Code Acquisition at Low SINR in Spread Spectrum Communications

Manish Amde*, Joel Marciano†, Rene Cruz* and Kenneth Yun*
*Department of Electrical and Computer Engineering
University of California San Diego
†Department of Electrical and Electronics Engineering
University of Philippines

Abstract— **We consider the problem of code acquisition at low signal to interference and noise ratio (SINR) in packet-based spread spectrum communications, where a centralized "pilot" synchronization signal and high-precision oscillators are not available. Such problems arise in the context of ad-hoc networks, for example, which require fast code acquisition times and gradual degradation of the network with decreasing SINR.**

We motivate the use of the "postdetection integration" approach, which utilizes the energy of multiple bits in a packet, for code acquisition at low SINR. We present the implementation of a fully parallel architecture which simultaneously looks at all possible code alignments over multiple bits. This leads to a drastic reduction in acquisition time compared to serial search based methods. We report some preliminary simulation and experimental results from a hardware prototype of a transceiver on which the code acquisition algorithm was implemented.

I. Introduction

Spread spectrum communication techniques are popular for several reasons including resistance to narrowband interference and jamming, multipath rejection and suitability for multiple access [1], [2], [3].

Currently deployed spread spectrum based communication networks, for example 802.11b [4], support high data rates at high signal to interference and noise ratio (SINR) for asynchronous packet-based communication. They otherwise require a separate pilot channel for synchronizing their communication to a pilot signal. In one-to-many downlink transmissions in cellular networks, for example IS-95 [2], the mobiles synchronize to the pilot signal sent by the base station. The synchronous communication using a pilot channel leads to the loss of power even in the absence of useful communication.

The operation at high SINR leads to a corresponding spatial profile which is narrow and steep, as a longer range leads to lower SINR at the receiver if the transmitted power is constant. The spatial range of networks can be extended using a higher processing gain in spread spectrum communication at the cost of lowering the data rate. This will facilitate a gradual degradation of network rather than a complete breakdown at low SINR.

Signal reception at low SINR poses many challenges. For accurate demodulation, the receiver needs to synchronize to the carrier frequency and phase of the transmitted signal. Symbol detection at the receiver requires synchronization to the clock used at the transmitter for generating the spreading code and data symbols. It further requires discovering the start of the spreading sequence in the received signal for correct de-spreading operation at the receiver. These synchronization operations take place in two steps [1], [2], [3], [5] :

- *Acquistion* : This step searches for the start of the pseudo noise (PN) code in the received signal for precise de-spreading operation.
- *Tracking* : This step performs fine synchronization of the receiver clock to the clock used for generating the data symbols and spreading code at the transmitter. For coherent demodulation, it also performs *carrier recovery* functionality at the receiver.

Tracking is done after *code acquisition* as the SINR per chip is typically too low to accurately determine chip transitions without leveraging the spreading gain of the code acquisition method. At low SINR, error correction techniques could be used for reliable estimation of transmitted data symbols even if a significant percentage of raw data symbols are erroneous.

Code acquisition requires testing all the possible hypotheses for correct alignment of the code phase in the received signal with a PN correlator used in the de-spreading operation for data symbol detection. This hypothesis testing was traditionally carried out in a serial manner [1], [6], [7] as the hardware complexity, cost and size prohibited use of parallel correlators for reducing the acquisition time. It is interesting to note that the tradeoff was mentioned a couple of decades back in [1] where the authors hinted at more parallel implementations in the future with improving VLSI technology.

Confident about being on the right side of the Moore's law, we present a fully parallel VLSI implementation for *code acquisition* at low SINR. The implementation is based on the "postdetection intergration"' method described in [3] and looks at the energy of multiple bits in a packet for correct alignment of the spreading code at the receiver.

The remainder of the paper is organized as follows. Sections II and III describe the physical layer model and the code acquisition algorithm, respectively. Section IV states the modifications made to the algorithm to make it suitable for implementation on available hardware. Section V demonstrates the performance of the algorithm on stored data samples captured over the wireless medium. Section VI describes the implementation of the algorithm on a hardware prototype

0-7803-9780-0/06/$20.00 ©2006 IEEE

along with experimental results. The conclusions are presented in Section VII.

II. Physical Layer Model

A. Model for Transmitted Signal

We assume that direct sequence spread spectrum modulation, with Binary Phase Shift Keying (BPSK) is used. The signal transmitted for a packet at baseband is $x(t)$, where

$$x(t) = d(t)c(t) \ ,$$

$d(t)$ is the data signal and $c(t)$ is the spreading signal. Both signals take on values in $\{-1, +1\}$.

The data packet has R bits, $b_1, b_2, \ldots, b_R$. The time duration of a bit transmission is T seconds. The data signal is defined as

$$d(t) = \begin{cases} b_j & \text{, if } (j-1)T \le t < jT \text{ where } 1 \le j \le R \\ 0 & \text{, otherwise} \end{cases}$$

where $b_j \in \{-1, +1\}$

The processing gain is m and the chip duration is T_c seconds, where $T_c = T/m$. In other words, m chips are sent per transmitted bit. The spreading sequence is a random sequence of $m * R$ chips and it is known at the receiver. The i^{th} chip is $c_i \in \{-1, +1\}\ \forall\ i$. The spreading signal is defined as $c(t) = c_i$, if $(i-1)T_c \le t < iT_c$.

The baseband signal $x(t)$ is passed through a linear pulse shaping filter $H(f)$ with impulse response $h(t)$ to reduce interchip interference (ICI) and limit the signal to its allocated spectrum. It is finally up-converted to a RF frequency ω_c as shown in Figure 1.

The transmitted signal is

$$s(t) = A'(x(t) * h(t)) \cos(\omega_c t) \ ,$$

where A' is an amplitude parameter that determines the transmission power.

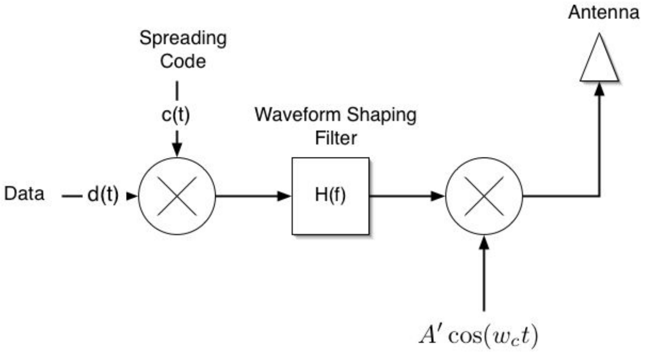


Fig. 1. Spread Spectrum Transmitter

B. Model for Received Signal

The received signal $r(t)$ is represented by

$$r(t) = as(t - t_0) + v(t) + n(t)$$

where a is the signal attenuation factor, $v(t)$ is an interference signal caused by other packet transmissions, $n(t)$ is additive white Gaussian noise and t_0 is a time offset denoting the start of packet transmission. For ease of exposition, we assume t_0 equal to zero.

The received signal can be written as

$$r(t) = A(x(t) * h(t)) \cos(\omega_c t) + V(t) \ ,$$

where $A = A' * a$ and $V(t) = v(t) + n(t)$ is assumed to be a zero mean Gaussian random process.

C. Down-conversion and Matched Filtering

The receiver RF signal is down-converted into two branches I and Q using oscillators at frequency offset $\Delta\omega_c$ and phase offset ϕ with respect to the carrier of the received signal as shown in Figure 2. The oscillators for the I and Q branches operate at the same frequency and are 90^o out of phase (they are typically generated using a single oscillator for the I branch and an appropriate delay element before the Q branch). It is then passed through a matched filter $H^*(f)$ which is matched to the pulse shaping filter used at the transmitter. The corresponding signal received at baseband can be written as

$$r_{bb}^{(I)}(t) = Ax(t) \cos(\Delta w_c t + \phi) + V^{(I)}(t) \ ,$$

$$r_{bb}^{(Q)}(t) = Ax(t) \sin(\Delta w_c t + \phi) + V^{(Q)}(t) \ ,$$

where $V^{(I)}(t)$ and $V^{(Q)}(t)$ are assumed to be zero mean Gaussian random processes with variance $V_o/2$.

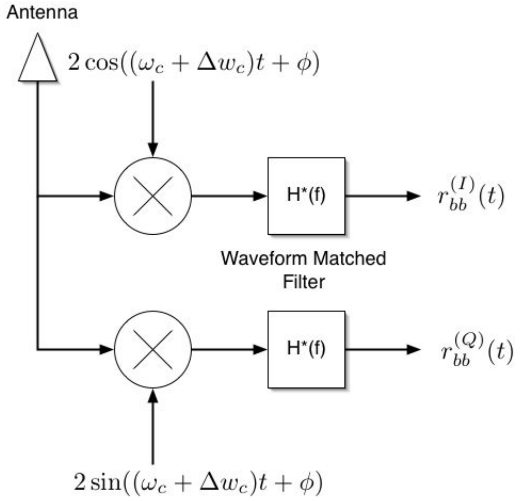

Antenna
$2\cos((\omega_c + \Delta w_c)t + \phi)$
H*(f)
$r_{bb}^{(I)}(t)$
Waveform Matched Filter
H*(f)
$r_{bb}^{(Q)}(t)$
$2\sin((\omega_c + \Delta w_c)t + \phi)$

Fig. 2. Down-conversion and Matched Filtering

III. Code Acquisition Algorithm

A. Energy Detection

The detection of a signal with an unknown phase can be accomplished using a non-coherent detection mechanism as shown in Figure 3 [3]. Both baseband signals $r_{bb}^{(I)}(t)$ and $r_{bb}^{(Q)}(t)$ are passed through a de-spreading correlating structure and then combined (addition of squares) to detect the energy of the signal. This detection structure is referred to as a *quadratic detector* [8].

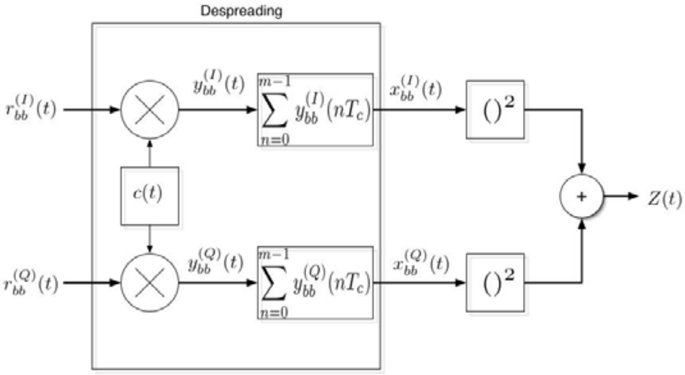


Fig. 3. Energy Detector

Under perfect code alignment, the outputs of the despreader are

$$x_{bb}^{I}(t) \approx Am\cos(\Delta\omega_c t + \phi)R(\tau) + I(t)$$

$$x_{bb}^{Q}(t) \approx Am\sin(\Delta\omega_c t + \phi)R(\tau) + I(t)$$

where $I(t)$ is zero mean Gaussian process with variance $V/2$ and $V = mV_0$. $R(\tau)$ is the auto-correlation of the PN sequence and accounts for the chip timing error even at correct alignment of code phase and m is the number of chips per bit. The chip timing error is rectified in the *tracking* phase which typically follows *code acquistion* [1], [2], [3].

The average value of the quadrature detector output at perfect code alignment is given by

$$E[Z(t)] = E[(x_{bb}^{(I)}(t))^2 + (x_{bb}^{(Q)}(t))^2] = M_D^2 = m^2A^2R^2(\tau)$$

This detector is analyzed in [3] for probability of detection (P_D) given a probability of false alarm (P_{FA}) and SINR μ defined as M_D^2/V.

B. Limitations on Long PN Correlation

The SINR μ from the last subsection is directly proportional to the spreading gain m. The underlying assumption for correct de-spreading is that the frequency error $\Delta\omega_c$ satisfies the condition $m\Delta\omega_c T \ll \pi$ where T is the period for each bit. This condition ensures that the remnant carriers after down-conversion $\cos(\Delta\omega_c t + \phi)$ and $\sin(\Delta\omega_c t + \phi)$ have the same sign over the entire de-spreading operation with a high probability.

Hardware complexity and cost prevent the use of a unique PN sequence across the packet. Therefore, many spread spectrum systems use the same PN sequence to encode every data bit.

C. PostDetection Integration

Postdetection integration (PDI) [3] has been suggested as a method for improving probability of detection P_D for a given probability of false alarm P_{FA} and SINR μ by using multiple observations. A new variable $D_{quad}(t)$ can be used as a test statistic for this quadratic detector where

$$D_{quad}(t) = \sum_{i=0}^{L-1} Z(iT)$$

L is the number of bits summed over (equivalent to number of observations). T is the time period for a data symbol and is equal to $m * T_c$ where m is the spreading ratio and T_c is the chip period. This operation in essence looks at the energy of multiple bits in a packet.

According to [3], the probability of false alarm is given by

$$P_{FA} = e^{-\theta/V} \sum_{k=0}^{L-1} \frac{(\theta/V)^k}{k!}$$

and the probability of detection is given by

$$P_D = \int_{\theta/V}^{\infty} \left(\frac{x}{L\mu}\right)^{(L-1)/2} e^{-(x+L\mu)} \mathcal{I}_{L-1}(2\sqrt{L\mu x})dx$$

where θ is the threshold for packet detection and $\mathcal{I}_{L-1}()$ is the $(L-1)^{th}$ order modified Bessel function.

The integral for P_D is the L^{th}-order Marcum Q-function [9], [8]. This integral is analyzed for large L in [8] using Gram-Charlier series. It is shown that for fixed probability of detection P_D and probability of false alarm P_{FA}, the minimum SINR required is proportional to $\sqrt{L}$. For example, if the minimum SINR required for a particular P_D and P_{FA} under a single observation ($L = 1$) is α, then to have the same P_D and P_{FA} for the SINR level at $\alpha/10$, the number of observations need to be repeated hundred times ($L = 100$).

Since the performance of the PDI detector improves with the length of the summation L, one could theoretically improve performance (albeit at the risk of increasing hardware cost) by summing over the total number of bits in the packet ($L = R$). This assumes perfect chip timing recovery at the receiver which is typically conducted after the code acquisition phase. Therefore, the optimal number of bits to be summed over for detection is dependent on the relative drift between the clocks used for chip generation at the transmitter and receiver. Once the clocks drift by more than a chip period, the code phase changes and subsequent addition of bits does not improve performance ($R(\tau) = 0$ for $|\tau| > T_c$)).

IV. Practical Considerations

The *code acquisition* algorithm has been described in continuous time in Sections II and III but the baseband section of the algorithm is implemented in digital reconfigurable logic using a field programmable gate array (FPGA). This requires a digital-to-analog and analog-to-digital conversion for interfacing the FPGA to RF transceivers which transmit and receive signals over the wireless medium.

The digital implementation of the algorithm described in Section VI computes the algorithm output every $T_s = T_c/n$ seconds where T_c is the period per chip, T_s is the analog to digital converter (ADC) sampling period and n is the number of samples per chip. This leads to a completely parallel implementation of the algorithm and a low acquisition time as a parallel search is performed over all possible code phases in the packet.

Spread spectrum radios perform code acquisition before carrier recovery for low SINR operation. Ideally, we would like to perform the PLL operation on the FPGA after the code acquisition stage. We used an off-the-shelf RF receiver for down-conversion which has phase-locked loops (PLLs) for carrier recovery. This leads to code acquisition (implemented on the FPGA) being performed after the carrier recovery phase. Hence, we could not perform very low SINR operation as the PLL's fail at low signal strength. Since we could do coherent demodulation at moderate SINR, we just implemented the I branch of the energy detector described in Section III-A.

To reduce the complexity of implementation for the code acquisition algorithm, we used the following test statistic

$$D_{abs}(t) = \sum_{i=0}^{L-1} |x_{bb}^{(I)}(iT)|$$

This detector sums the absolute values of the de-spreader output for detecting energy using multiple bits. The next two sections discuss the performance of the code acquisition algorithm using the detector described above and the implementation of a completely parallel architecture which simultaneously looks at all the possible code phase alignments over multiple bits in a packet.

We believe that we can implement a similar architecture for the code acquisition algorithm described in Section III if we are given a suitable RF transceiver. Our future research involves the design of a custom RF receiver which allows us to closely replicate the optimal code acquisition algorithm for low SINR operation in hardware.

V. Results of Offline Processing in a Controlled Environment

We evaluated the performance of our algorithm through offline processing of measured data obtained from wired transmission through attenuators. The attenuators were set such that the received signal level is $-110\,\text{dBm}$, measured at the receiver antenna output. We implemented a packet transmitter and a digital IF samples receiver on a Virtex-4 FX FPGA [10] on the Memec Virtex-4 FX LC development kit. This FPGA is interfaced to the computer using Gigabit Ethernet for collection of large blocks of data at extremely fast rates for offline processing. This makes our data acquisition prototype low-cost and mobile, and eliminates the requirement for expensive and bulky deep memory logic analyzers.

An evaluation board for the AD9862 chip, a Mixed Signal Front-End (MxFE) Processor from Analog Devices, is used for converting a digital signal from the FPGA board to an analog intermediate frequency (IF) signal as well as for converting an analog IF signal to digital. On the transmit (Tx) path, the MxFE has two 14-bit 128 MSPS D/A converters and digital mixers for frequency up-conversion. On the receive (Rx) path, the MxFE has two 12-bit A/D converters that could sample up to 64 MSPS. The output of the DAC and the input of the ADC are centered at 44 MHz.

The MxFE board is interfaced to a transceiver evaluation board from RFMagic, which works at the RF frequency of 2.4 GHz (ISM band). On the RF side, this board is interfaced to commercially available antennas or an attenuator for controlled tests.

The received samples from the ADC are sent to the computer where they are stored for analysis. The proposed algorithm is implemented in software and its performance is analyzed using the stored samples. This setup provides us with a powerful tool to capture packets over the air or in a controlled environment using attenuators and improve our algorithm based on real wireless data samples.

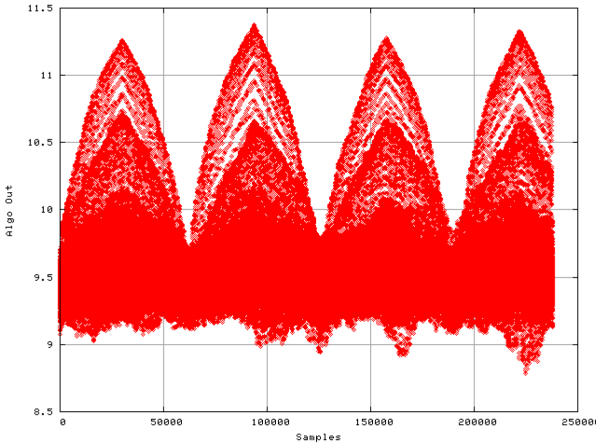


Fig. 4. Algorithm output every sampling cycle with alternating transmission and received input signal strength $-110\,\text{dBm}$

Figures 4 and 5 show $D_{abs}(t)$ computed off-line every sampling period for data obtained from actual packet transmissions received by our RF front end hardware and data collection system. In this implementation, $n = 16$ samples/chip, $m = 50$ chips/bit and the packet length R along with the observation interval L is equal to 40 bits.

For Figure 4, the received signal consists of on-off packet transmission at regular, equally-spaced intervals. The triangular envelope depicts the algorithm output when the the packet slides in and out of the PDI detector. The peak of each envelopes coincides with the correct time for code acquisition and is obtained by looking at the energy of all the bits in the packet.

Figure 5 shows results for a continuous packet transmission with no interval between consecutive transmissions,

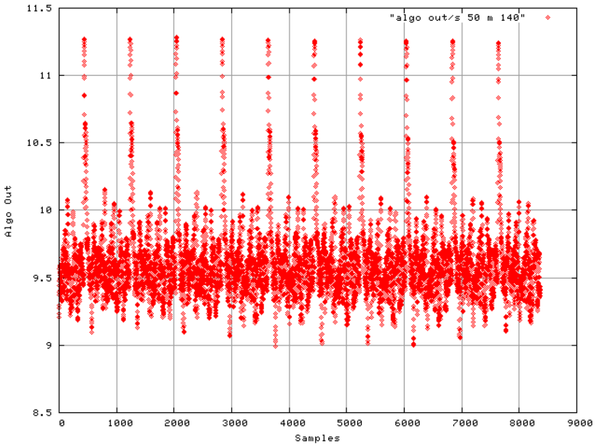


Fig. 5. Algorithm output every sampling cycle with continuous transmission and received input signal strength $-110\,\mathrm{dBm}$

and provides close-up detail of the peaks within the packet transmission time interval. As expected, the peaks recur after every $m*n = 800$ sampling periods, i.e. one bit period. These results show that the algorithm is able to effectively process low-level signals obtained from our prototype data collection system.

VI. Prototype Implementation

A. Hardware Prototype

We have implemented the detection part of the algorithm on a digital prototyping board which hosts a Xilinx Virtex-II Pro (XC2VP20) FPGA on it. This board has been developed by our research group and provides all the necessary functionalities for a complete wireless MAC and PHY layer implementation. A brief summary of the FPGA implementation is shown in Table 1. For understanding these parameters one can refer to the Virtex-II Pro documentation [11]. We use the same AD9862 and RFMagic evaluation boards mentioned in Section V to complete our transceiver. The area utilized on the FPGA is proportional to the spreading ratio (m) and number of bits of the packet observed by the algorithm (L). The hardware setup in the lab is shown in Figure 6.

Logic Distibution	Used	Avail.	Utilized
Number of 4-Input LUTs	4054	18560	21%
Number of BRAMs	85	88	96%
Number of bonded IOBs	28	556	5%

TABLE I
FPGA Implementation details.

B. Experiment settings

We have used the following parameters for implementing our algorithm. We chose a data rate of 40 Kbps and a packet length of 5 bytes $(R = 40)$. The algorithm looks at the energy of the entire packet $(L = 40)$. The chipping rate is 2 MHz (m = 50) and the sampling rate is 64 MHz $(n = 32)$.

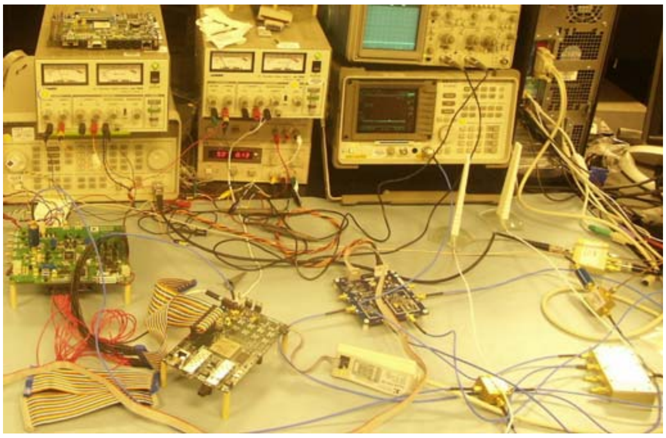
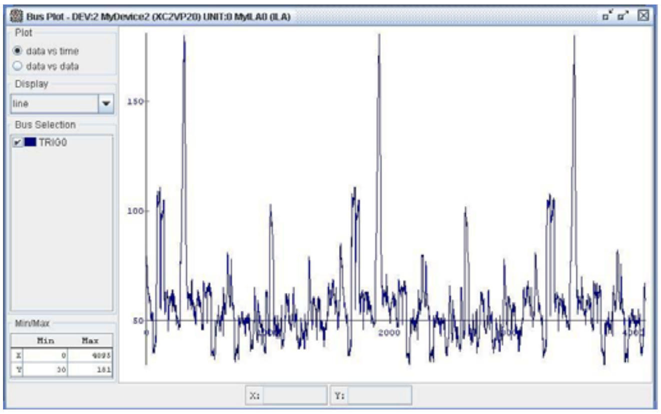


Fig. 6. Lab Setup

The transmitter continuously transmits packets with a zero time interval between packets. In order to prove that our algorithm works at lower signal strength, we conducted our test over a wireless channel with an input signal of $-110\,\mathrm{dBm}$ measured at the antenna output of the receiver.

C. Results

We examined the output of the algorithm using the Chipscope Pro software from Xilinx, which provides an on-chip debug and real-time logic analysis environment. We looked at the output of the algorithm, proportional to $D_{abs}(t)$, at random intervals of time for a duration of 4096 sampling cycles. The plot of the output is shown in Figure 7. The highest peak is at 181 units. The second highest peak which could be attributed to a multi-path component is around 110 units and also repeats every 1600 sampling cycles. We observed peaks repeating every 1600 sampling cycles $(n \times m = 32 \times 50 = 1600)$ which, as expected, is equal to one bit period.



Bus Plot - DEV:2 MyDevice2 (XC2VP20) UNIT:0 MyILA0 (ILA)

Plot
● data vs time
○ data vs data

Display: line

Bus Selection
☑ TRIG0

Min/Max

	Min	Max
X	0	4095
Y	30	181

X: Y:

Fig. 7. Algorithm output observed every sampling cycle using Chipscope Pro at $-110\,\mathrm{dBm}$

VII. Conclusions

A code acquisition algorithm for operation at low SINR has been presented. Offline analysis of captured packet transmissions proves that the algorithm works at low input signal levels. Finally, the algorithm is implemented on a prototype of a wireless node to show the feasibility of implementation in hardware and ability of the algorithm to detect packets with low signal strength.

ACKNOWLEDGMENT

The authors would like to thank Sushil Singh, Cahit Akin and Yoav Nebat for helpful discussions.

REFERENCES

[1] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, “Theory of spread-spectrum comunications - a tutorial,” *IEEE Transactions on Communications*, vol. COM-30, no. 5, pp. 855–884, May 1982.

[2] J. Proakis, *Digital Communications*. McGraw Hill, 2001.

[3] A. J. Viterbi, *CDMA, Principles of Spread Spectrum Communications*. Addison-Wesley, 1995.

[4] IEEE standards association, IEEE 802.11, 1999 edition. [Online]. Available: http://standards.ieee.org/getieee802/802.11.html

[5] A. Polydoros and C. L. Weber, “Analysis and optimization of correlative code-tracking loops in spread-spectrum systems,” *IEEE Transactions on Communications*, vol. 33, no. 1, pp. 30–43, Jan. 1985.

[6] R. B. Ward, “Acquisition of pseudonoise signals by sequential estimation,” *IEEE Transactions on Communications Technology*, vol. COM-13, pp. 474–483, Dec. 1965.

[7] R. B. Ward and K. P. Yiu, “Acquisition of pseudonoise signals by recursion-aided sequential estimation,” *IEEE Transactions on Communications*, vol. COM-25, pp. 784–794, Aug. 1977.

[8] C. W. Helstrom, *Statistical Theory of Signal Detecion*. Pergamon Press, 1960.

[9] J. I. Marcum, *Tables of Q-Functions*. Rand Corporation Report, RM-339, 1950.

[10] Xilinx Virtex-4. [Online]. Available: http://www.xilinx.com/virtex4

[11] Xilinx Virtex-II Pro. [Online]. Available: http://www.xilinx.com/virtex2pro