the plots. We do not use an MMSE error criterion due to the inherent delay in discrete-time PLLs. Instead, the output SNR is measured by dividing the signal power by the noise power in the frequency spectrum, computed over an average 250 Hz band around the signal of interest (the noise is relatively flat over the spectral region of interest, and traditionally out-of-band noise is filtered at the output of a PLL).

From Figure 5, we see that at high input SNR, there is an initial sharp drop in the SNR when compression is used due to the introduced cross-term noise, but then it smooths to a more gradual 3 dB per factor of 2 compression due to noise-folding. Wideband noise folding continues to be an inevitable consequence of using CS, just as it appears in traditional CS recovery algorithms [12].

The performance of the CS-PLL improves slightly when an additional offset factor is added to the phase detector based on the current phase estimate derived with our maximum likelihood estimator in (9), though in many practical scenarios it can be ignored.

Compressions rates in the wideband receiver are not limited to powers of 2. In Figure 6, we apply the CS-PLL to a simulated high-power cordless phone input signal of 25 dB input SNR with a compression ratio of 20. The output of the CS-PLL closely resembles the result of FM demodulation using a Hilbert transform and phase unwrapping.

6. CONCLUSIONS

This paper has developed a framework for constructing phase-locked loops that perform phase and frequency tracking directly on compressive measurements. The ideas could be extended relatively easily to any variants of the PLL that use a multiplier phase detector. A variety of applications could utilize the CS-PLL, ranging from communication schemes using phase or amplitude modulation, GPS, and industrial control applications. Future work includes a more thorough probabilistic non-linear analysis of the transient characteristics of the CS-PLL, such as lock time, cycle slipping, and instantaneous phase error.

5. SIMULATIONS

To simulate the performance of the CS-PLL, we use a sampling rate of 2.048MHz, and a oscillator frequency of 120kHz, and simple second-order filter with loop bandwidth of 10kHz (adjusted in the discrete domain for the appropriate compressive sampling rate). Our first input signal is a compressively sampled FM-modulated 2.5kHz signal with frequency deviation of 1.6kHz. Data is sampled using a random demodulator with $\pm 1$ taps.

Fig. 5. Output SNR for a random demodulator when using Bernoulli coefficients, normalized Gaussian coefficients, and when including the offset in the phase detector, each for varying input SNRs.

Fig. 6. CS-PLL output compared to traditional FM demodulation of the Nyquist-rate samples.

7. REFERENCES