IMPLEMENTATION AND SIMULATION OF DS/CDMA SYSTEM UNDER FADE CHANNEL

by

Heung Sub Shim
B.Sc., Myongji University, 1996

RESEARCH PROJECT SUBMITTED PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR DEGREE OF

MASTER OF ENGINEERING

In the School
Of
Engineering Science

SIMON FRASER UNIVERSITY

Fall 2004

Approval

NAME: Shim, Heung Sub

DEGREE: Master of Engineering

TITLE OF PROJECT: Implementation And Simulation Of DS/CDMA System Under Fade Channel

EXAMINING COMMITTEE:

Stephen Hardy, Professor
Senior Member, IEEE

Tejinder Randhawa, Adjunct Professor
Member

DATE: November 15th, 2004

SIMON FRASER UNIVERSITY

PARTIAL COPYRIGHT LICENCE

Abstract

In this project, our focus lies on development of a baseband DS/CDMA simulator which is of the same structure as the real hardware but much easier to understand. In addition, the project includes both software and hardware implementations of the DS/CDMA modem which provides point-to-point communications between a station and a high speed mobile. The major blocks of the MATLAB/Simulink based simulator comply with the FPGA design of the hardware *(Altera EP1S30B956)*.

Suggestions for future work include verification of the simulator results against measurements of real data over a real communications link, and verifying and remodeling the channel model if necessary. This could be followed by point-to-multipoint or multipoint-to-multipoint communications applications, building up interactive data links.

Acknowledgements

I would like to record my appreciation to Dr. Stephen Hardy and Dr. Tejinder Randhawa for many inspiring suggestions and much helpful advice. I would also like to express my gratitude to my loving wife having been patient throughout the course of the project as well as the entire program.

Contents

List of Figures

List of Tables

1. Introduction

The DS/CDMA system finds its superiority where privacy matters, multipath fade occurs and intensive jamming sources interfere. As far as privacy is considered, the DS/CDMA system itself provides ultimate means of security in the physical level whereas other systems depend on privacies implemented in higher layers. This is the biggest reason why many military applications adopt the DS/CDMA system despite inferiority in bandwidth utilization. The DS/CDMA system undergoes relatively less multipath interference as code acquisition process discriminates the line-of-sight path from the others. The anti-jamming property of the DS/CDMA system is illustrated in Figure 1.

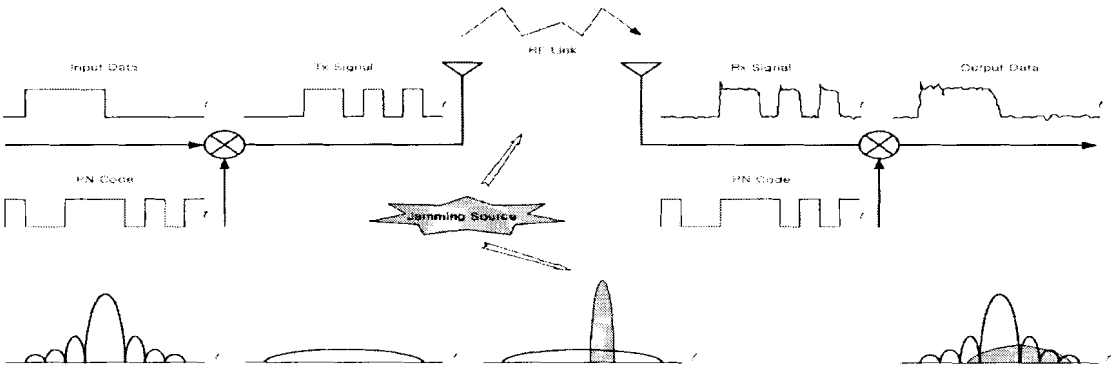


Figure 1 Anti-jamming property of DS/CDMA

On the other side, the DS/CDMA system has such negative features as increased bandwidth,

increased complexity and computational load. The system employs a PN (Pseudo-Noise) code, independent of the information data, to spread the signal energy over a bandwidth much greater than the signal information bandwidth, proportional to SF (Spread Factor, the ratio of chip rate to information data rate). The spread signal normally occupies several or tens times as wide bandwidth as the unspread information data requires. This is why the spread spectrum system is not the best choice for multimedia services. Most DS/CDMA systems invoke PN correlation characteristics in order to acquire the timing information between the transmitter and the receiver. The timing acquisition process (searcher plus code tracking loop) involves a great amount of computation, which makes the system device-dependant. In spite of the negative features, the CDMA market (including both DS and frequency hopping) is still expanding and efforts to integrate with other systems are directed all over the places on the planet, especially in many Asian countries.

The project includes both hardware and software implementations of the DS/CDMA modem which provides a station and a high speed mobile with point-to-point communications. The major blocks of the MATLAB/Simulink based simulator exactly comply with the FPGA design of the hardware *(Altera EP1S30B956)*.

The DS/CDMA modem assumes the station is the transmitter and the high speed mobile is the receiver. More specifically, the receiver is assumed to fly or travel over a multipath channel at so high a speed in a rural area that fade and Doppler shift are present. Therefore, the modem is to be capable of handling impairments of the channel varying with time. The specific features of the modem are summarized as follows.

Table 1. Features of the DS/CDMA modem

Items	Spec.
Channel	Multipath fade(3 paths i.e. 1 line-of-sight and 2 reflections)
Multiplex	DS/CDMA (Direct Sequence/Code Division Multiple Access)
Modulation/demodulation	Baseband QPSK
Spread sequence	PN m-sequence (order 17)
Spread	Complex Spread
Channel encode/decode	Convolutional encoder/Viterbi decoder (K=7, R=1/2)
Interleaver/deinterleaver	Block interleaver/deinterleaver
Chip rate	8.192Mcps
Signaling pulse	Square root raised cosine (order 48, rolloff factor 0.35)
Number of sources	3 data sources plus 1 pilot channel, TLM (512kbps), video 1 (1.024bps) and video 2 (1.024bps)
Spread factor	8 (pilot, video 1 and video 2), 16 (TLM)
Channel identification	Order 8 Walsh
Carrier frequency	400MHz
Data format	Bit stream

The modem takes into account the following requirements.

Table 2. Requirements for the DS/CDMA modem

Items	Requirements
Speed of mobile	4.8km/sec
Max Dopler freq. offset	±6.4kHz
Max carrier freq. offset	±5kHz
Max timing drift	±10ppm (over 32.768MHz clock source)
Max reacquisition time	Within 1 sec

2. Fundamentals

2.1 System structure

2.1.1 Tx (a station)

The transmitter shown in Figure 2 multiplexes pilot and three data sources (TLM, video 1 and video 2, respectively) each assigned a different Hadamard-Walsh code (whose row vectors are orthogonal to one another). Because the dimension of the Walsh matrix used is 8 by 8, the modem can multiplex up to as many as 8 independent channels.

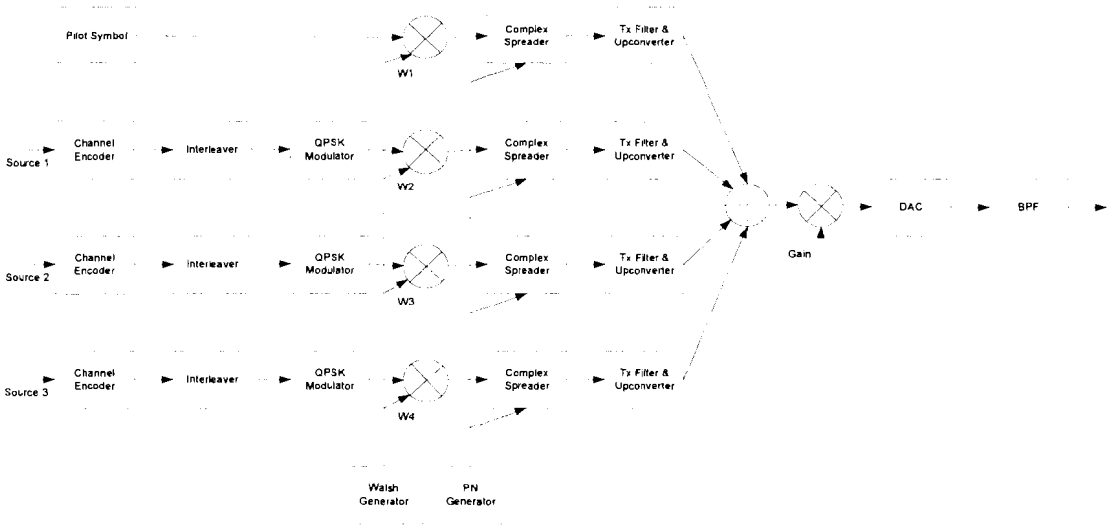


Figure 2. Tx block diagram

Channel encoder

The Forward Error Correction (FEC) block invokes a convolutional encoder of coding rate r =1/2 and constraint length K = 7.

Interleaver

Burst errors under a fading channel should be randomized so that the decoder can detect and correct the errors. The modem interleaves incoming data using a block-based interleaver. The block is constituted with a frame of 8msec.

Walsh code

Multiplexed channels should be identified at the receiver. The vector orthogonality of Hadamard-Walsh matrix provides the receiver with discrimination among data channels. Before combined, the data channels each are assigned a Walsh code of length 8 that is orthogonal to any other of the Walsh code set. Based on the known Walsh indices, the receiver restores the channels. Since the Walsh matrix is 8 by 8, up to 8 channels can be transmitted.

PN sequence

A Pseudo-Noise (PN) code sequence acts as a noiselike carrier used for bandwidth spreading of the signal energy. Its autocorrelation has properties similar to those of white noise. The autocorrelation of a PN sequence has a large peak for perfect synchronization with itself. The receiver invokes this property for synchronization. As shown in Figure 1, a PN sequence multiplied by the transmitted signal spreads the signal spectrum over the bandwidth corresponding to the chip rate. The perfect synchronization at the receiver unspreads the spread signal spectrum, but spreads any jammer to which the PN code is unknown. This is the anti-jamming property common to

DS/CDMA systems. The order 17 m-sequence are used for both I and Q channels. The generator polynomials of the m-sequence are given as follows.

$$p_I(D) = D^{17} + D^3 + 1$$
$$p_Q(D) = D^{17} + D^3 + D^2 + D + 1$$

Complex spreader

The encoded data are spread occupying a much wider bandwidth proportional to the SF. Complex spread shall be used in the project because the scheme is known to be free from I/Q interference over a phase offset and robust to jamming comparing to QPSK spread or BPSK spread. In the complex spread, a complex data symbol is multiplied by a complex PN set, different from QPSK spread where I and Q data bits are independently spread by the PN bits. This is simply a phase addition (i.e. rotation of constellation) between the data symbol and the PN sequence.

$$\begin{aligned} s_I + js_Q &= (d_I + jd_Q)(p_I + jp_Q) \\ &= (d_I p_I - d_Q p_Q) + j(d_I p_Q + d_Q p_I) \end{aligned} \qquad (2.1)$$

where s_I, s_Q, d_I, d_Q, p_I *and* p_Q *are the I and Q components of transmitted signal, data symbol and PN set, respectively* $(d_I, d_Q \in \{-1,1\}$ *and* $p_I, p_Q \in \{-1,0,1\})$.

The constellation of the PN is given as in Figure 3 below.

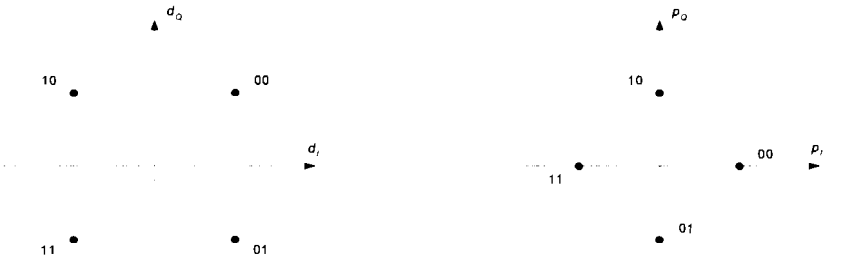


Figure 3. Constellations of data symbol and PN sequence

Tx filter

The spread signal is 4 times upsampled and fed into a 48-tap square root raised cosine

filter before transmitted. The delayed taps are 1/32.768μsec spaced. The rolloff factor of the SQRC filter is 0.35 and the total bandwidth required for transmission therefore becomes 11.0592MHz (= 8.192MHz × 1.35). The filter outputs of the 4 channels are then combined together.

2.1.2 Rx (a high speed mobile)

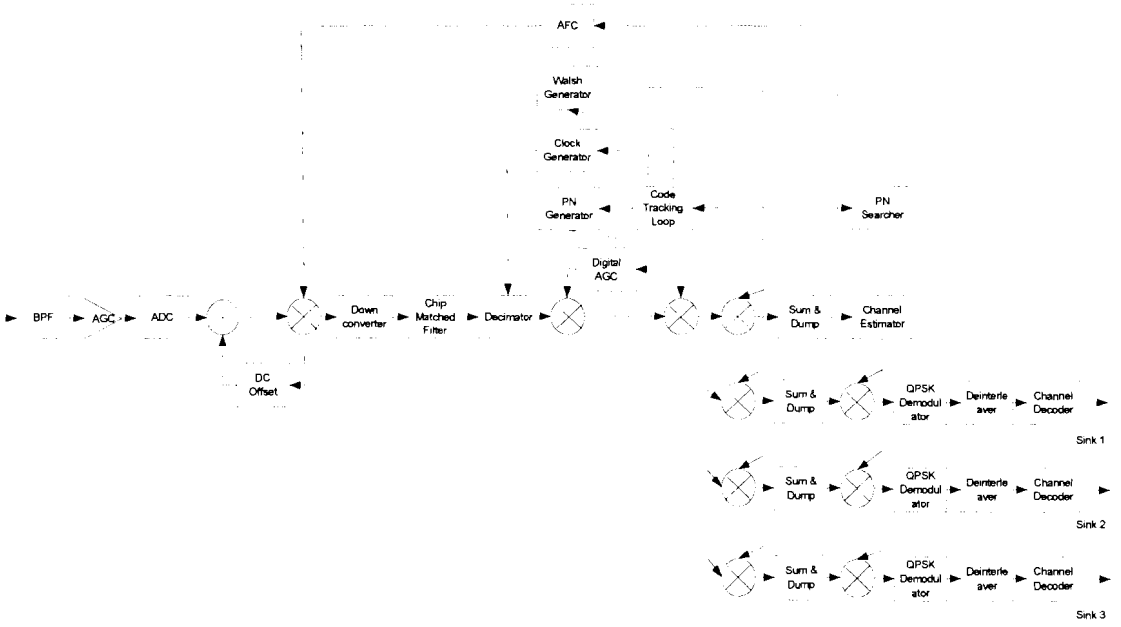


Figure 4. Rx block diagram

DC remover

The analog received signal is sampled to be digitized at the ADC. The ADC output is DC-biased due to the quantization process of the ADC. The undesired DC offset affects clipping level, bit precision, threshold and so on in the next blocks. The following 1st order filter effectively removes all DC biases accumulated along the signal path so far.

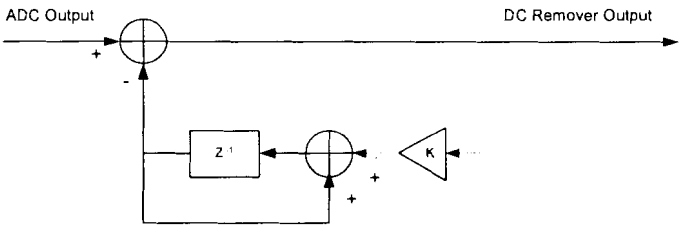


Figure 5. DC remover

Rx filter (chip matched filter)

The Rx filter is deployed in order to maximize the SNR and the basic structure of the Rx filter is the same as that of the Tx filter. Since the ADC samples the received signal 4 times as fast as the chip rate, the 1/32.768μsec spaced samples are fed into the 48-tap SQRC filter. As in the Tx filter, the rolloff factor is 0.35.

Digital AGC

As the distance between the mobile and the station increases, the received signal strength decreases. Also, the transmitted signal undergoes periodic fluctuations due to the multipath fade. This is an undesired situation because we have to adjust the threshold related values according to received signal strength. DAGC resolves the problem by regulating the magnitude of the received signal around the reference.

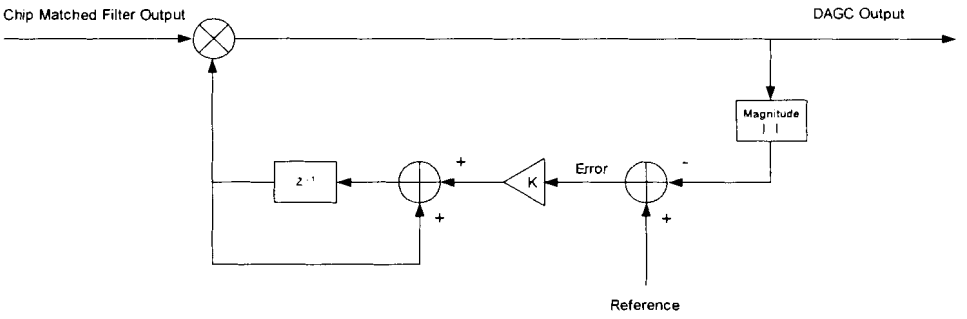


Figure 6. Digital AGC

The DAGC loop filter controls the gain so as to minimize the error generated by difference between the magnitude of the current symbol and the reference. As time goes by, the average magnitude of the chip matched filter output converges to the reference.

Such core algorithm blocks as searcher, code tracking loop and AFC are illustrated in 2.4 Algorithm blocks.

2.2 Channel model

According to the definition used in Reference 4, our channel is categorized as a wideband channel. Since our environmental assumption complies with the geometric conditions in Reference 4, we shall borrow the wideband channel model. The channel model is given by

$$h(t) = \delta(t) + \sum_{k=1}^{L-1} \Gamma_k \exp\{-j\omega_c \tau_k\} \delta(t - \tau_k) \qquad (2.2)$$

where Γ_k, τ_k and ω_c are complex gain, propagation delay in the k^{th} path and RF carrier frequency, respectively.

The first term and the second summation terms represent the line-of-sight path and the reflective paths, respectively. The number of multipath reflections L suggests a trade-off between model accuracy and complexity. Under the given geometric conditions, L=3 is found to be most suitable for the model. Therefore our multipath channel is modeled as shown in Figure 7.

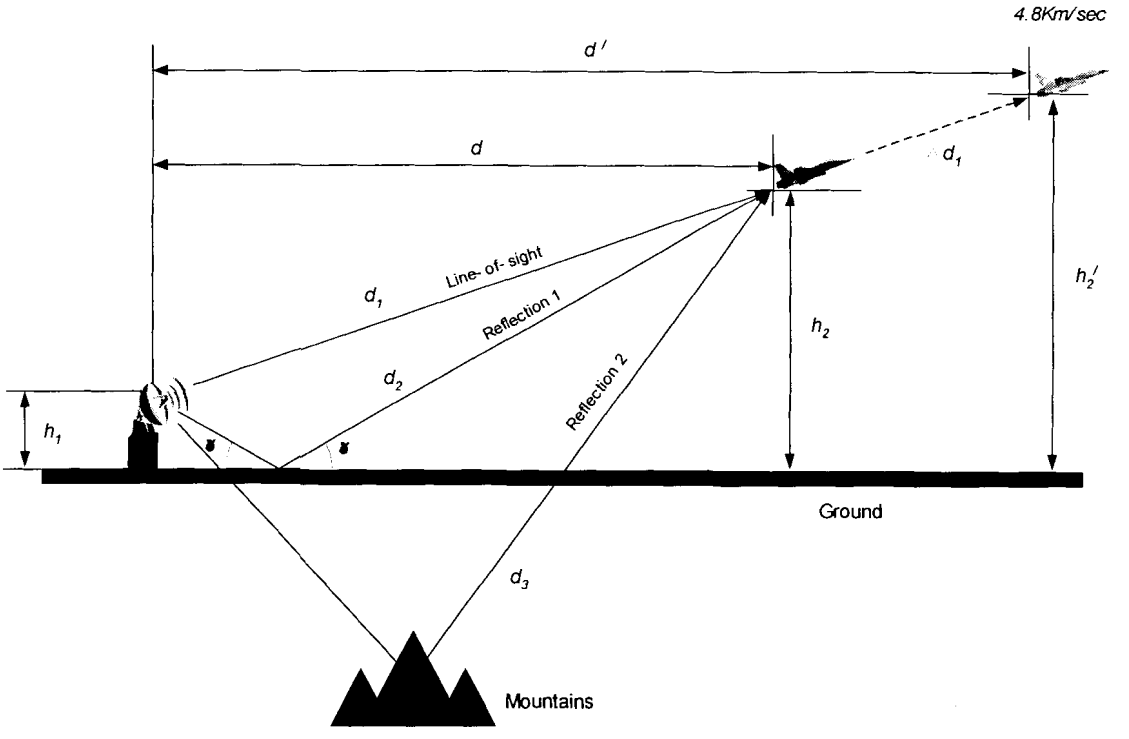


Figure 7. Channel model

In Reference 4, the first reflection is characterized by a relative amplitude of 70% to 96% of the line-of-sight amplitude and a delay of 10 – 80 ns, and 2% to 8% of the line-of-sight amplitude and the mean delay of 155ns for the second reflection. Given the channel geometry in Figure 7, we can specify our channel based on the following assumptions [4].

- The mobile travels Δd for Δt in a straight line along the line-of-sight path opposite to the station.

- The second reflection caused by mountains is characterized by a complex Gaussian random variable.

- The first reflection is of an amplitude of 96% of the line-of-sight amplitude.

- The second reflection is of an amplitude of 8% of the line-of-sight amplitude.

- The path loss is inversely proportional to the power of 3 of r km (distance between the tx antenna and the mobile).

- Since the second reflection hardly contributes to the multipath fade and has a Guassian pdf like noise, only the effect of the first one is taken into account for channel modeling.

Based on the above assumptions, we can further simplify Eq. 2.2 as follows.

$$h(t) = \delta(t) + 0.96\exp\{-j\omega_c\tau\}\delta(t-\tau) \qquad (2.3)$$

where τ is the delay for the reflection path.

The frequency domain representation of Eq. 2.3 [8] is given by

$$H(\omega) = 1 + 0.96\exp\{-j(\omega\tau + \omega_c\tau)\}$$
$$= 0.04 + 1.92\exp\{-j(\omega\tau + \omega_c\tau)/2\}\cos((\omega\tau + \omega_c\tau)/2) \quad (2.4)$$

and the magnitude spectrum $|H(\omega)|$ is

$$|H(\omega)| = |0.04 + 1.92\cos((\omega\tau + \omega_c\tau)/2)| \quad (2.5)$$

This causes a frequency-selective fading as shown in Figure 8 below.

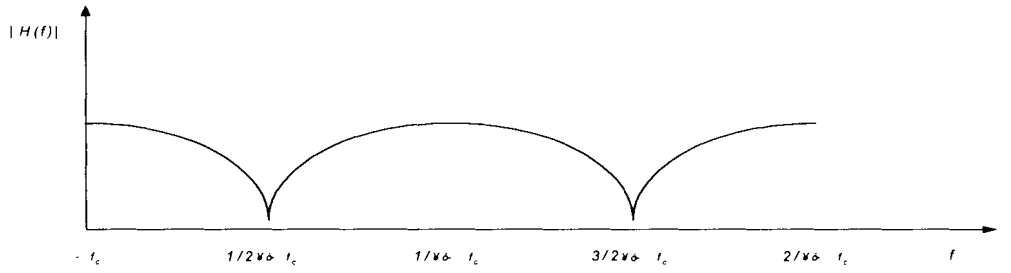


Figure 8. Frequency-selective fading caused by multipath

It is clear that the length of the reflection path becomes closer to that of the LOS path as distance between the mobile and the station gets farther (assuming the distance significantly increases while the altitude of the mobile does not change much) and consequently τ becomes smaller and smaller. However, for our application, the multipath fade will be significant for first several seconds. The Doppler frequency shift proportional to the velocity of the mobile should be compensated for. Otherwise, moving at the maximum speed, the mobile causes the maximum Doppler frequency shift so that fading occurs in a very short period of time. The AFC estimates and compensates for the Doppler shift and the carrier frequency offset due to impairments among local oscillators as well. The DAGC compensates for only loss caused by increased distance, but not for fast fade.

2.3 Algorithm blocks

2.3.1 Searcher

Searcher measures the correlations between the received signal and the PN sequence over a given size of window, compares them with the predefined threshold and acquires the PN phase information of the received signal through the verification of the candidate PN phases. Actually, the energy of the correlation rather than the correlation itself is used to trace the correct PN phase. The following equation summarizes the energy calculation process described above.

$$E = \sum_{i=0}^{N_n-1} \left| \sum_{j=0}^{N_c-1} r(iN_c + j)\, p^*(iN_c + j) \right|^2 \qquad (2.6)$$

where $r(n)$ and $p(n)$ are respectively the n^{th} decimator output and the local PN sequence generated by Searcher, and N_n and N_c are the number of non – coherent combines and the correlation length.

As we can see in the equation, an energy is an accumulation of several squared correlations, which is so called *non-coherent combining*. This is to diversify contribution to the energy lest local correlation impairments due to surges should greatly affect the energy calculation.

Dual search

According to whether the search scheme is *single* or *dual*, search resolution becomes a chip or half a chip, respectively. In this project, we adopt the dual search where Searcher works using two 1/2 chip-apart decimators in parallel. Therefore, PN phase information in 1/2 chip resolution is delivered to the Code Tracking Loop block.

Window based search

Searcher examines all the possible phases for every single window corresponding to a

frame size of 8 msec in the initial search mode and however, during reacquisition, only a few windows around the point of time of lose-lock. This is to reduce time required for reacquisition.

Detection/verification

The search process is divided into the two modes, *detection* and *verification*. Searcher screens out candidate phases based on the relatively low threshold in the detection mode and then picks the first one passing the verification test where the threshold is set much higher than the one in the detection mode. This two-stage search greatly improves the false alarm characteristic of the modem.

Slew

Losing the timing sync, the receiver should reinitialize the PN clock generator so that Searcher starts to trace the PN phase of the received signal based on the last acquisition information. Searcher makes the PN clock slower or faster depending on the information so that the last PN phase in lock comes in the middle of the window to be tested for reacquisition. This is what is so called *slew*.

On acquiring the PN phase information of the received signal, Searcher sends it to the code tracking loop block where the sample timing is recovered.

2.3.2 Code Tracking Loop (CTL)

In the DS/CDMA system, timing recover is in general performed in two stages, PN phase acquisition by Searcher and fine sample timing recovery by Code Tracking Loop. Searcher acquires the initial phase of the received signal within 1/2 chip and, based on the phase information from Searcher, Code Tracking Loop traces the exact sample time

as well as the timing drift between the station and the mobile. Most code tracking algorithms invoke PN autocorrelation characteristics. Timing recovery in the past was done by an analog PLL circuit. As the Digital Signal Processing (DSP) technology evolves, the analog circuit has been integrated on a single chip. Our CTL structure complies with the well-known early-late gate algorithm in Figure 9, one of the most popular CTL techniques [6].

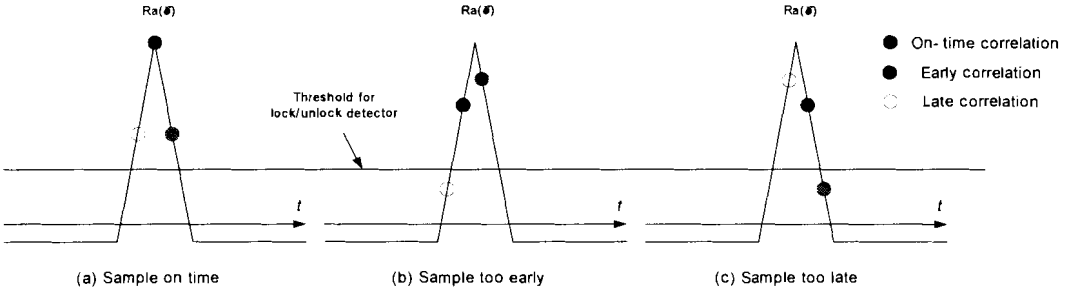


Figure 9. Early-late gate algorithm

We take the 1/2-chip early sample by decimating the input to CTL 1/2 chip earlier than the current on-time sample time and delay the sample by 1 chip to obtain the 1/2-chip late sample. Subtraction of late correlation from early correlation generates an error. The CTL filter in Figure 10 [6] is so designed as to minimize the error. Depending on the sign of accumulative errors, the CTL clock is slewed to drive a 1/4-chip late (Figure 9-(c)) or early (Figure 9-(b)) decimation. Once CTL converges, the early and late correlations will be equal to each other (Figure 9-(a)). In the meanwhile, the lock/unlock detector repeatedly examines the status of the TR block, observing in a given period of time how many times the correlator outputs have passed the threshold test. In order to get a 1/8-chip timing resolution, we introduce a linear interpolator into our system. The mean value of two consecutive samples is assigned to the sample in the middle of the two so that we can choose any of the 8 sampling moments available per chip. The structure of CTL is depicted in Figure 10.

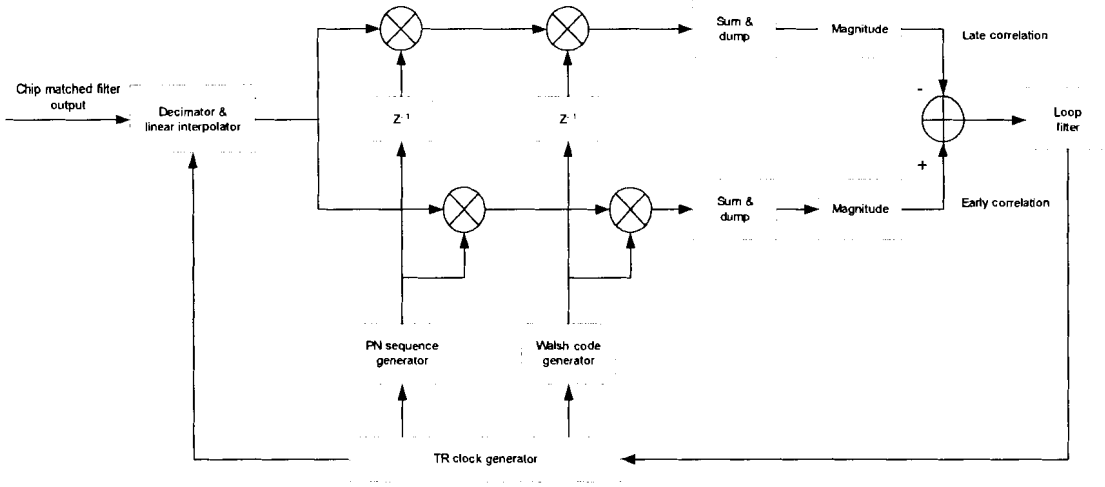


Figure 10. Code Tracking Loop

The first-order loop filter in Figure 11 can track out both sample time and frequency error (timing drift). The filter consists of two paths, a *proportional path* and an *integral path*. The former can trace a sampling time error while the latter can track out a sampling frequency error.

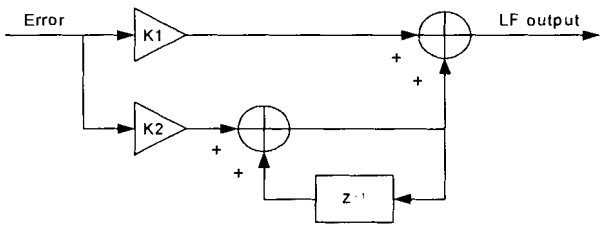


Figure 11. CTL loop filter

2.3.3 Automatic Frequency Control (AFC)

The received baseband signal contains undesired frequency offset that results in the signal constellation turning around at the rate of the offset. The frequency offset is clarified by two offset sources, *carrier frequency offset* by a discord between Tx and Rx local oscillators and *Doppler frequency offset* by mobility of the terminal. Doppler frequency offset is given by

$$f_d = \frac{f_c v}{c}\cos\theta \qquad (2.7)$$

where f_c, v, c and θ are carrier frequency, velocity of mobile, velocity of light and moving direction, respectively.

Assuming the maximum velocity of the mobile is 4.8km/sec, the carrier frequency is 400MHz and the distance between the station and the mobile is far enough for the cosine term to be eliminated, we have a maximum Doppler frequency of f_d = ±6.4kHz. If the maximum carrier frequency offset is ±5kHz, the AFC block should be able to handle up to a maximum frequency offset of 11 ~ 12kHz. If the frequency offset is not properly compensated for, the constellation of the demodulated signal will continuously turn around, which causes symbol detection to be degraded or even impossible. The AFC block diagram is as shown in Figure 12.

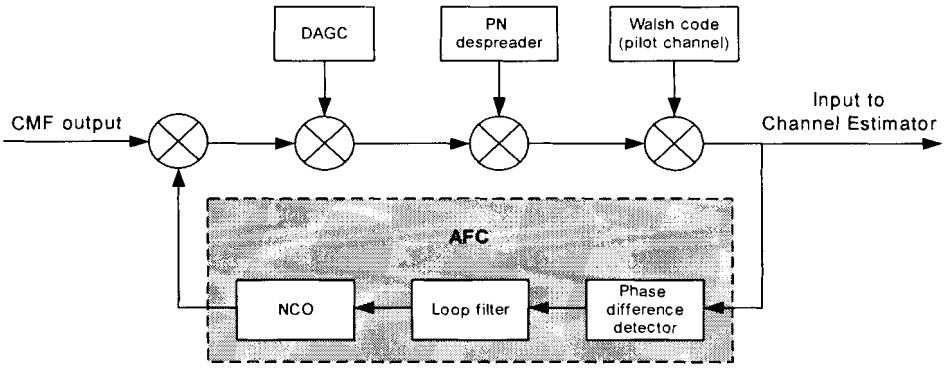


Figure 12. AFC block diagram

Phase difference detector

Before calculating and compensating for frequency offset, we should have a good strategy to read changes in the pilot symbol constellation. We first divide the signal plane into 16 sectors, each with its own sector ID. (assigned 0 to 15 counterclockwise as in Figure 13). A number of pilot symbols are accumulated together to reduce background noise and dumped after being read by the Phase Difference Detector block. The block then detects change in the pilot constellation by subtracting the previous output of the

accumulator from the current output. Therefore a phase difference can be any integer value between –15 and 15. This is illustrated in Figure 13. It is very important to determine an observation period so carefully that a phase change between two consecutive accumulator outputs should not exceed π rads. In other words, an observation period must be so carefully chosen as to follow up the maximum frequency offset.

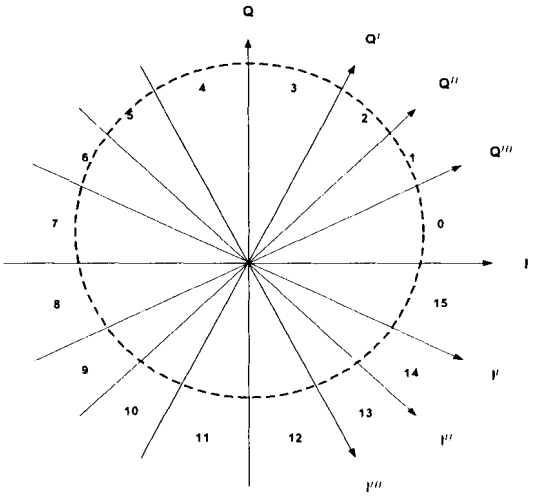


Figure 13. Sixteen sectors to read change in the pilot symbol constellation

In Figure 13, the bisectional planes by the axes I, I', I'', I''', Q, Q', Q'' and Q''' correspond to the signs of the equations S_Q, $S_I + 2S_Q$, $S_I + S_Q$, $2S_I + S_Q$, S_I, $2S_I - S_Q$, $S_I - S_Q$ and $S_I - 2S_Q$ respectively, where S_I and S_Q are the in-phase component and quadrature component of the despread pilot symbol $S_p = S_I + jS_Q$. The above sectoring allows us to identify the sector in which the phase of the symbol falls by simply investigating the signs of a symbol projected onto the predefined axes. Therefore any of the sixteen sectors is fully specified by the signs of the eight equations representing the eight axes.

There is a problematic situation. If the current pilot constellation falls in the sector 8 with the previous one in the sector 0, rotating clockwise, Phase Difference Detector adds not

-8 but +8 to the phase accumulator because the detector doesn't care the current direction of the constellation revolving. In order to avoid such an undesirable situation, we establish a mapping of phase difference to phase accumulation below.

Table 3. Mapping of phase difference to phase accumulation

Phase Difference	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
PDD output	0	1	2	3	4	3	2	1	0	-1	-2	-3	-4	-3	-2	-1

Loop filter

A phase difference to be accumulated should be scaled down by the first loop gain K1 for fine frequency compensation. The second loop gain K2 determines a step size of compensation. The LF output is translated into a ROM address that corresponds to the complex value to compensate for the offset. The structure of the AFC loop filter is shown in Figure 14.

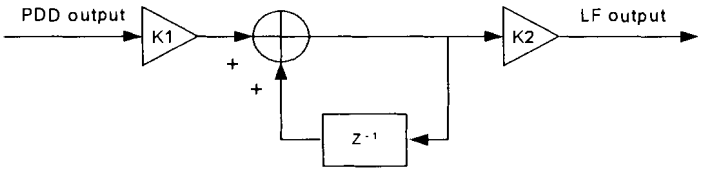


Figure 14. AFC loop filter

NCO (Numerically Controlled Oscillator)

The NCO is simply a ROM table that contains complex values corresponding to angles in degree, that is, the NCO reads out the ROM table according to multiples of the LF output. As the AFC converges, the frequency offset is left with a residual frequency offset plus a random channel phase. This channel phase is compensated for in the Channel Estimator (CE) block.

Lock detector

The Lock Detector block tests convergence of the AFC by the residual frequency. The calculation of residual frequency is obtained by observing in a given period of time changes in the signs of the I and Q components of the frequency offset-recovered symbols.

2.3.4 Channel Estimator (CE)

A transmit channel is specified by an amplitude and a phase, both time-variant. In this project, no matter what the instantaneous SNR may be, the magnitude of the received signal is regulated around the predetermined reference. Also, a QPSK system compared with an M-PSK system is relatively insensitive to symbol amplitude. For these reasons, we shall compensate for the channel phase only. The frequency offset-recovered pilot symbols form a constellation cloud deviated from the original pilot constellation by a random channel phase varying with time. We are to invoke a simple Moving Average model to estimate the time-variant channel phase.

In Figure 15, the output of the tapped delay line is simply a mean of N pilot symbols. We can estimate a channel phase by taking the complex conjugate of an MA output and subtracting the phase of the transmitted pilot signal known to the receiver. The channel phase is compensated for at the middle point of the line as shown in the figure above.

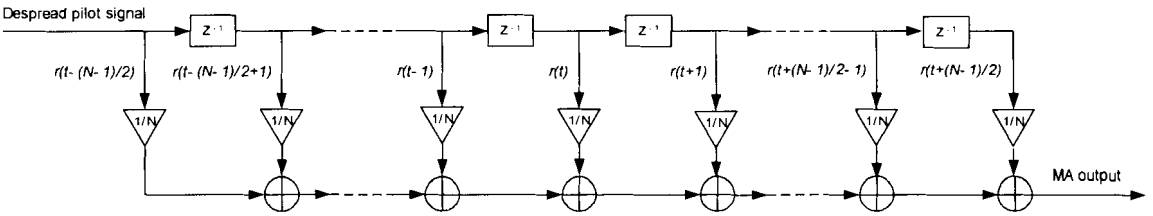


Figure 15. Moving Average model for channel estimation

3. MATLAB/Simulink Based Software Model

The MATLAB/Simulink model presents a baseband physical layer of DS/CDMA system. The software includes a Simulink schematic design file and eleven MATLAB m files for the configuration of simulation. The implementation of the simulator is in a full compliance with the hardware implementation excepting some blocks (i.e. searcher, Viterbi decoder). In other words, the simulator provides a clear inside view of the hardware implementation.

3.1 Schematic design

3.1.1 Top design

In Figure 16, the top design block enables one to set up relevant parameters by algorithm blocks before simulation. Further calculation required parameters are specified in *sim_setup.m* and *channel_param.m*. The Timing Drift, Carrier Offset and AWGN Channel blocks representing system impairments are deployed between the Transmitter and Receiver blocks. Their parameter values are specified in Table 1 and Table 2 of Chapter 1.

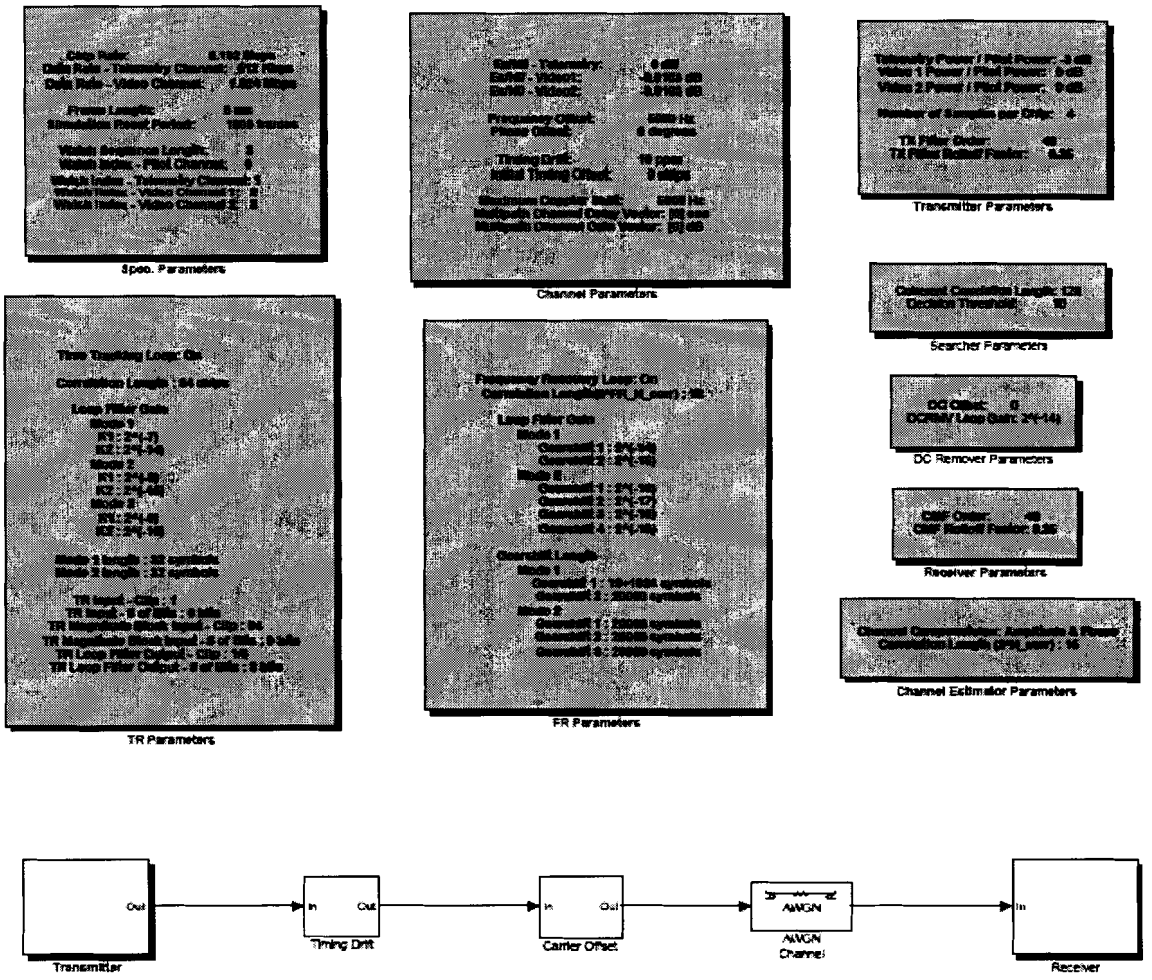


Figure 16. Top design of the DS/CDMA model

3.1.2 Transmitter

The Transmitter design is straightforward itself. Three data sources each are encoded, block-interleaved, QPSK-modulated, assigned a Walsh code, spread by the PN sequence and filtered by the pulse shaping filter. The pilot source is an exponent of a magnitude of 1 and a phase of $\pi/4$, that is, $e^{j\pi/4}$. This is used as the reference for all algorithm blocks. Different line colors represent different sampling times.

3.1.3 Receiver

The Receiver block is constituted with several subsystem blocks such as Searcher, AFC,

Code Tracking Loop, together with some smaller blocks. Excepting that one is component-based and the other is gate-based, there exists no big difference between the MATLAB/Simulink model and the actual FPGA design.

As illustrated in Figure 18, the received signal is quantized into 6-bit samples, dc-removed, chip match-filtered by a SQRC filter and gain-controlled by the DAGC. The Searcher then searches for the PN phase inherited in the received signal using the DAGC output. Only those correlations that have exceeded the lower threshold are fed into the verification block. The block picks out the first one that has passed the higher threshold test and delivers the phase information of the correlation to the TR block that includes the CTL. Once the PN sync is acquired, the CTL and the AFC are simultaneously enabled. Assuming that sampling time is not perfect and a certain amount of frequency offset is present, the two blocks should work out though each other is not in convergence. Different from the Searcher, they process the despread pilot signal. The CTL searches for the optimum sampling moment as well as tracks out the timing drift based on the early-late gate algorithm. The AFC calculates the frequency offset to be compensated for and reads out the ROM table (NCO) according to the calculation. In the meanwhile, the CE extracts the channel phase information from the phase of the despread pilot signal. Since the CE compensates for the channel phase only, no equalizer is implemented in the receiver. Along the data signal path, the receiver QPSK-demodulates and deinterleaves the sampling time and frequency offset recovered data signals. In order to align the deinterleaver block boundary with the frame boundary, we have deployed an extra delay block. This is crucial because both interleaver and deinterleaver are block-based and, with no boundary sync between the frame and the interleaver/deinterleaver, the deinterleaved data at the receiver will be messy. The output of the deintereleaver is finally decoded by a Viterbi decoder. A combination of

interleaving and channel encoding recovers burst errors occurring through the transmission link.

The hierarchical structure of the Simulink schematic design ultimately provides users with component level understandings over system level blocks. Therefore, refer to the lower blocks for further details. The specifications of the DS/CDMA Simulink model are summarized in the next section.

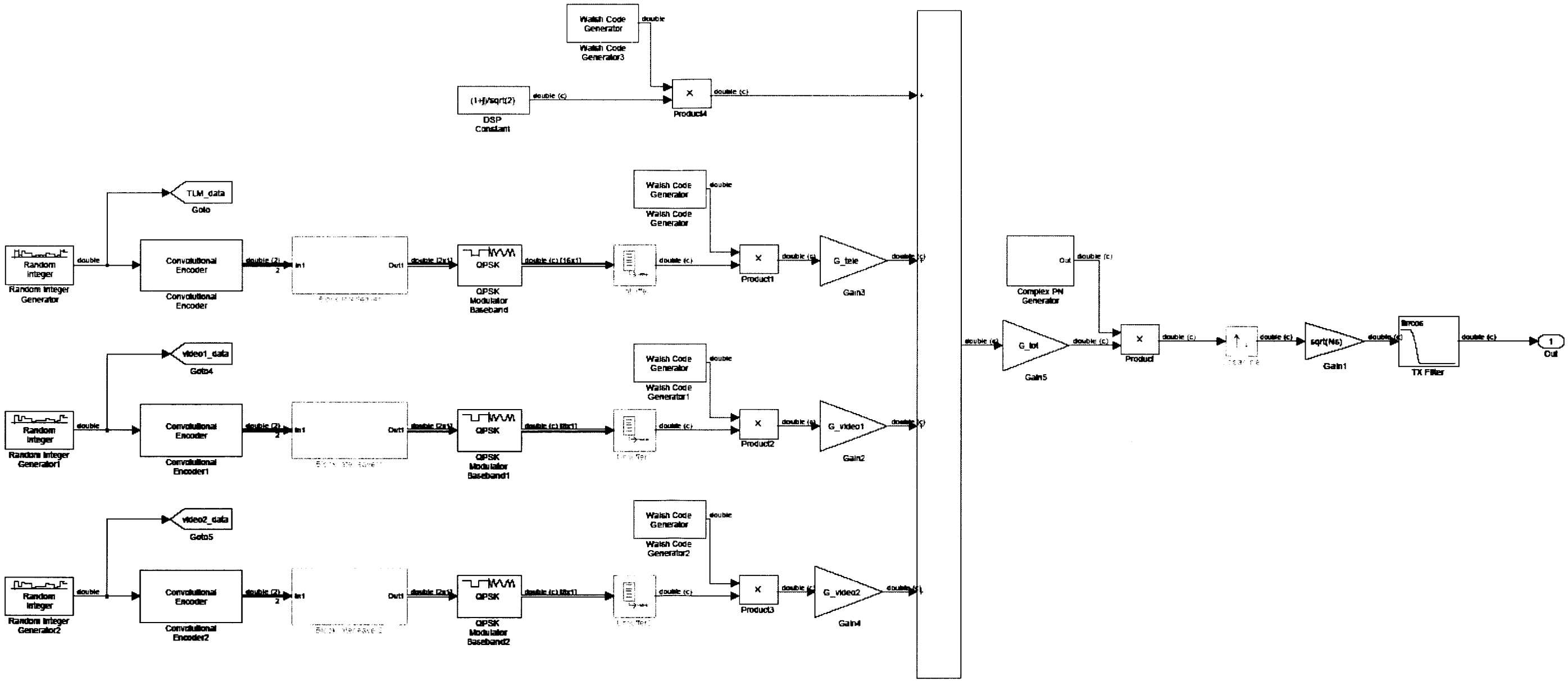


Figure 17. Transmitter

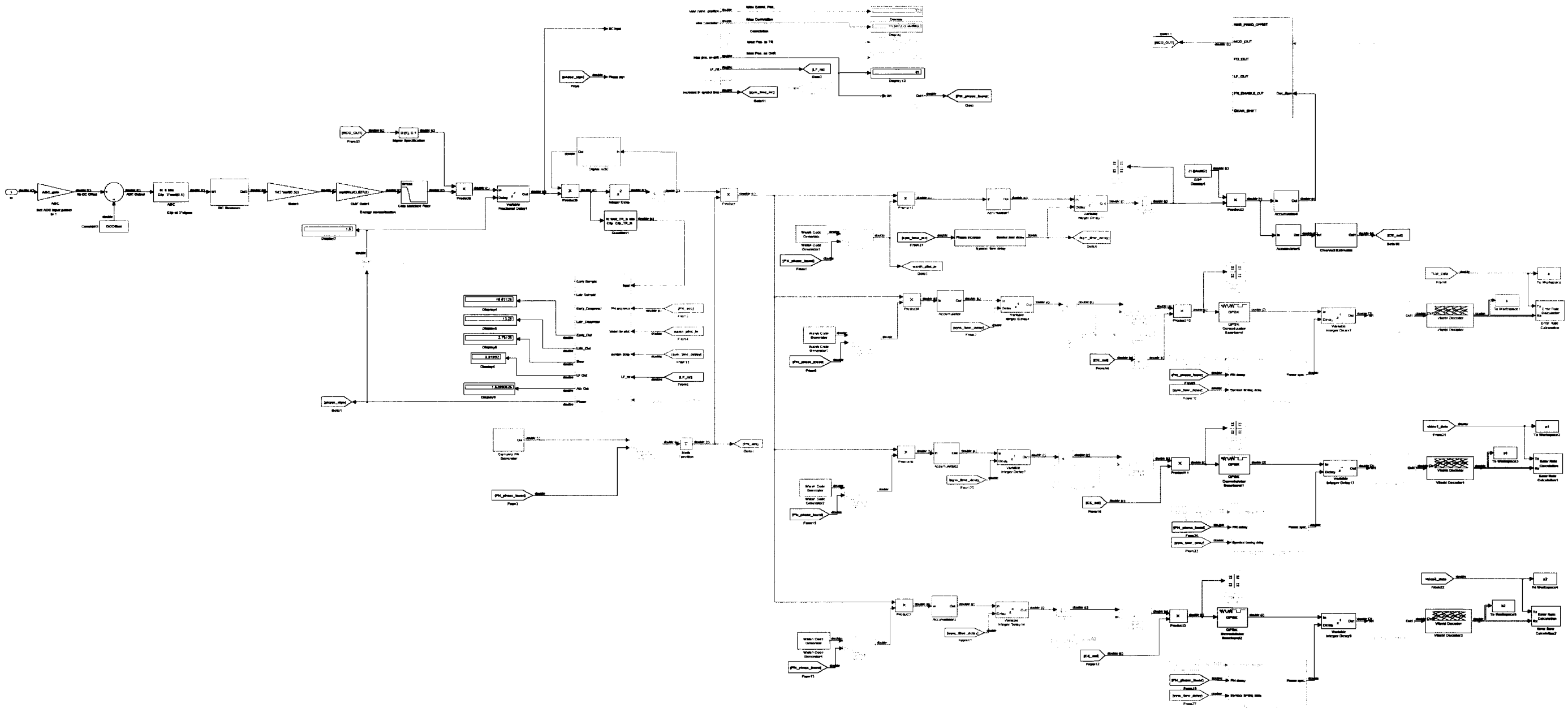


Figure 18 Receiver (continued on next page)

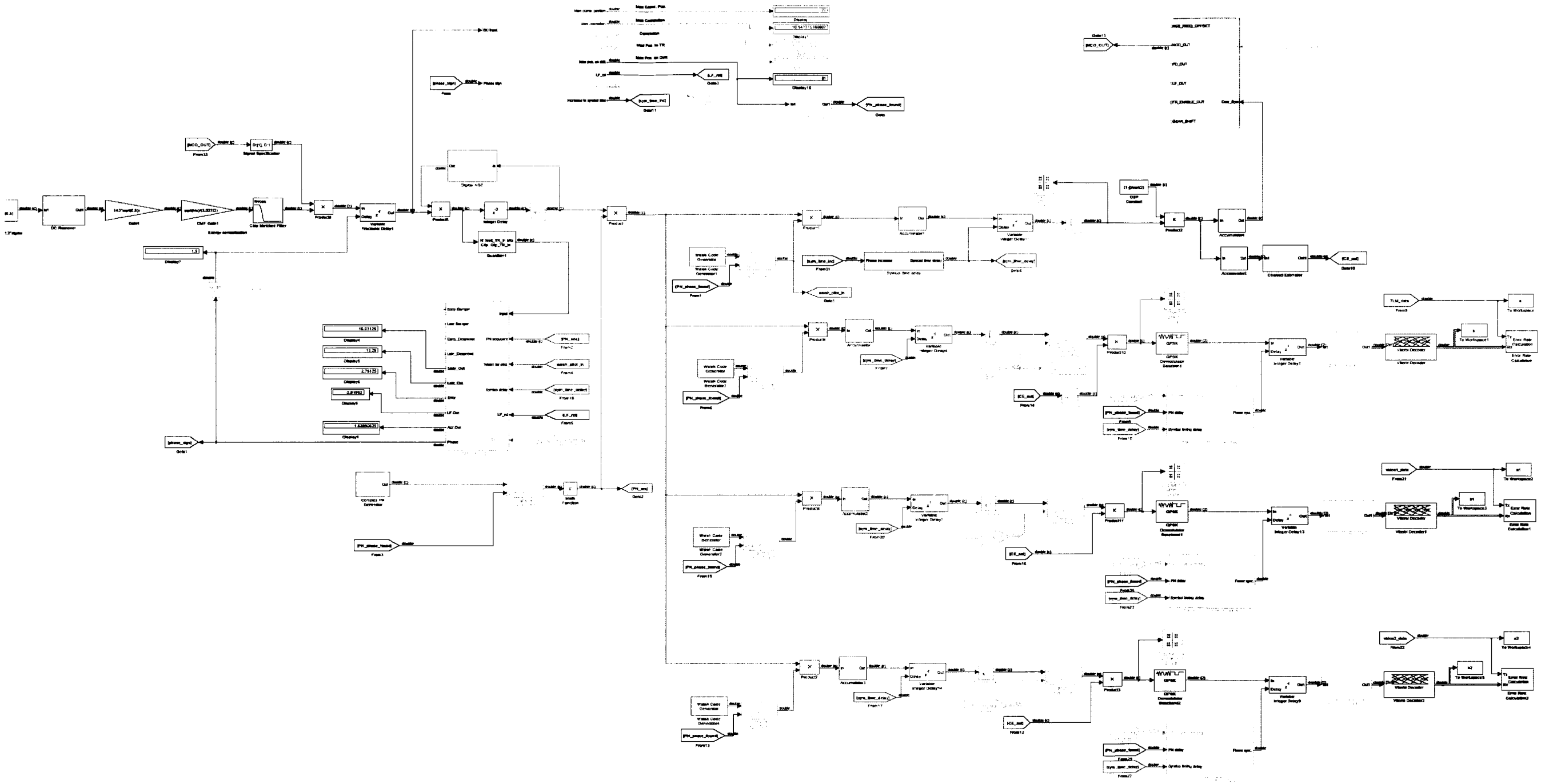


Figure 18 Receiver (continued from previous page)

3.2 Features and key parameters

Table 4. Specifications of the DS/CDMA Simulink model

Items	Spec.
Multiplex	DS/CDMA
Sources	Telemetry(512kbps), Video 1(1.024Mbps), Video 2(1.024Mbps)
Chip rate	8.192Mcps
Spread sequence	PN m-sequence (order 17)
Spread factor	8 (Pilot, Video 1/2), 16 (TLM)
Encoder	Convolutional encoder (K =7, R = 1/2, g = [171 133])
Modulation/demodulation	Baseband QPSK
Interleaver/deinterleaver	Block interleaver/deinterleaver
Tx/Rx filter	Square root raised cosine (order = 48, rolloff = 0.35)
Upsample	4×
System impairments	Timing Drift (10ppm), Carrier Offset (10kHz)
Channel identification	Order 8 Walsh
Timing recovery	Searcher, Code Tracking Loop (digital)
Carrier recovery	AFC (digital)
Channel estimate	Channel Estimator (phase compensation only)
Other algorithms	DC remover, Digital AGC
Decoder	Viterbi decoder (K = 7, g = [171 133], hard decision)

4. Hardware Implementation

The Project mainly focuses on algorithms required for the DS/CDMA system and the software implementation of it. In the meantime, the DS/CDMA hardware supports the feasibility of the system constructed on the basis of the proposed algorithms. Hence we shall briefly demonstrate only the consistency of the hardware with the software in this chapter.

4.1. Devices

The use of all devices but the CPU in Table 5 is very obvious. The CPU allows for initialization of the block registers in the FPGA design, control among the algorithm blocks and communications interface with the external for monitoring.

Table 5. Device list of the DS/CDMA hardware

Devices	Part name	Function
FPGA chip	EP1S30B956C6 (Stratix family), *Altera*	FPGA design
Configuration device	EPC8QC100, *Altera*	Non-volatile memory
FPGA design tool	Quartus II 3.0, *Altera*	FPGA design
Master clock	TCXO (32.768MHz), *Jung Technology*	Reference clock source
CPU	PIC18F452, *Microchip Technology*	CPU
D/A converter	AD9762, *Analog Devices*	D/A conversion
A/D converter	AD9214, *Analog Devices*	A/D conversion
Transformer	T1-1T, *Minicircuits*	Baseband/RF circuit separation
Decoder	S2060CCR (Viterbi decoder), *Intel*	FEC Vitervi decoder (3-bit, soft decision)

Figure 19 shows the actual prototyping board of the DS/CDMA modem.

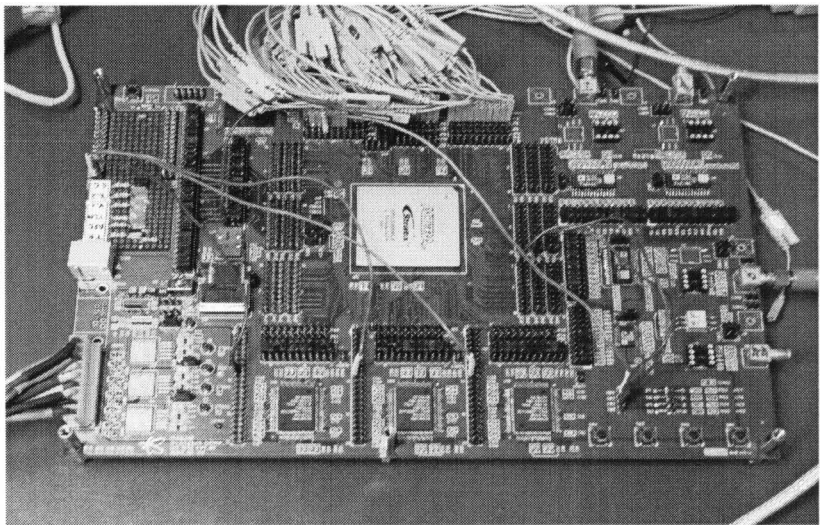


Figure 19. Prototyping board of the DS/CDMA modem

4.2 FPGA design

Since the gate level FPGA design of the DS/CDMA system is out of scope in the project, we shall provide an overview of the design only.

We have made the prototyping board as simple as possible, by implementing on a single FPGA chip traditional analog blocks such as DC bias remover, AGC, PLLs for various purposes, and so on. All algorithm blocks but the decoder are integrated on the single FPGA chip. Considering the complexity of implementation, we have deployed commercial Viterbi decoder chips (S2060CCR) as many as the number of data sources.

The algorithm blocks are designed on the basis of a schematic design scheme using megafunctions and components available in Quartus II and the FPGA device. As mentioned earlier, the gate-level FPGA design of the major algorithm blocks is, in all aspects, consistent with the component-level MATLAB /Simulink design of those.

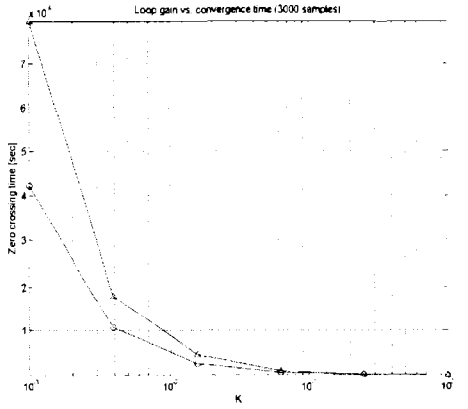
The CPU is introduced to initialize the block registers in the FPGA chip and the decoder chips, describe the operation of the algorithm blocks and provide monitoring of processes and operations by steps or blocks. The CPU reads from the FPGA the peak correlations and their positions in a descending order, compares them with the lower threshold, followed by the higher threshold with the first test passed and, if the peak correlation is verified to be valid, enables the TR and the AFC. The reason why the CPU tests multiple peak correlations is to reduce false alarms due to multipath fade. Once the system goes into a lock status as the CTL and the AFC converge, the CPU listens to the interrupt port to handle interrupts issued by the algorithm blocks. While in operation, the CPU keeps collecting important information including lock/lose-lock status, algorithm convergence, decoder out-of-sync status and so on and, on an interrupt occurring, issues the countermeasure.

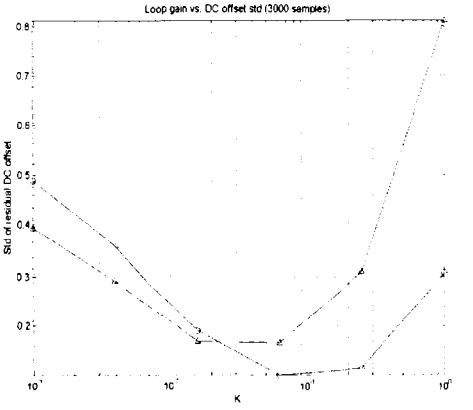
5. Simulation

5.1 Block-wide simulation

DC remover

Figure 20-(a) shows zero-crossing time by different K for 3000 samples, which indicates how fast the DC remover block converges. The zero-crossing time means the point of time at which the sign of the residual error between initial DC offset and compensation changes for the first time. According to K, the standard deviation of DC residues is plotted as in Figure 20-(b). It is obvious that K greater than 10^{-2} no longer improves the jitter performance of the loop filter. As far as hardware implementation is concerned, order of K is strictly involved in the bit precision of the DC remover block. In our case, a selection of $K = 2^{-5}$ or 2^{-6} offers a good compromise among convergence time, jitter and hardware complexity.



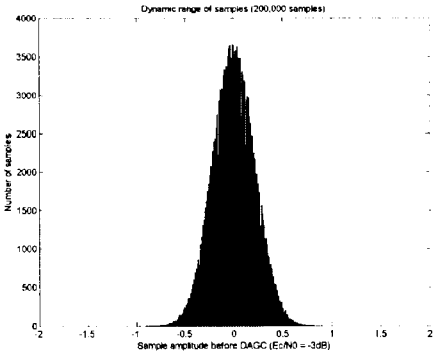


(a) K vs. zero-crossing time (b) K vs. std of the residual DC offset

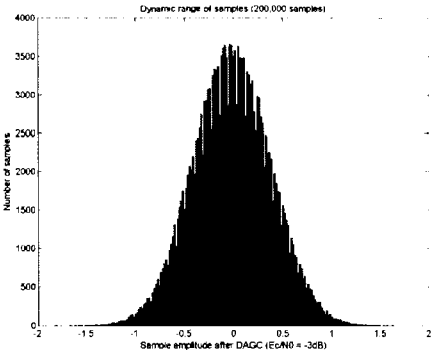
Figure 20. Zero-crossing time and std for different loop gains (o: *Ec/N0* = ∞, Δ: *Ec/N0* = -3dB)

Digital AGC

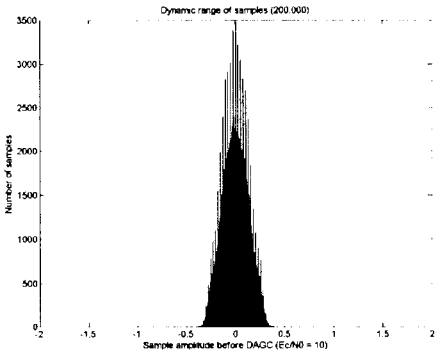
Depending on channel conditions and SNR, an amplitude range of received samples varies. This may force the system to apply different thresholds and different bit precisions over the following blocks. The DAGC clears away such worries, regulating the magnitude of the received signal around the reference. The DAGC loop filter has been found to be relatively insensitive to the loop gain K. Figure 21 illustrates examples of dynamic range of samples and the effect of the DAGC.



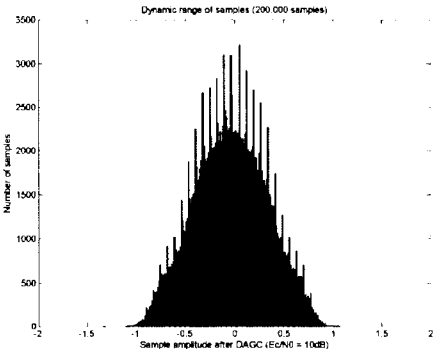
(a) Before DAGC (Ec/N0 = -3dB)



(a) After DAGC (Ec/N0 = -3dB)



(a) Before DAGC (Ec/N0 = 10dB)



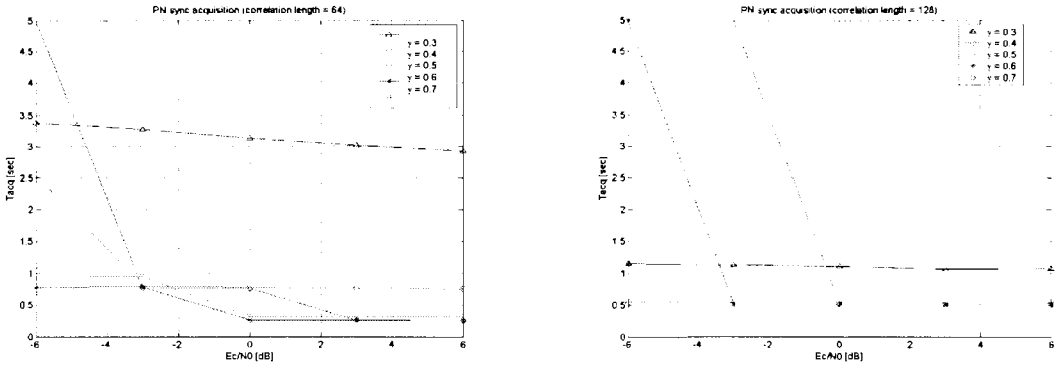
(a) After DAGC (Ec/N0 = 10dB)

Figure 21 Dynamic range of received samples

Searcher

It is not an exaggeration to say that timing recovery accounts for more than 70% of the whole gravity in designing a CDMA baseband system. Without a robust synchronization, we can never obtain a high BER performance. Furthermore, the system may lose the sync so that it may have to be initialized again. This is an undesirable situation that should be avoided if possible. Therefore, under a low SNR condition, the synchronization process of a CDMA system should be capable of maintaining the sync up to a certain measure of tolerance even though the BER performance of the system is so poor that we may have many of the received bits in error.

Since our timing recovery process adopts a traditional correlation based method, a threshold to discriminate the correct PN phase should be so carefully determined that the system can normally maintain the sync under the above condition, up to a certain measure (-6dB for our case). In Figure 22, threshold γ is a normalization of the mean correlation taken under SNR = ∞. We assume that a false alarm penalty accounts for 1msec and the system operates in presence of a timing drift of 10ppm and a carrier frequency of 10KHz. The simulation result , that γ = 0.5 for N=128 is the best choice for our system. A too low γ causes more frequent false alarms while a too high one does not normally work under a low SNR condition. In the hardware implementation, the PN acquisition process is classified into two mode*s initial acquisition* and *reacquisition* so that, in reacquisition mode, the system can reduce time required for acquisition by searching for the correct PN phase using the information of the most recent lose-lock position.



(a) PN sync acquisition for N = 64

(b) PN sync acquisition for N = 128

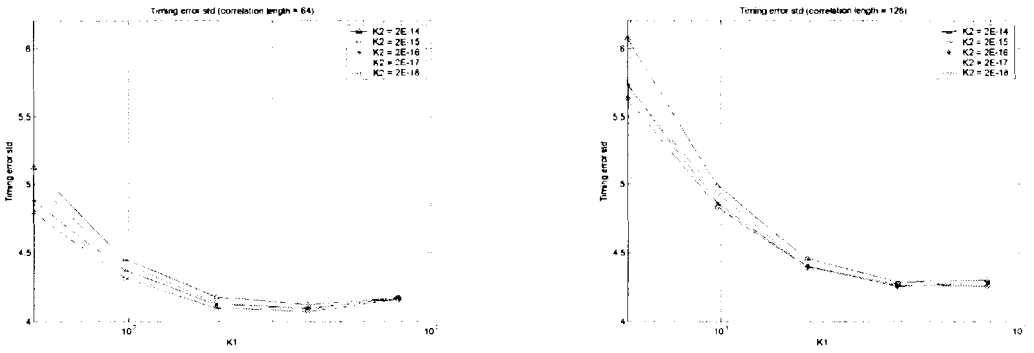
Figure 22. PN sync acquisition time by thresholds

CTL

Cooperating with the Searcher, the CTL tracks out the optimum signaling moment of the received signal in presence of a timing drift of 10ppm and a maximum frequency offset of

10KHz. In the same manner as the Searcher, the CTL should normally operate under a low SNR condition. The CTL is a first-order loop filter which consists of two paths, proportional path and integral path. The gains K1 and K2 determine the jitter performance of the filter.

Our criterion to choose the best K1 and K2 is standard deviation of timing errors after convergence. The simulation results in Figure 23 illustrate that, if K1 and K2 increase, the std of the timing errors proportionally increases because they are not fine enough to filter out the background noise and, if K1 decreases, the std of the timing errors accordingly increases because the CTL cannot track out the timing drift. This timing drift effect is severer in case of N = 128, compared with N= 64. Since the smaller K is, the more bits are required in the actual implementation, it is important to choose the best between the tradeoffs.



(a) Timing error std for different K (N = 64)

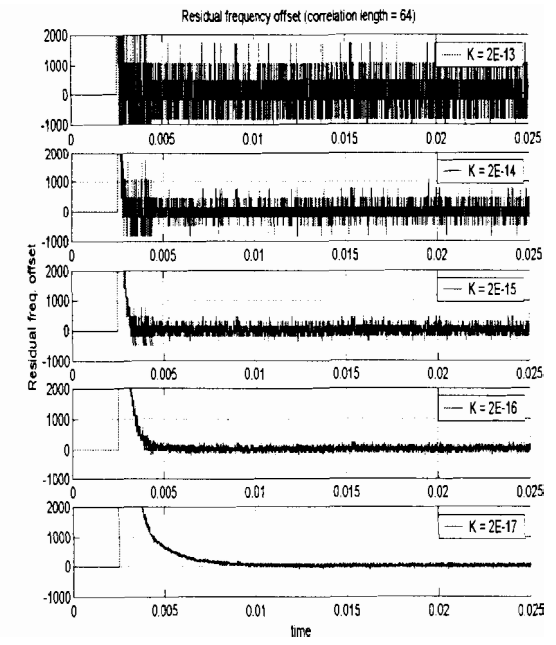
(b) Timing error std for different K (N = 128)

Figure 23. Timing error std for different K

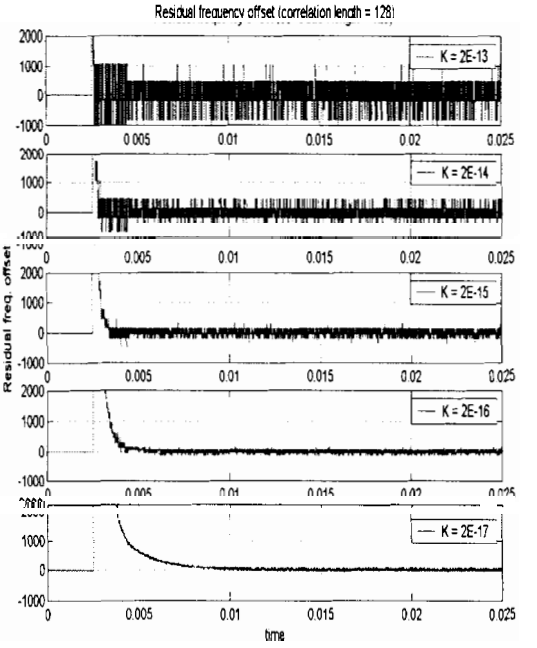
AFC

The AFC estimates and compensates for carrier frequency offset by a discord between Tx and Rx local oscillators and Doppler frequency shift by mobility of terminal. The algorithm assumes a maximum frequency offset of 10KHz and a SNR of –6dB. If the

algorithm works out, updates the NCO output every 128 chips and a residual frequency offset is within 500Hz, the frequency deviation is just 1/128 cycle (2.81 degree in constellation) at most by the next update and hardly affects the system performance. Therefore we set the convergence threshold to |500Hz|. Figure 24 shows the convergence process of the AFC starting with an initial offset of 10KHz. For the same K, the figure clearly shows that the case of N = 128 outperforms that of N = 64. This is because the background noise decreases as correlation length increases. Based on the convergence threshold above, the pull-in time accounts for about 5msec for the worst case ($K = 2^{-17}$). The reason why there is no significant difference in pull-in time between the two cases is that they are involved in estimate period only. The compensation period is the same for both and, therefore, the AFC compensates for the frequency offset in a fixed period based on a given period of observation. The compensation period depends on the sampling rate after decimation. The AFC and TR blocks stay in an idle state until the CPU enables them on obtaining the PN sync by the Searcher.



(a) Residual frequency offset for N = 64



(b) Residual frequency offset for N = 128

Figure 24. Residual frequency offset for different K

Figure 25 compares jitter performances of the AFC filter after convergence. Two cases, N = 64 and N = 128, are considered. The figure shows $K = 2^{-16}$ for N = 128 is the best. As in the case of the TR, there is a tradeoff between bit precision and filter performance.

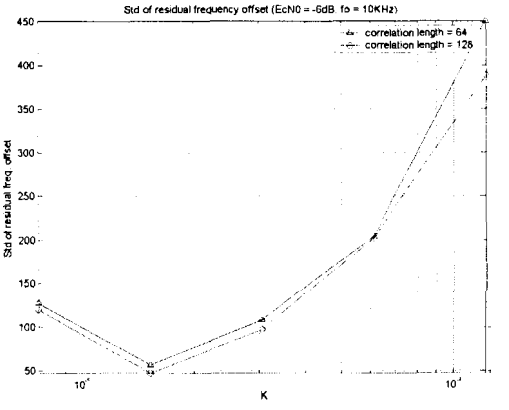
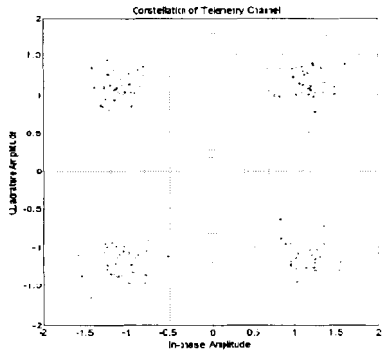


Figure 25. Std of residual frequency offset for different K

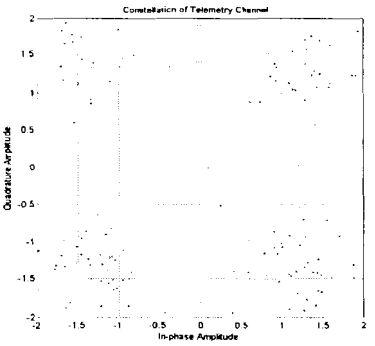
5.2 System-wide simulation

The simulations in this section have been performed on the basis of the results from the previous section that describes how to determine major parameters for each block. In other words, we assume that all the algorithms described in the previous sections are invoked and also the block-wide simulation results are reflected. The simulation environment complies with Table 4 in the section 3.2.

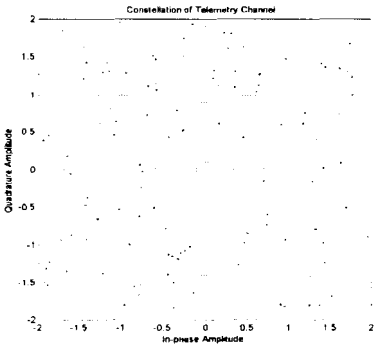
Figure 26 shows constellations of the TLM channel by SNR. The ADC output at the receiver contains a certain level of background noise because the ADC quantizes the received signal into six bits. We have not measured the quantization noise level and SNR hence is used in sense of relative comparison. The actual SNR may be somewhat lower than the SNR settings in Simulink. As we can infer, the above figures show the scatter plots of CE (Channel Estimator) outputs for different SNR. The results themselves are very obvious.



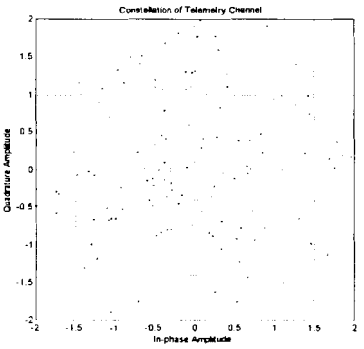
(a) Constellation of TLM channel for Ec/N0 = 20dB



(a) Constellation of TLM channel for Ec/N0 = 10dB



(a) Constellation of TLM channel for Ec/N0 = 0dB



(a) Constellation of TLM channel for Ec/N0 = -10dB

Figure 26. Constellation of TLM channel for different SNR

Figure 27 depicts the BER performance of the DS/CDMA model. We set the transmit channel power ratio to 1: 2: 1: 1 (Pilot: TLM: Video 1: Video 2, respectively) since TLM data should be of a higher requirement than real-time data. Nevertheless, the simulation shows there is no significant BER gain for the TLM channel. Our DS/CDMA model has proven to work relatively well under a low SNR condition. This is because a certain degree of random errors can be detected and corrected by a channel encoder.

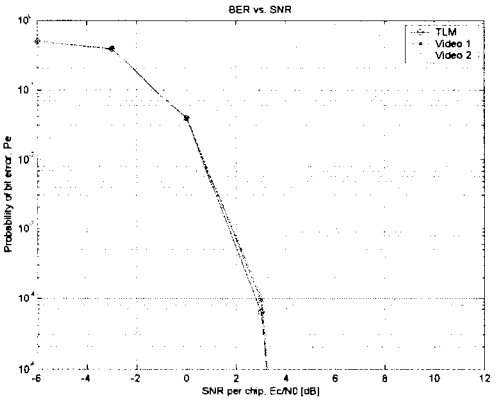


Figure 27. SNR (Ec/N0) vs. Pe

We shall leave out further details involved in simulations because users can verify how blocks each work and perform necessary simulations themselves, mounting or dismounting them.

6. Conclusion

Compared with others, the CDMA system is known to be relatively complex. Most of the hardware complexity is subject to the TR (*Searcher* plus *CTL*) process. As the number of fingers increases, the complexity will soar. In this project, our focus lies on provision of a baseband DS/CDMA simulator which is of the same structure as the real hardware but much easier to understand. For this reason, we restrict the system coverage to a point-to-point communications link. Extension to a network based DS/CDMA model would be our future work.

By far, we have introduced a likely channel model, relevant algorithms in depth and practical software/hardware implementations. The simulation results justify our choice of the parameters used for the system and verify the consistency between the software and hardware implementations. Under the given requirements and assumptions, the DS/CDMA model has proved to work very well tracking out such impairments as timing drift, carrier frequency offset, dc bias and so on. All the algorithms used in the simulator take the hardware implementation into account. For instance, some multiplications like power or loop gains are based on power of 2 so that a binary bit shifter can replace a multiplier in the FPGA. Furthermore, digitally implemented algorithm blocks on a single chip replace their analog counterparts that may have been deployed in places on the board.

Hardware tests have been performed in the LAB only, that is, no field tests have been done yet. The applicability of our system to the field is proven by the simulator only. Therefore, our short-term plan is to test the hardware in various manners, based on the simulation results. We will identify the channel observing the real data transmitted over the real communications link first, verify the suitability of our channel model, remodel it if necessary and then tune up relevant parameters according to simulation results. This work, as mentioned above, may be followed by point-to-multipoint or multipoint-to-multipoint communications applications building up interactive data links.

7. References

1. R.L. Peterson, R.E. Ziemer and D.E. Borth, *Introduction to Spread Spectrum Communications*, Prentice-Hall, Inc., 1995
2. A.J. Viterbi, *Principles of Spread Spectrum Communication, Addison & Wesley*, 1995
3. M.K. Simon, J.K. Omura, R.A. Sholts and B.K. Levitt, *Spread Spectrum Communications Handbook*, McGraw-Hill Inc., 1994
4. Michael Rice, Adam Davis, Christian Bettwieser, *Wideband Channel Model for Aeronautical Telemetry*, IEEE Transactions of Aerospace and Electronic Systems Vol. 40, No. 1, Jan. 2004
5. Edward A. Lee, David Messerschmitt, *Digital Communication 2nd Edition*, Kluwer Academic Publishers, Chapter 16, pp. 725-736
6. Louis Litwin, *Matched Filtering and Timing Recovery in Digital Receivers*, RFDESIGN, pp. 32-48, Sep. 2001
7. John G. Proakis, *Digital Communications 4th Edition*, McGraw-Hill Inc., 2001
8. Samuel C. Yang, *CDMA RF System Engineering*, Artech House Publishers, 1998