Table of Contents

ACOLADE IS-95-A/CDMA Library Overview

1223 Peoples Ave.
Troy, NY 12180
(518)274-7711

1. Introduction

This document provides a brief description of an optional library available for the Advanced COmmunication Link Analysis and Design Environment (ACOLADE), designed to model certain Code-Division Multiple-Access (CDMA) communication techniques. The mnemonic IS-95-A refers to a public standard for wideband spread-spectrum cellular systems published by the Telecommunications Industry Association (TIA)[1]. This document can be ordered from "Global Engineering Documents", 15 Inverness Way East, Englewood, CO, 80112-5704, or call 1-800-854-7179. The models within the library follow the standard closely in those cases where the standard is applicable.

Section 2 provides some general background on CDMA spread spectrum systems. Section 3 discusses the components of forward and reverse IS-95-A communications links. A definition of the parts of the IS-95-A standard that are outside the scope of the library is provided in Section 4. The library models are enumerated and described in Section 5. Section 6 covers useful performance bounds that are provided as part of the library. Finally, some detailed examples are provided in Section 7.

2. Background

Spread-spectrum techniques have been employed in military communication and radar systems for about 50 years. The primary purposes in communications have been to combat the effects of jamming and to effect covertness. In radar systems, the primary purpose has been to provide accurate ranging (delay measurement). Numerous books and papers have been written on spread-spectrum techniques used for these purposes over the years.

Spread-spectrum techniques are also useful in commercial applications, although the motivation to apply such techniques to these applications has developed only fairly recently. The most important use of spread-spectrum techniques in the commercial world is in multi-user communications. Spreading the signal of multiple users with a unique spreading waveform assigned to that user can allow simultaneous access to a shared communication channel. This technique is called Code-Division Multiple-Access (CDMA) and forms the basis of the IS-95-A standard. Many papers have been published on CDMA techniques over the years, but the first textbook devoted to CDMA appeared but recently [2].

The IS-95-A design provides superior multiple-access capabilities, in addition to a number of other desirable attributes for wireless cellular service. These attributes include optimum subscriber station power management, universal frequency reuse, soft handoff, and enablement of the use of optimum receiver structures for time-varying multipath fading channels. We assume that the reader has some familiarity with CDMA principles and the IS-95-A standard, in particular. We provide a basic description of the architecture of the forward (base station to subscriber) and reverse (subscriber to base station) links in the next section.

3. Basic IS-95-A Link Architecture

The IS-95-A standard covers both analog and digital signaling formats, providing specifications for both modes of operation. The analog portion of the specification describes how the system must operate in order to be compatible with the existing Advanced Mobile Phone Service (AMPS) equipment. The library described here models only the digital mode of operation.

The digital IS-95-A standard employs Direct-Sequence (DS) spectrum spreading by multiplying the user's narrowband waveform by a wideband signal. This wideband signal is generated by spreading "codes" consisting of sequences of 64 "chips", associated with each symbol interval. The entire sequence of chips is used to modulate the carrier during each symbol period, resulting in a widened or "spread" spectrum. The underlying information sequence is mapped into the chip sequence in different ways in the forward vis-à-vis the reverse channel. In both cases, the characteristics of the spread-spectrum signal provide some important advantages:

1) The spreading sequences are chosen so that multiple subscribers can access the channel at the same time over the same frequency band. The spreading sequences are chosen so that signals corresponding to other subscribers are exactly or substantially uncorrelated with a given subscriber, after appropriate processing at the receiver.

2) The autocorrelation functions associated with wideband spread-spectrum signals are much narrower than those associated with the underlying information-bearing signal, thus enabling much finer delay resolution for multipath signals. Since the sets of spreading waveforms employed are independent of the data, accurate gleaning and combining of such multiple signals is possible at the receiver.

3) CDMA systems can operate with a much lower carrier to interference C/I ratio, allowing significant immunity to various types of interfering signals when operating at a given carrier power.

When considered purely as a multiple access technique over Additive White Gaussian Noise (AWGN) channels, CDMA will actually support fewer users within the same bandwidth as Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA) techniques. However, certain aspects of the mobile cellular environment allow CDMA to provide vastly increased efficiency over the other two techniques:

1) Universal frequency reuse is possible, since users occupy a common spectrum.

2) Transmitter energy scattered over multiple paths can be consumed more efficiently at the receiver. The spread-spectrum waveform enables the constructive combining of multipath signal components, as described above.

3) Soft handoff between cells is possible, allowing more efficient use of base station power. This also allows continuous coverage, with the potential elimination of dropped calls between cells.

4) Since interference of all types appears "noise-like" at the base station receiver, it is possible to perform accurate interference power estimation and closed-loop subscriber power control. This allows subscribers to transmit at minimal power levels, minimizing their interference power on other subscribers as seen at the base station. This effectively eliminates the common "near-far" problem traditionally associated with DS spread-spectrum systems.

In addition to CDMA spread-spectrum modulation techniques, the IS-95-A standard employs other sophisticated signal processing techniques. Powerful convolutional coding techniques are employed, in order to embed controlled redundancy into the transmitted symbol stream. This redundancy is exploited at the receiver to allow accurate data bit estimates to be made, based on observations of very noisy received symbols. Channel symbol interleaving is also employed, allowing the occurrence of low-reliability symbols to be randomized at the decoder input. This allows the effective use of the large class of random error-control codes.

For a further elaboration of these points, and a history of the development of IS-95-A, see [3].

In the remainder of this section, we describe the principles behind the digital IS-95-A system design by reviewing block diagrams of both the forward and reverse links. The TIA IS-95-A air interface specification prescribes the format of the signal that is to be transmitted. Thus, it focuses

on the transmitting side of the equipment. In this area, the IS-95-A library elements follow the IS-95-A specification quite closely. At the receiving end, the standard specifies only certain broad guidelines for receiver design. Receiver design principles for CDMA systems can be found in [2]. The design of the receiver models in the library draws heavily from this reference and on ICUCOM's past experience in the modeling of general spread-spectrum systems.

3.1 Forward (coherent) link

A block diagram of the base station to subscriber link is shown in Figure 1. This link is called the Forward CDMA Channel. This link operates coherently, enabled by the transmission of a pilot signal. We will discuss the forward traffic channel only. We will not discuss the sync and paging channels here. The implementation of these channels is straightforward. Within the diagrams in this document, we employ a small circle around the operator within summation or multiplication icons to indicate that the operation is performed over a binary field of {0,1}.

The Forward link is capable of accepting information bits at the rate of 0.8 Kbps, 2.0 Kbps, 4.0 Kbps and 8.6 Kbps. Channel symbols are transmitted in frames, with each frame containing a number of information bits and a convolutional encoder tail sequence to drive the convolutional encoder into a known state at the end of each frame. Additionally, the frame incorporates a frame quality indicator sequence for the two highest data rates. The frame quality indicator sequence is nothing but a set of parity check digits that detect errors in the frame. It is also used to establish the incoming data rate at the subscriber station.

After the possible addition of the check digits and the addition of the tail bits, each frame is convolutionally encoded with a rate = 1/2 constraint length 9 convolutional encoder with tap connections = {753, 561} (Octal).

Depending on the data rate, the encoder output sequence is repetition-encoded by a factor of 1 (no encoding), 2, 4 or 8. This brings the final baseband symbol rate up to a constant 19.2 Kbps.

A block interleaver of special design is employed to interleave and permute the order of the baseband symbols before transmission.

After interleaving, the baseband sequence is scrambled by a "long code generator", the decimated output of which is modulo-2 added to the baseband symbol stream. The long code output is also used to randomize the placement of a power control bit within the baseband symbol stream.

The power control bit is generated at an average rate of 800 bps in response to changing received SNR conditions at the Base station. The power control bit commands the subscriber equipment to either raise or lower its power output to achieve adequate but not wasteful received power levels. This is a key element in both the superior capacity and power efficiency attributes of CDMA cellular systems.

After multiplexing, the baseband symbol stream is spread by multiplication with a Walsh sequence of length 64, thus creating a baseband "chip" sequence at a rate of 1.2288 Mcps. There are 64 orthogonal Walsh sequences of length 64, certain of which are assigned to different users of the channel. All user's transmissions occur synchronously from the base station, so these transmissions are also synchronized at any individual subscriber's receiver. The use of a set of orthogonal sequences thus allows perfect rejection of other-user interference associated with any given transmission path.

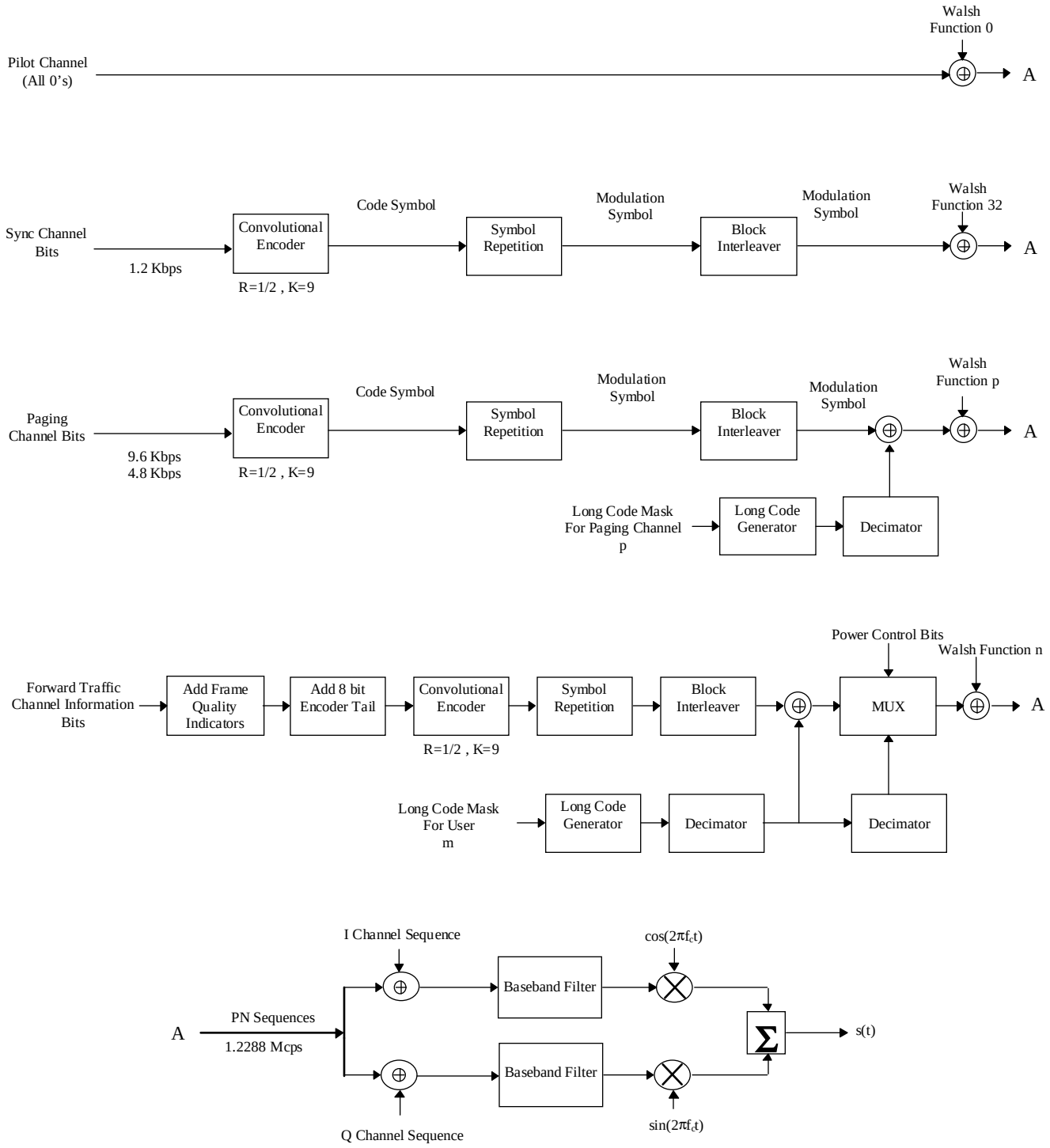

Figure 1. Forward CDMA Channels.

The baseband sequence is finally BPSK-modulated on both the inphase (I) and quadrature (Q) channels. The same baseband sequence is duplicated on both channels, then spread with different pilot sequences on the I and Q channels. This technique allows independent despreading and amplitude measurement of both channels. This, in turn, allows the embedded pilot sequence to be employed for channel sounding purposes to determine the amplitudes and phases of various multipath components received at the subscriber station.

After spreading by the pilot sequences, the chip sequences are passed through identical baseband filters to produce the baseband I/Q modulating signals. The filters are specified in the IS-95-A standard on the basis of a spectral mask and allowed mean-square deviation from a discrete-time impulse response. The impulse response is specified at 4 samples per chip with length 48. The baseband filters in the Library are of this rate and length. Figure 1 indicates that each channel is applied to an individual modulator. In this case, the various signals, s(t), would be added together to form a composite multi-user signal. In practice, the signals might be combined algebraically at baseband before being input to the modulator.

At the receiving end, the IS-95-A specification is silent on the specific architecture of the subscriber receiver. It does, however, refer to the IS-97 “Recommended Minimum Receiver Performance Standards for Base Stations Supporting Wideband Spread Spectrum Cellular Mobile Stations”, which supplies general minimum operational performance specifications. The generic receiver architecture commonly used in IS-95-A systems is known as a “Rake” receiver. This is a multipath reception technique that attempts to perform what might be deemed an obvious procedure for combining multiple received signals. The Rake technique attempts to time-align the various delayed paths and add them up again. The paths must be weighted in an optimum fashion, which depends heavily on the long-term fading statistics of the various received paths. If the Rake receiver additionally attempts to extract and exploit information concerning the phase of the received paths, the Rake receiver is deemed coherent. Due to the presence of a pilot sequence in the forward IS-95-A transmitted signal, subscriber receivers usually employ a coherent technique. A description of Rake receivers can be found in [4].

In the context of cellular mobile radio, Rake receivers have two functions. The first is to periodically perform measurements on the time-varying channel to determine the multipath structure (delays, amplitudes and possibly phases) and track its changes over time. This is often called channel “sounding”. Since the multipath structure of mobile radio channels is usually changing more rapidly than “quasi-static” multipath channels (like HF and TROPO, for example), this aspect of the receiver’s operation becomes critical. For a given multipath structure, the Rake receiver must then act by combining a selected number of received paths in some optimum or approximately optimum fashion. The number of paths to be combined is often fixed as part of the receiver specification, so the receiver usually attempts the recombining of the N strongest received paths. N typically ranges from 3-6 in most Rake receivers.

The Rake receiver may be implemented with a variety of structures, each with its own set of advantages and disadvantages. The Library provides models for several different implementations. Classically, Rake receivers have been implemented with a number of discrete receiving elements, called “fingers” of the Rake. This is most appropriate for implementation with monolithic components, used to actually implement the several fingers. A model of this approach is shown in Figure 2, where a parallel Rake finger combiner is illustrated, along with the internal details of one of its fingers. On the other hand, modern DSP technology has made it possible to perform more and more baseband functions within programmable chips or ASICs. In this case, individual paths may be processed serially. With this type of implementation, certain DSP techniques make it possible to perform some of the receiver functions very efficiently. A model for serial implementation is shown in Figure 3, where a serial weighting and combining circuit is illustrated. These and other techniques may be mixed and matched in any given receiver design. We will discuss some of these options in more detail in the examples.

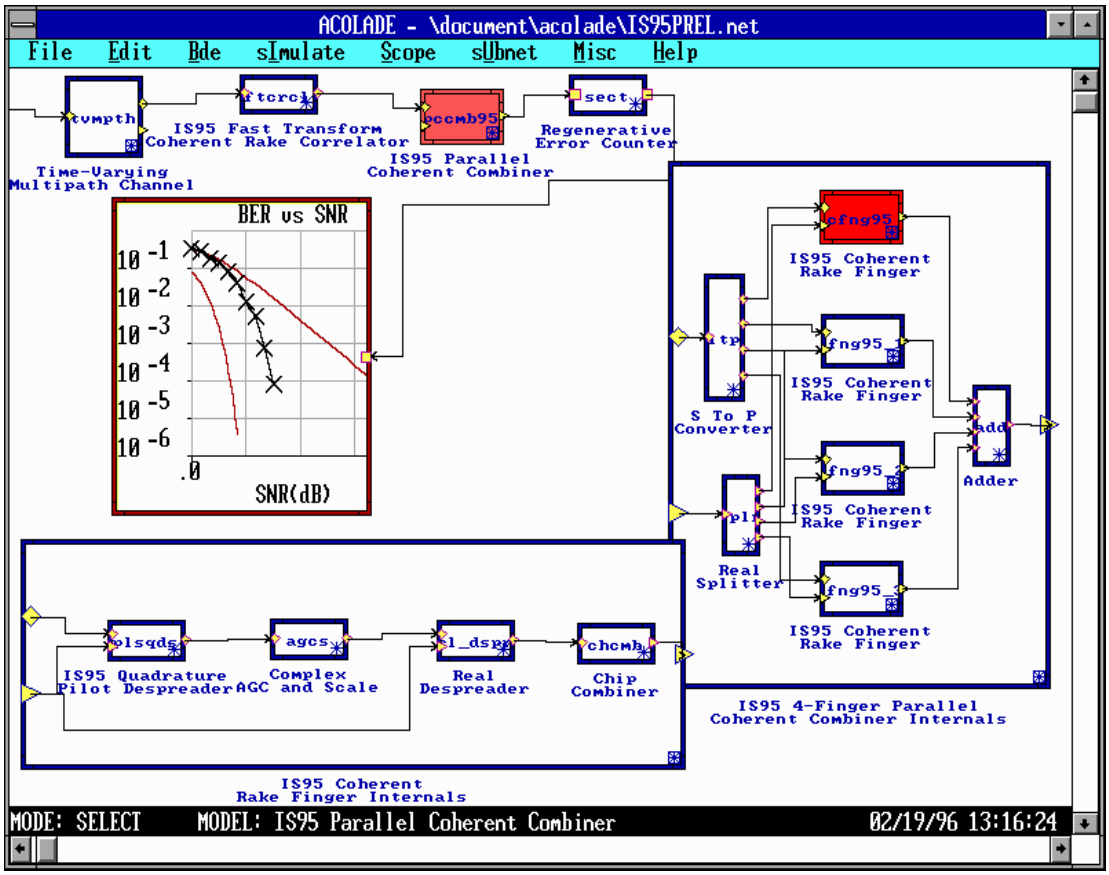

Figure 2. Parallel Rake Finger Combiner.

3.2 Reverse (noncoherent) link

The IS-95-A subscriber-to-base station link employs a noncoherent transmission technique. Generally, the subscriber equipment is power-limited, thus transmission of a separate pilot sequence to enable coherent operation is not attractive. The determination of multipath structure and subsequent path combining is still possible, however, with a bit more sophisticated signal processing. Since the base station equipment can be larger (and generally more costly), the implementation of more complex processing is usually not problematic.

A block diagram of the subscriber-to-base link is shown in Figure 4. This link is called the Reverse Traffic Channel link. Many of the components will be recognized from the forward link, but some are used in different ways.

Information rates and the channel signaling rate remain the same, but some intermediate rates differ based on different processing, modulation and encoding techniques. The frame quality indicator is still employed for the two higher data rates, as well as the frame quality indicator bits.

In this case, a rate = 1/3, constraint length 9 convolutional code is employed, which provides a greater output symbol rate than the forward channel encoder.

The code symbols are again repeated for all data rates except the highest and input to the same type of block interleaver.

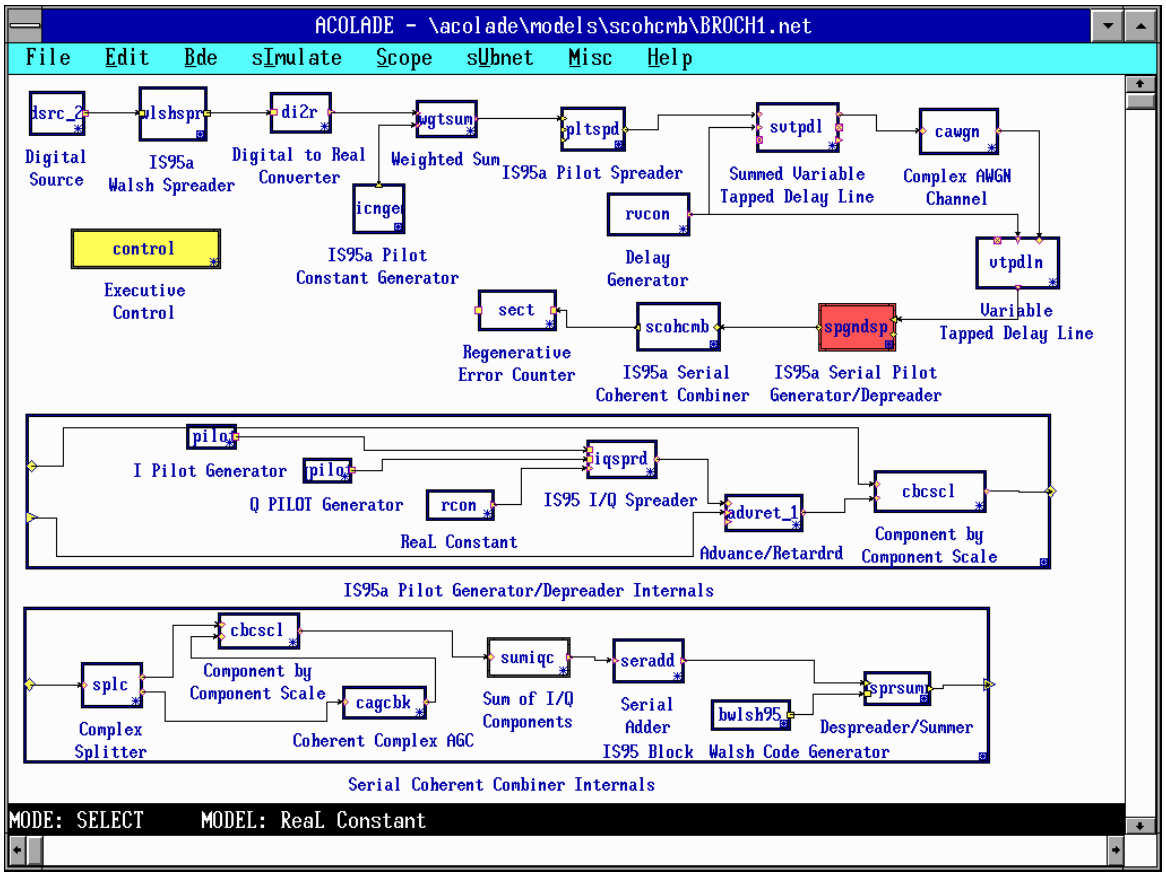

Figure 3. Serial Implementation of Rake Finger Combiner.

In the reverse link, the Walsh codes are employed at this point to provide a noncoherent orthogonal sequence modulation technique. The Walsh modulator accepts the six bits needed to create an index for the set of 64 64-ary orthogonal sequences which may be transmitted. This produces a Walsh chip rate of 307.2 kcps.

A “data burst randomizer” is employed to delete all but one copy of each code symbol from the symbol stream that is eventually transmitted. The randomizer operates by dividing each frame into 16 groups of 6 64-ary modulation symbols and determining which of the 16 groups is to be transmitted. The transmitter is to be gated off during periods of no transmission. The long code generator spreads the randomizer output with four PN chips per Walsh chip, giving a final output chip rate of 1.2288 Mcps. The baseband modulating sampled-data waveforms are created in the same fashion as in the forward channel, with the exception of a one-half chip delay imposed on the quadrature channel. Thus, the modulation on the reverse channel is offset QPSK.

The receiver for the reverse channel is still usually of the Rake type, although the processing is somewhat different and more complex, due to the noncoherent modulation. The demodulation process involves noncoherent correlation between the received complex Walsh-modulated sequence and 64 candidate Walsh sequences to determine that most likely transmitted. This process can be efficiently implemented in DSP hardware, however, in the form of a Walsh transform, which requires only additions and subtractions. We will provide further elaboration in the examples.

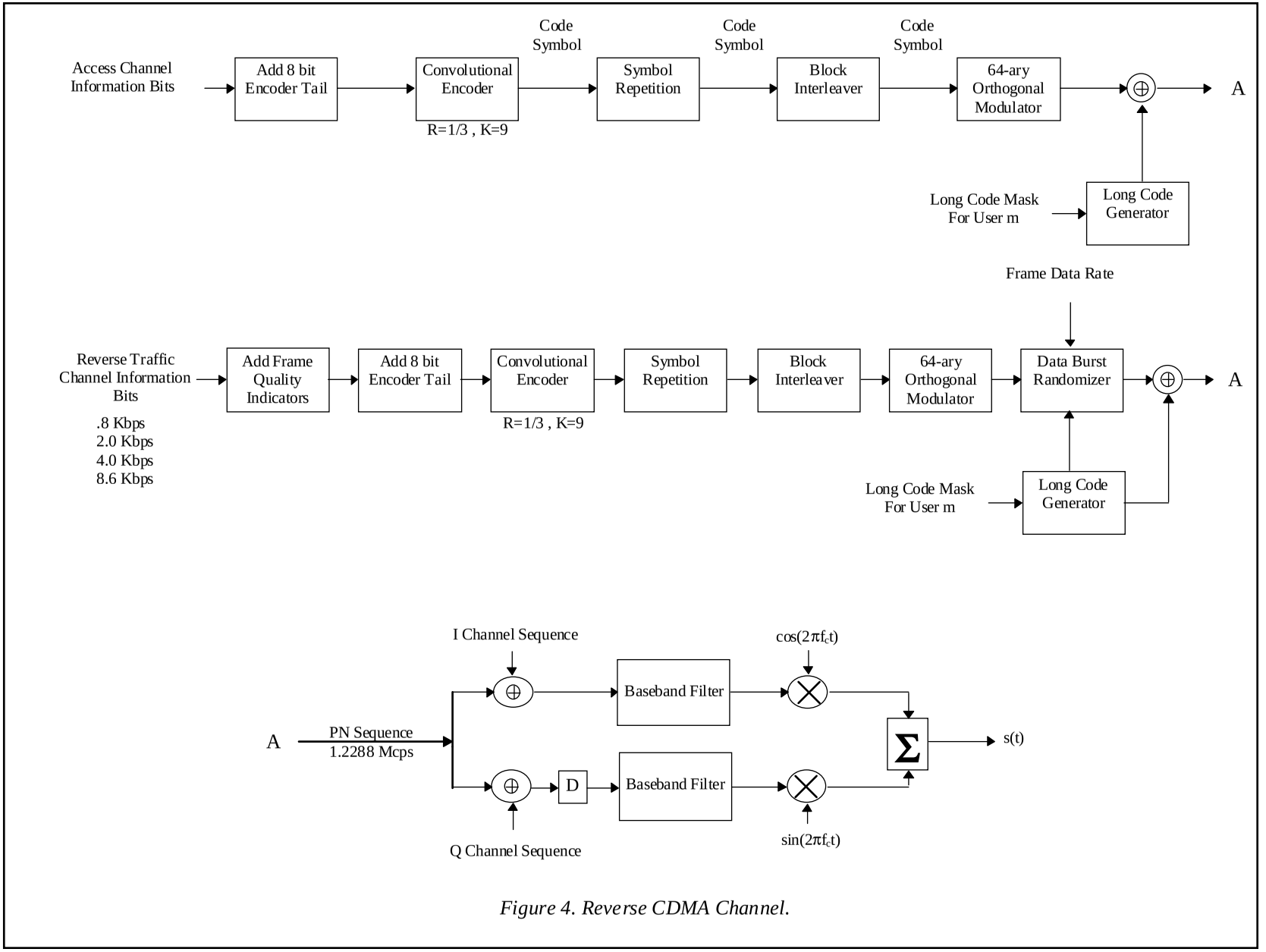

Figure 4. Reverse CDMA Channel.

4. Portions of the Standard Not Covered

The Library does not provide the capability to model the cellular network protocol procedures associated with the IS-95-A standard, which are outside its scope. The Library currently does not provide the capability to model secondary and signaling traffic within the IS-95-A system. The Library is designed to model the components and subsystems necessary to implement certain parts of the physical layer of the IS-95-A specification only, primarily focusing on the signal processing associated with forward and reverse traffic channels.

5. Model List and Description

This section contains a listing of all models in the ACOLADE IS-95-A CDMA Library and a short description of each.

5.1 IS-95-A General Modules and Subnets (is95gen)

This class contains general models that are used in both forward and reverse channels.

The following is a list of the General models in the ACOLADE IS-95-A CDMA Library:

- IS-95-A 9600 bps Add 8-Bit encoder Tail (a8bt9695)
- IS-95-A 4800 bps Add 8-Bit encoder Tail (a8bt4895)
- IS-95-A 2400 bps Add 8-Bit encoder Tail (a8bt2495)
- IS-95-A 1200 bps Add 8-Bit encoder Tail (a8bt1295)
- IS-95-A BLocK REPeater (blkrep95)
- IS-95-A Block WaLSH code generator (bwlsh95)
- IS-95-A CenTeR of GraVity (ctrgv195)
- IS-95-A COG SeQuence generator 1 (cogsq195)
- IS-95-A CHanneL SPreaDer (chlspd95)
- IS-95-A I/Q CoMplex SPreader (iqcmsp95)
- IS-95-A Component By Component SCaLe (cbcscl95)
- IS-95-A 9600 bps CRC Decoder (crcd9695)
- IS-95-A 4800 bps CRC Decoder (crcd4895)
- IS-95-A 9600 bps Frame Quality Indicator (fqi9695)
- IS-95-A 4800 bps Frame Quality Indicator (fqi4895)
- IS-95-A LoNG CoDe generator 1 (lngcd95)
- IS-95-A LoNG CoDe generator 2 (lngcd295)
- IS-95-A Pilot CoNstant GENerator (pcngen95)
- IS-95-A I PILOT sequence generator (ipilot95)
- IS-95-A Q PILOT sequence generator (qpilot95)
- IS-95-A Serial PiLot DeSPreader (spldsp95)
- IS-95-A Serial Pilot Generator/DeSPreader (spgdsp95)
- IS-95-A Serial PRoGrammable Multipath Rake (sprgmr95)
- IS-95-A WaLSH code generator (wlsh95)
- IS-95-A WaLSH SPreader (wlshsp95)

IS-95-A 9600 bps Add 8-Bit encoder Tail (a8bt9695)

This Subnet adds an 8-bit all-zeroes tail sequence to drive a constraint length k = 9 binary convolutional encoder into the all-zeroes state after a block of 184 input bits.

IS-95-A 4800 bps Add 8-Bit encoder Tail (a8bt4895)

This Subnet adds an 8-bit all-zeroes tail sequence to drive a constraint length k = 9 binary convolutional encoder into the all-zeroes state after a block of 88 input bits.

IS-95-A 2400 bps Add 8-Bit encoder Tail (a8bt2495)

This Subnet adds an 8-bit all-zeroes tail sequence to drive a constraint length k = 9 binary convolutional encoder into the all-zeroes state after a block of 40 input bits.

IS-95-A 1200 bps Add 8-Bit encoder Tail (a8bt1295)

This Subnet adds an 8-bit all-zeroes tail sequence to drive a constraint length k = 9 binary convolutional encoder into the all-zeroes state after a block of 16 input bits. For each iteration of this Subnet, 16 bits are consumed and 24 bits are produced.

IS-95-A BLocK REPeater (blkrep95)

This module repeats an input block of data values a number of times. This Module may process a fixed number of input blocks per invocation, or may simply process all blocks that are available at the input.

IS-95-A Block WaLSH code generator (bwlsh95)

This model is a Walsh sequence generator for IS-95-A CDMA systems. It allows the user to specify 1 of 64 orthogonal Walsh-Hadamard sequences of length 64 with an input parameter. For a description of the Walsh-Hadamard sequence, see the TIA/EIA IS-95-A specification. The user may specify the number of Walsh sequences to be emitted by this model per invocation.

IS-95-A CenTeR of GraVity (ctrgv195)

This model is a classical center of gravity circuit used in coherent Rake systems. For background see, for example: Price, R., and Green, P. E., Jr. (1958). “A Communication Technique for Multipath Channels”, PROC. IRE 46, 555-570.

IS-95-A COG SeQuence generator 1 (cogsq195)

This Module generates a positive-going ramp of a certain length for use as a correlation sequence in a center of gravity circuit. The ramp is centered at 0.0. If, for example, it has an odd length, the center value is 0.0. If the length is even, the middle two samples straddle 0.0.

IS-95-A CHanneL SPreaDer (chlspd95)

This is the IS-95-A Channel spreader module. It multiplies a real chip sequence with a PN pilot sequence. The spreading PN pilot sequence is DIGITAL data (0.0's and 1.0's), so it is mapped to 1.0 and -1.0, respectively before used as a multiplier for the input real sequence. The output of this module is a real sequence. The actual multiplication process is implemented as a conditional (if the PN chip is 1) sign reversal of the real input chip values.

IS-95-A I/Q CoMplex SPreader (iqcmsp95)

This is a complex spreader for spread-spectrum systems. It multiplies individual inphase (I) and quadrature (Q) components by the spreading chip sequences on ports 0 and 1, respectively.

IS-95-A Component By Component SCaLe (cbcscl95)

This Module scales the Inphase and Quadrature components of an incoming serial-multiplexed signal stream by corresponding Inphase and Quadrature weights. Each input block is of length "Num_Signals." Each complex element of the block is a sample associated with one of "Num_Signals" time-multiplexed signal streams. The weighting samples are input from port 1, if it is connected. The user is allowed to specify how often the weighting port is read, if it is connected.

IS-95-A 9600 bps CRC Decoder (crcd9695)

This model is the 9600 bps CRC error correction code decoder. It uses the code generated by the Frame Quality Indicator model to determine if a single error occurred during the transmission. This model is capable of correcting one single error per frame. The generator polynomial used is identical to the 9600 bps Frame Quality Indicator model, which is:

$$g(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^4 + x + 1$$

This model will remove the 12 CRC bits added by the Frame Quality Indicator model. The model will output 172 data bits.

IS-95-A 4800 bps CRC Decoder (crcd4895)

This model is the 4800 bps CRC error correction code decoder. It uses the code generated by the Frame Quality Indicator model to determine if a single error occured during the transmission. This model is capable of correcting one single error per frame. The generator polynomial used is identical to the 4800 bps Frame Quality Indicator model, which is:

$$g(x) = x^8 + x^7 + x^4 + x^3 + x + 1$$

This model will remove the 8 CRC bits added by the Frame Quality Indicator. The model will output 80 data bits.

IS-95-A 9600 bps Frame Quality Indicator (fqi9695)

This model is the 9600 bps Frame Quality Indicator. It is a CRC encoder which uses the following generator polynomial:

$$g(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^4 + x + 1$$

This model adds the tail bits to the end of the information bits. The matching CRC decoder will determine if an error occured during the transmission by recalculating these bits. 172 information bits are used per frame. 12 CRC bits are added to the end of these information bits.

IS-95-A 4800 bps Frame Quality Indicator (fqi4895)

This model is the 4800 bps Frame Quality Indicator. It is a CRC encoder which uses the following generator polynomial:

$$g(x) = x^8 + x^7 + x^4 + x^3 + x + 1$$

This model will add the tail bits to the end of the information bits. The matching CRC decoder will determine if an error occured during the transmission by recalculating these bits. Eighty information bits are used per frame. Eight CRC bits are added to the end of these information bits.

IS-95-A LoNG CoDe generator 1 (lngcd95)

This is the Long Code Generator module as specified in the IS-95-A standard. The long code is a periodic binary code with period of $((2^{24}) - 1)$ chips. It has the following characteristic polynomial:

$$p(x) = x^{42} + x^{35} + x^{33} + x^{31} + x^{27} + x^{26} + x^{25} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{10} + x^7 + x^6 + x^5 + x^3 + x^2 + x + 1$$

Each PN chip of the long code is generated by the modulo-2 inner product of a 42-bit mask that is fed to the module from the input port number 0. This module is an implementation of the shift register that represents the above polynomial. The initial load of this shift register is specified as a parameter. The mask will be read from port number 0 in blocks of 42 bits and updated after each output bit.

IS-95-A LoNG CoDe generator 2 (lngcd295)

This is the Long Code Generator module as specified in the IS-95-A standard. The long code is a periodic binary code with a period of $((2^{24}) - 1)$ chips. It has the following characteristic polynomial:

$$p(x) = x^{42} + x^{35} + x^{33} + x^{31} + x^{27} + x^{26} + x^{25} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{10} + x^{7} + x^{6} + x^{5} + x^{3} + x^{2} + x^{1} + 1$$

Each PN chip of the long code is generated by the modulo-2 inner product of a 42-bit mask that is specified by the user in the input parameters table and the 42-bit state vector of the sequence generator.

IS-95-A Pilot CoNstant GENerator (pcngen95)

This Subnet is used in coherent IS-95-A CDMA transmitters to generate a constant 1.0 binary chip stream. A constant source is used to generate a number of 1.0's, corresponding to symbols. Each incoming symbol is first repeated 63 times and the resulting constant sequence is output. For each symbol generated, 64 chips are output.

IS-95-A I PILOT sequence generator (ipilot95)

This is an inphase (I) channel pilot sequence generator for IS-95-A systems. The inphase channel spreading sequence is periodic with period 2^{15} chips based on the following polynomial:

$$P(x) = x^{15} + x^{13} + x^{9} + x^{8} + x^{7} + x^{5} + 1$$

IS-95-A Q PILOT sequence generator (qpilot95)

This is a Quadrature (Q) channel pilot sequence generator for IS-95-A systems. The quadrature channel spreading sequence is periodic with period 2^{15} chips based on the following polynomial:

$$P(x) = x^{15} + x^{12} + x^{11} + x^{10} + x^{6} + x^{5} + x^{4} + x^{3} + 1$$

IS-95-A Serial PiLot DeSPreader (spldsp95)

This Subnet is a component by component complex pilot despreader for serial-multiplexed received IS-95-A signals. This Subnet is fed by the output of Num_Fingers fingers associated with a Rake receiver. The received individual chips from these fingers are assumed to be serial-multiplexed into contiguous blocks of length Num_Fingers. The model employs a component-by-component scale Module to perform the despreading of the pilot chip sequence against the various multiplexed sequences from the individual fingers. The I/Q components of the received sequence are assumed to have mean {+A, -A}. The pilot sequence components are assumed to be in real {+1.0, -1.0} format.

IS-95-A Serial Pilot Generator/DeSPreader (spgdsp95)

This Subnet is used in IS-95-A CDMA receivers to generate a complex PN sequence and to despread a set of multiplexed received complex sequences with it. The I/Q spreading sequences are generated by separate I/Q pseudonoise generators and then combined and converted to real {+1.0,-1.0} format. This sequence is then delayed or advanced by an advret Module and input to a cbcscl Module to do the actual despreading. The cbcscl Module multiplies the inphase component of the spreading sequence by the inphase component of each of Num_Fingers contiguous time-multiplexed received chip sequences. The same is done for the quadrature components.

IS-95-A Serial PRoGrammable Multipath Rake (sprgmr95)

This Subnet is used in Rake receivers to extract a number of multipath components at specified delays and to weight the multipaths with certain REAL weights. This Subnet employs a programmable tapped delay line model.

IS-95-A WaLSH code generator (wlsh95)

This model is a Walsh sequence generator for IS-95-A CDMA systems. It allows the user to specify 1 of 64 orthogonal Walsh-Hadamard sequences of length 64 with an input parameter. For a description of the Walsh-Hadamard sequence, see the TIA/EIA IS-95-A specification. This document can be ordered from “Global Engineering Documents”, 15 Inverness Way East, Englewood, CO, 80112-5704, or call 1-800-854-7179.

IS-95-A WaLSH SPreader (wlshsp95)

This Subnet is used in coherent IS-95-A CDMA transmitters to spread a binary symbol stream by a 64-ary Walsh function of a given number. Each incoming symbol is first repeated 63 times and the resulting constant sequence is multiplied component-by-component by the specified Walsh sequence. For each symbol input, 64 chips are output.

5.2 Reverse Channel Modules and Subnets (is95rev)

This class contains models used in the reverse channel.

The following is a list of the Reverse Channel models in the ACOLADE IS-95-A CDMA Library:

- IS-95-A Binary WaLsh Demodulator/Detector (bwldd95)
- IS-95-A Binary WaLsh MoDulator (bwlmd95)
- IS-95-A 9600 bps Reverse channel block DeiNterleaVer (rdnv9695)
- IS-95-A 4800 bps Reverse channel block DeiNterleaVer (rdnv4895)
- IS-95-A 2400 bps Reverse channel block DeiNterleaVer (rdnv2495)
- IS-95-A 1200 bps Reverse channel block DeiNterleaVer (rdnv1295)
- IS-95-A 9600 bps Reverse channel block INterleaVer (rinv9695)
- IS-95-A 4800 bps Reverse channel block INterleaVer (rinv4895)
- IS-95-A 2400 bps Reverse channel block INterleaVer (rinv2495)
- IS-95-A 1200 bps Reverse channel block INterleaVer (rinv1295)
- IS-95-A Coherent WaLsh sequence Demodulator/DeteCtor (cwlddc95)
- IS-95-A 9600 bps Reverse channel ConVolutional Encoder (rcve9695)
- IS-95-A 4800 bps Reverse channel ConVolutional Encoder (rcve4895)
- IS-95-A 2400 bps Reverse channel ConVolutional Encoder (rcve2495)
- IS-95-A 1200 bps Reverse channel ConVolutional Encoder (rcve1295)
- IS-95-A 4800 Data Burst RandomiZer (dbrz4895)
- IS-95-A 2400 bps Data Burst RandomiZer (dbrz2495)
- IS-95-A 1200 bps Data Burst RandomiZer (dbrz1295)
- IS-95-A Fast Binary WaLsh Demodulator/Detector (fbwldd95)
- IS-95-A Fast M-ary WaLsh Demodulator/Detector (fmwldd95)
- IS-95-A Fast NonCoherent Rake Correlator 1 (fncrc195)
- IS-95-A Fast NonCoherent Walsh DeModulator (fncwdm95)
- IS-95-A Fast NonCoherent Walsh Demodulator/Detector (fncwdd95)
- IS-95-A Reverse channel Long CoDe DeSpreader (rlcdds95)
- IS-95-A Reverse channel Long CoDe SPreader (rlcdsp95)
- IS-95-A M-ary WaLsh Demodulator/Detector (mwldd95)
- IS-95-A M-ary WaLsh MoDulator (mwlmd95)
- IS-95-A Noncoherent Metric Generator 1 (ncmg195)
- IS-95-A Noncoherent Metric Generator 2 (ncmg295)
- IS-95-A Noncoherent Metric Generator 3 (ncmg395)
- IS-95-A NonCoherent WaLsh sequence Demodulator/Detector (ncwldd95)
- IS-95-A Reverse channel PiLoT SPreader (rpltsp95)
- IS-95-A Reverse channel TransMiTteR 1 (rxmtr195)
- IS-95-A 9600 bps Reverse channel VITerbi decoder (rvit9695):
- IS-95-A 4800 bps Reverse channel VITerbi decoder (rvit4895)
- IS-95-A 2400 bps Reverse channel VITerbi decoder (rvit2495)
- IS-95-A 1200 bps Reverse channel VITerbi decoder (rvit1295)
- IS-95-A Reverse channel WAveform DeModulator 1 (rwadm195)
- IS-95-A Reverse channel WAveform MoDulator 1 (rwamd195)

IS-95-A Binary WaLsh Demodulator/Detector (bwldd95)

This Subnet correlates each received input sequence against $M=2^N$ Walsh sequences of length M and selects the sequence index with the highest correlation value. This index is transformed into a block of bits of length N by an M-ary to binary converter before outputting from the Subnet. The raw sequence of M correlation values is optionally output from an auxiliary port.

IS-95-A Binary WaLsh MoDulator (bwlmd95)

This Subnet selects from among $M=2^N$ orthogonal Walsh sequences of length M to emit, based on an M-ary symbol. The M-ary symbol is created from each block of N serial bits input to the Subnet. This conversion is performed by a binary to M-ary converter Module.

IS-95-A 9600 bps Reverse channel block DeiNterleaVer (rdnv9695)

This Subnet is the IS-95-A 9600 bps Block deinterleaver. The inputs to this Subnet are normally the despread and combined chip sequences from the fingers of a Rake receiver. This deinterleaver is an array with 18 rows and 32 columns (576 cells) where the code symbols are taken from the Subnet input port and written column-wise to the deinterleaver cells until all cells are filled. The cells are then output row-wise. It reverses the 9600 bps interleaving process which consists of an array of the same size but with the number of rows equal to the number of columns of this deinterleaver. The Subnet processes 576 data samples at each iteration.

IS-95-A 4800 bps Reverse channel block DeiNterleaVer (rdnv4895)

This Module is the IS-95-A 4800 bps Block deinterleaver. The inputs to this Subnet are normally the despread and combined chip sequences from the fingers of a Rake receiver. This deinterleaver is an array with 18 rows and 32 columns (576 cells) where the code symbols are taken from the Subnet input port written column-wise to the deinterleaver cells until all cells are filled. The cells are then output row-wise. It reverses the 4800 bps interleaving process which consists of an array of the same size but with the number of rows equal to the number of columns of this deinterleaver. The Subnet processes 576 data samples at each iteration.

IS-95-A 2400 bps Reverse channel block DeiNterleaVer (rdnv2495)

This Subnet is the IS-95-A 2400 bps Block deinterleaver. The inputs to this Subnet are normally the despread and combined chip sequences from the fingers of a Rake receiver. This deinterleaver is an array with 18 rows and 32 columns (576 cells) where the code symbols are taken from the Subnet input port and written column-wise to the de-interleaver cells until all cells are filled. The cells are then output row-wise. It reverses the 2400 bps interleaving process which consists of an array of the same size but with the number of rows equal to the number of columns of this deinterleaver. The Subnet processes 576 data samples at each iteration.

IS-95-A 1200 bps Reverse channel block DeiNterleaVer (rdnv1295)

This Subnet is the IS-95-A 1200 bps Block deinterleaver. The inputs to this Subnet are normally the despread and combined chip sequences from the fingers of a Rake receiver. This deinterleaver is an array with 18 rows and 32 columns (576 cells) where the code symbols are taken from the Subnet input port and written column-wise to the deinterleaver cells until all cells are filled. The cells are then output row-wise. It reverses the 1200 bps interleaving process which consists of an array of the same size but with the number of rows equal to the number of columns of this deinterleaver. The Subnet processes 576 data samples at each iteration.

IS-95-A 9600 bps Reverse channel block INterleaVer (rinv9695)

This Subnet is the IS-95-A 9600 bps Block interleaver. The inputs to this Subnet are the code symbols on the Reverse Traffic Channel. This interleaver is an array with 32 rows and 18 columns (576 cells). The code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled. The cells are then written to the output row-wise. The Subnet processes 576 data values at each iteration.

IS-95-A 4800 bps Reverse channel block INterleaVer (rinv4895)

This Subnet is the IS-95-A 4800 bps Block interleaver. The inputs to this Subnet are the code symbols on the Reverse Traffic Channel. This interleaver is an array with 32 rows and 18 columns (576 cells). The repeated code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled. The cells are then written to the output row-wise with the following row order:

1,3,2,4,5,7,6,8,9,11,10,12,13,15,14,16,17,19,18,20,21,23,22,24,25,27,26,28,29,31,30,32

IS-95-A 2400 bps Reverse channel block INterleaVer (rinv2495)

This Subnet is the IS-95-A 2400 bps Block interleaver. The inputs to this Subnet are the code symbols on the Reverse Traffic Channel. This interleaver is an array with 32 rows and 18 columns (576 cells). The repeated code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled. The cells are then written to the output row-wise with the following row order:

1,5,2,6,3,7,4,8,9,13,10,14,11,15,12,16,17,21,18,22,19,23,20,24,25,29,26,30,27,31,28,32

IS-95-A 1200 bps Reverse channel block INterleaVer (rinv1295)

This Subnet is the IS-95-A 1200 bps Block interleaver. The inputs to this Subnet are the code symbols on the Reverse Traffic Channel. This interleaver is an array with 32 rows and 18 columns (576 cells). The repeated code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled. The cells are then written to the output row-wise with the following row order:

1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16,17,25,18,26,19,27,20,28,21,29,22,30,23,31,24,32

IS-95-A Coherent WaLsh sequence Demodulator/DeteCtor (cwlddc95)

This subnet is a coherent correlation/maximization demodulator/detector for orthogonal Walsh - modulated signals. This demodulator assumes that one of 64 Walsh sequences of length 64 has been transmitted in each time slot, or block of length 64. The received sequence is correlated against each possible transmitted sequence. The number of the sequence with the maximum value is emitted from port 0. The 64 real correlation values for each block are optionally output from port 1, if it is connected. This data is used to generate decoder soft-decision metrics of various types, when this model is used in coded systems.

IS-95-A 9600 bps Reverse channel ConVolutional Encoder (rcve9695)

This Subnet implements a frame-based convolutional encoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification for the 9600 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 184-bit frame of the input data. An a8bt9695 (IS-95-A 9600 bps Add 8-Bit Tail) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These 192 bits are then encoded by the standard convolutional encoder, producing 576 binary output symbols. This model will consume all available frames at its input.

IS-95-A 4800 bps Reverse channel ConVolutional Encoder (rcve4895)

This Subnet implements a frame-based convolutional encoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification for the 4800 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 92-bit frame of the input data. An a8bt4895 (IS-95-A 4800 bps Add 8-Bit Tail) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These bits are then encoded by the standard convolutional encoder, producing 276 binary output symbols. This model will consume all available frames at its input.

IS-95-A 2400 bps Reverse channel ConVolutional Encoder (rcve2495)

This Subnet implements a frame-based convolutional encoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification for the 2400 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 40-bit frame of the input data. An a8bt2495 (IS-95-A Add 8-Bit Tail-2400 bps) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These 48 bits are then encoded by the standard convolutional encoder, producing 144 binary output symbols. This model will consume all available frames at its input.

IS-95-A 1200 bps Reverse channel ConVolutional Encoder (rcve1295)

This Subnet implements a frame-based convolutional encoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification for the 1200 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 16-bit frame of the input data. An a8bt1295 (IS-95-A 1200 bps Add 8-Bit Tail) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These 24 bits are then encoded by the standard convolutional encoder, producing 72 binary output symbols. This model will consume all available frames at its input.

IS-95-A 4800 bps Data Burst RandomiZer (dbrz4895)

This Module implements the 4800 bps reverse channel data burst randomizer. The module uses a masking pattern of ‘0’s and ‘1’s that randomly masks out the redundant data generated by the code repetition. As specified by the IS-95-A standard, the masking pattern is determined by a block of 14 bits taken from the long code. These 14 bits shall be the last 14 bits of the long code used for spreading in the previous to the last power control group of the previous frame. The masking pattern is provided by the user through input port 1.

IS-95-A 2400 bps Data Burst RandomiZer (dbrz2495)

This Module implements the 2400 bps reverse channel data burst randomizer. The module uses a masking pattern of ‘0’s and ‘1’s that randomly masks out the redundant data generated by the code repetition. As specified by the IS-95-A standard, the masking pattern is determined by a block of 14 bits taken from the long code. These 14 bits shall be the last 14 bits of the long code used for spreading in the previous to the last power control group of the previous frame. The masking pattern is provided by the user through input port 1.

IS-95-A 1200 bps Data Burst RandomiZer (dbrz1295)

This Module implements the 1200 bps reverse channel data burst randomizer. The module uses a masking pattern of ‘0’s and ‘1’s that randomly masks out the redundant data generated by the code repetition. As specified by the IS-95-A standard, the masking pattern is determined by a block of 14 bits taken from the long code. These 14 bits shall be the last 14 bits of the long code used for spreading in the previous to the last power control group of the previous frame. The masking pattern is provided by the user through input port 1.

IS-95-A Fast Binary WaLsh Demodulator/Detector (fbwldd95)

This Subnet correlates each received input sequence against $M=2^N$ Walsh sequences of length M and selects the sequence index with the highest correlation value. This index is transformed into a block of bits of length N by an M-ary to binary converter before outputting from the Subnet. The raw sequence of M correlation values is optionally output from an auxiliary port. This Subnet performs the M correlations against the Walsh sequences through the use of a Fast Walsh Transform.

IS-95-A Fast M-ary WaLsh Demodulator/Detector (fmwldd95)

This Subnet correlates each received input sequence against $M=2^N$ Walsh sequences of length M and selects the sequence index with the highest correlation value. The raw sequence of M correlation values is output from an auxiliary port. This Subnet performs the M correlations against the Walsh sequences through the use of a Fast Walsh Transform.

IS-95-A Fast NonCoherent Rake Correlator 1 (fncrc195)

This Module is a monolithic implementation of a Rake multipath correlator for use with noncoherent Walsh-modulated signals. It employs a Walsh transform technique to perform efficient correlations against a set of Walsh functions.

IS-95-A Fast NonCoherent Walsh DeModulator (fncwdm95)

This Subnet model is a noncoherent demodulator for complex signals employing noncoherent Walsh sequence modulation. The Subnet splits the incoming signal into inphase and quadrature components before performing separate Walsh transforms on each component. The I/Q Walsh transform sequences are then squared to annihilate the effects of the unknown received phase before being added element by element.

IS-95-A Fast NonCoherent Walsh Demodulator/Detector (fncwdd95)

This Subnet model is a noncoherent demodulator/detector for complex signals employing noncoherent Walsh sequence modulation. The Subnet employs an IS-95-A Fast Noncoherent Walsh Sequence Demodulator to perform a noncoherent demodulation of the incoming complex Walsh sequences. It employs a Largest Element in Vector model to map the sequences back into 64-ary digital symbols.

IS-95-A Reverse channel Long CoDe DeSpreader (rlcdds95)

This Subnet is used in noncoherent IS-95-A CDMA transmitters to despread an incoming binary chip stream by the output of a “long code” generator. The long code is a binary (0,1) pseudo-random sequence. Four despread chips are combined by summing, producing one Walsh chip.

IS-95-A Reverse channel Long CoDe SPreader (rlcdsp95)

This Subnet is used in noncoherent IS-95-A CDMA transmitters to spread a binary Walsh chip stream by the output of a “long code” generator. The long code is a binary (0,1) pseudo-random sequence.

IS-95-A M-ary WaLsh Demodulator/Detector (mwldd95)

This Subnet correlates each received input sequence against $M=2^N$ Walsh sequences of length M and selects the sequence index with the highest correlation value. This index is output from the Subnet on port 0. The raw sequence of M correlation values is optionally output from an auxiliary port.

IS-95-A M-ary WaLsh MoDulator (mwlmd95)

This Subnet selects from among $M=2^N$ orthogonal Walsh sequences of length M to emit, based on an M-ary symbol. The M-ary symbol is input from port 0.

IS-95-A Noncoherent Metric Generator 1 (ncmg195)

This Subnet generates the suboptimal soft-decision metric described in equation 4.51 of the book: “CDMA, Principles of Spread-Spectrum Communication”, Andrew J. Viterbi, Addison-Wesley, 1995. It is designed to calculate a simple quasi-optimum non-parametric soft-decision statistic for use with Walsh sequence modulation and noncoherent reception.

IS-95-A Noncoherent Metric Generator 2 (ncmg295)

This Subnet generates a binary hard-decision decoding metric for binary linear-combining decoders used with M-ary signaling. The M-ary symbols are simply detected, converted to binary bit stream and then converted to an antipodal format with the mapping: {0,1}<->{1.0,-1.0}.

IS-95-A Noncoherent Metric Generator 3 (ncmg395)

This Subnet generates a non-parametric soft-decision metric for error-control codes used in conjunction with M-ary signaling.

IS-95-A NonCoherent WaLsh sequence Demodulator/Detector (ncwldd95)

This subnet is a noncoherent correlation/maximization demodulator/detector for orthogonal Walsh - modulated signals. This demodulator assumes that one of 64 Walsh sequences of length 64 has been transmitted in each time slot, or block of length 64. The received sequence is correlated against each possible transmitted sequence. The result of the correlation is then squared to account for sign change of the signal during transmission. The number of the sequence with the maximum value is emitted from port 0. The 64 real correlation values for each block are optionally output from port 1, if it is connected. This data is used to generate decoder soft-decision metrics of various types, when this model is used in coded systems.

IS-95-A Reverse channel PiLoT SPreader (rpltsp95)

This Subnet is used in noncoherent IS-95-A CDMA transmitters to multiply a binary chip stream by Inphase and Quadrature (I/Q) pilot spreading sequences. The I/Q spreading sequences are generated by separate I/Q pseudonoise generators. These sequences are then delayed or advanced by adv/ret Modules and input to an I/Q spreader that multiplies a copy of the incoming chip sequence by each of the I and Q sequences to produce the inphase and quadrature components of the output complex sequence. In the reverse channel, the incoming chip sequence is already data-modulated. The spreading operation overlays data-independent PN codes on the I/Q channels.

IS-95-A Reverse channel TransMiTteR 1 (rxmtr195)

This Subnet implements the IS-95-A reverse channel chip transmitter. It consists of four modules. Two channel spreaders spread the incoming data sequence with specified spreading pilot digital sequences from port 0 or port 1 of this subnet. The spreading sequences are digital with component values of 0 or 1.

IS-95-A 9600 bps Reverse channel VITerbi decoder (rvit9695)

This Subnet implements a Viterbi decoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A 4800 bps Reverse channel VITerbi decoder (rvit4895)

This Subnet implements a Viterbi decoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A 2400 bps Reverse channel VITerbi decoder (rvit2495)

This Subnet implements a Viterbi decoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A 1200 bps Reverse channel VITerbi decoder (rvit1295)

This Subnet implements a Viterbi decoder for the rate 1/3, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A Reverse channel WAveform DeModulator 1 (rwadm195)

This Subnet implements an IS-95-A reverse channel chip demodulator. This model is responsible for correlating real-valued samples from a sampled-data waveform against stored pulses for the purpose of matched-filter chip demodulation.

IS-95-A Reverse channel WAveform MoDulator 1 (rwamd195)

This Subnet implements an IS-95-A reverse channel chip modulator. This model is responsible for employing real-valued chip modulating coefficients to weight pulse-like segments of a sampled-data waveform. These overlapping segments are superposed and combined to form a discrete approximation to a continuous-time modulating waveform.

5.3 Forward Channel Modules and Subnets (is95fwd)

This class contains models used in the Forward Channel.

The following is a list of the Forward Channel models in the ACOLADE IS-95-A CDMA Library:

- IS-95-A 9600 bps Forward channel block DeiNerleaVer (fdnv9695)
- IS-95-A 4800 bps Forward channel block DeiNerleaVer (fdnv4895).
- IS-95-A 2400 bps Forward channel block DeiNterleaVer (fdnv2495)
- IS-95-A 1200 bps Forward channel block DeiNterleaVer (fdnv1295)
- IS-95-A 9600 bps Forward channel block INterleaVer (finv9695)
- IS-95-A 4800 bps Forward channel block INterleaVer (finv4895)
- IS-95-A 2400 bps Forward channel block INterleaVer (finv2495)
- IS-95-A 1200 bps block Forward channel INterleaVer (finv1295)
- IS-95-A Coherent Complex AGC (ccagc95)
- IS-95-A Coherent RaKe Channel Sounder 1 (crkcs195)
- IS-95-A Coherent Rake FiNGer 1 (crfng195)
- IS-95-A Coherent RaKe LooP 1 (crklp195)
- IS-95-A Complex AGC and SCale (cagcsc95)
- IS-95-A 9600 bps Forward channel ConVolutional Encoder (fcve9695)
- IS-95-A 4800 bps Forward channel ConVolutional Encoder (fcve4895)
- IS-95-A 2400 bps Forward channel ConVolutional Encoder (fcve2495)
- IS-95-A 1200 bps Forward channel ConVolutional Encoder (fcve1295)
- IS-95-A Forward channel Long CoDe DeSpreader (flcdds95)
- IS-95-A Forward channel Long CoDe SPreader (flcdsp95)
- IS-95-A Parallel Coherent CoMBiner 1 (pccmb195)
- IS-95-A Forward channel PiLoT SPreader (fpltsp95)
- IS-95-A Serial Coherent CoMBiner 1 (sccmb195)
- IS-95-A Serial Coherent CoMBiner 2 (sccmb295)
- IS-95-A Forward channel TransMitTeR 1 (fxmtr195)
- IS-95-A Uncoded Coherent Combining AWGN BER generator (uccabe95)
- IS-95-A Uncoded Coherent Combining Rician BER generator (uccrbe95)
- IS-95-A 9600 bps Forward channel VITerbi decoder (fvit9695)
- IS-95-A 4800 bps Forward channel VITerbi decoder (fvit4895)
- IS-95-A 2400 bps Forward channel VITerbi decoder (fvit2495)
- IS-95-A 1200 bps Forward channel VITerbi decoder (fvit1295):
- IS-95-A Forward channel WAveform DeModulator 1 (fwadm195)
- IS-95-A Forward channel WAveform MoDulator 1 (fwamd195)

IS-95-A 9600 bps Forward channel block DeiNterleaVer (fdnv9695)

This module is the IS-95-A 9600 bps Block deinterleaver. This deinterleaver is an array with 16 rows and 24 columns (384 cells) where the code symbols are taken from the Subnet input port and written column-wise to deinterleaver cells until all cells are filled. It reverses the 9600 bps interleaving process which consists of an array of the same size. The Subnet processes 384 data samples at each iteration.

IS-95-A 4800 bps Forward channel block DeiNterleaVer (fdnv4895)

This module is the IS-95-A 4800 bps Block deinterleaver. This deinterleaver is an array with 16 rows and 24 columns (384 cells) where the code symbols are taken from the Subnet input port and written column-wise to the deinterleaver cells until all cells are filled. It reverses the 4800 bps interleaving process which consists of an array of the same size. The Subnet processes 384 data samples at each iteration.

IS-95-A 2400 bps Forward channel block DeiNterleaVer (fdnv2495)

This module is the IS-95-A 2400 bps Block deinterleaver. This deinterleaver is an array with 16 rows and 24 columns (384 cells) where the code symbols are taken from the Subnet input port and written column-wise to the deinterleaver cells until all cells are filled. It reverses the 2400 bps interleaving process which consists of an array of the same size. The Subnet processes 384 data samples at each iteration.

IS-95-A 1200 bps Forward channel block DeiNterleaVer (fdnv1295)

This module is the IS-95-A 1200 bps Block deinterleaver. This deinterleaver is an array with 16 rows and 24 columns (384 cells) where the code symbols are taken from the Subnet input port and written column-wise to the deinterleaver cells until all cells are filled. It reverses the 1200 bps process which consists of an array of the same size.

IS-95-A 9600 bps Forward channel block INterleaVer (finv9695)

This Module is the IS-95-A Forward channel 9600 Block interleaver. The inputs to this model are the code symbols on the Forward Traffic channel. This interleaver is an array with 24 rows and 16 columns (384 cells). The code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled.

IS-95-A 4800 bps Forward channel block INterleaVer (finv4895)

This Module is the IS-95-A Forward channel 4800 bps Block interleaver. The inputs to this model are the code symbols on the Forward Traffic channel. This interleaver is an array with 24 rows and 16 columns (384 cells). The code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled.

IS-95-A 2400 bps Forward channel block INterleaVer (finv2495)

This Module is the IS-95-A Forward channel 2400 Block interleaver. The inputs to this model are the code symbols on the Forward Traffic channel. This interleaver is an array with 24 rows and 16 columns (384 cells).The code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled.

IS-95-A 1200 bps Forward channel block INterleaVer (finv1295)

This Module is the IS-95-A Forward channel 1200 bps Block interleaver. The inputs to this model are the code symbols on the Forward Traffic channel. This interleaver is an array with 24 rows and 16 columns (384 cells).The code symbols are taken from the Subnet input port and written column-wise to the interleaver cells until all cells are filled.

IS-95-A Coherent Complex AGC (ccagc95)

This is an AGC model for a bank of coherent signals that are input serially. There are Num_Signals whose samples are assumed to be time-multiplexed into input blocks. Num_Blocks of these blocks are processed per invocation of this Module. The long-term average of the inphase component and the quadrature component of each signal is calculated. These estimates are output on the output port. At the beginning of the simulation, estimates based on partial sums are output until Integration_Length samples are collected from each signal. This Module is optimized for block processing with large Num_Signals and/or Num_Blocks. The time between outputs of new estimates is under user control.

IS-95-A Coherent RaKe Channel Sounder 1 (crkcs195)

This Subnet is a Rake channel sounder for use with coherent systems employing a pilot sequence. It operates by correlating a pilot sequence against the incoming chip demodulator output samples. The pilot sequence is shifted in time with respect to the demodulator sequence before correlating. Shift_Window*2+1 shift-and-correlate operations are performed, starting with -Shift_Window relative offset and ending at Shift_Window relative offset. A complex/complex-conjugate correlation is performed and Shift_Window*2+1 magnitudes are output. This model consumes as many blocks of complex data of length Correlation_Length as present at its input, producing Shift_Window*2+1 outputs corresponding to each block. The timing adjustment port (port 3) is read once per input block. The timing reset port (port 2), if connected, is read once per input block.

IS-95-A Coherent Rake FiNGer 1 (crfng195)

This Subnet is a Rake “finger” that is designed to receive one multipath signal out of several to be processed within a coherent Rake multipath receiver. This Subnet model is often employed with several identical models in a parallel configuration. The outputs of the fingers are combined in some fashion to form a composite output signal. The output signal is subsequently detected and/or passed on to a decoder or some other processing block.

IS-95-A Coherent RaKe LooP 1 (crklp195)

This Subnet is an estimator/tracker for a set of spread-spectrum signals that are received under multipath conditions. The signals are assumed to be received with various delays, with bounded delay spread. The delays are assumed to be time-varying, but with variation that is slow enough to allow the weighted mean of the delays to be estimated and tracked over time. A feedback-loop arrangement is employed for this purpose. Since this architecture was first commonly used in conjunction with multipath receivers employing the “Rake” technique, it is called a Rake loop. This particular implementation of the Rake loop uses an unmodulated “pilot” to facilitate the separation of the individual delayed paths. Other implementations employ a training sequence that is embedded into the signal stream from time to time to allow the delay estimation to be performed periodically.

IS-95-A Complex AGC and SCale (cagcsc95)

This subnet is designed to perform an automatic gain control and scale operation for complex sequences. This model is normally employed to process outputs of an IS-95-A pilot despreader in the forward channel of the IS-95-A system. The output of this Subnet would then be passed on to a Walsh despreader/chip combiner for coherent chip combining.

IS-95-A 9600 bps Forward channel ConVolutional Encoder (fcve9695)

This Subnet implements a frame-based convolutional encoder for the rate 1/2, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification for the 9600 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 184-bit frame of the input data. An a8bt9695 (IS-95-A 9600 bps Add 8-Bit Tail) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These 192 bits are then encoded by the standard convolutional encoder, producing 384 binary output symbols.

IS-95-A 4800 bps Forward channel ConVolutional Encoder (fcve4895)

This Subnet implements a frame-based convolutional encoder for the rate 1/2, constraint length 9 convolutional code employed in the reverse noncoherent link of the IS-95-A system specification for the 4800 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 92-bit frame of the input data. An a8bt4895 (IS-95-A 4800 bps Add 8-Bit Tail) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These bits are then encoded by the standard convolutional encoder, producing 184 binary output symbols.

IS-95-A 2400 bps Forward channel ConVolutional Encoder (fcve2495)

This Subnet implements a frame-based convolutional encoder for the rate 1/2, constraint length 9 convolutional code employed in the forward coherent link of the IS-95-A system specification for the 2400 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 40-bit frame of the input data. An a8bt2495 (IS-95-A 2400 bps Add 8-Bit Tail) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These 48 bits are then encoded by the standard convolutional encoder, producing 96 binary output symbols.

IS-95-A 1200 bps Forward channel ConVolutional Encoder (fcve1295)

This Subnet implements a frame-based convolutional encoder for the rate 1/2, constraint length 9 convolutional code employed in the forward coherent link of the IS-95-A system specification for the 1200 bps rate. It operates the Convolutional Encoder Module in a frame-based mode, encoding each 16-bit frame of the input data. An a8bt1295 (IS-95-A 1200 bps Add 8-Bit Tail bps) Subnet is used to add eight zero bits to the end of the data frame to drive the encoder into a known state, thereby terminating the trellis. These 24 bits are then encoded by the standard convolutional encoder, producing 48 output binary symbols.

IS-95-A Forward channel Long CoDe DeSpreader (flcdds95)

This Subnet is used in coherent IS-95-A CDMA transmitters to despread an incoming binary chip stream by the output of a “long code” generator. The long code is a binary (0,1) pseudo-random sequence.

IS-95-A Forward channel Long CoDe SPreader (flcdsp95)

This Subnet is used in coherent IS-95-A CDMA transmitters to spread a binary Walsh chip stream by the output of a “long code” generator. The long code is a binary (0,1) pseudo-random sequence.

IS-95-A Parallel Coherent CoMBiner 1 (pccmb195)

This Subnet coherently combines each of the Num_Fingers received multipath signals derived from its internal Rake fingers. Each Rake finger is responsible for AGC and phase-alignment of its own chip sequence. This particular model uses four fingers, but can be used as a template for different numbers of fingers.

IS-95-A Forward channel PiLoT SPreader (fpltsp95)

This Subnet is used in coherent IS-95-A CDMA transmitters to spread a binary chip stream by Inphase and Quadrature (I/Q) pilot spreading sequences. The I/Q spreading sequences are generated by separate I/Q pseudonoise generators. These sequences are then delayed or advanced by adv/ret Modules and input to an I/Q spreader that multiplies a copy of the incoming chip sequence by each of the I and Q sequences to produce the inphase and quadrature components of the output complex sequence. Each time this Subnet runs, it inputs 1 symbol (64 chips) from the data port and outputs the same number of PN-modulated chips. It will run until all full 64-chip blocks available at the input are consumed.

IS-95-A Serial Coherent CoMBiner 1 (sccmb195)

This Subnet coherently combines each of the Num_Fingers received multipath signals at its input. These Num_Fingers signals are assumed to be multiplexed into blocks before being input to this model. This model is employed in IS-95-A forward links to optimally combine time-delayed, amplitude-faded and phase-shifted multipath signals.

IS-95-A Serial Coherent CoMBiner 2 (sccmb295)

This Subnet coherently combines certain of the received multipath signals which are selected from the Num_Search_Positions received multipath signals appearing in serial-multiplexed format at its input. Num_Search_Positions represents the full width of the Rake time-delay window and may be up to several hundred chips or sub-multiples of chips. The chip data streams entering this Subnet are assumed to have already been despread by the IS-95-A complex pilot sequence prior to entering.

IS-95-A Forward channel TransMitTeR 1 (fxmtr195)

This Subnet implements the IS-95-A forward channel chip transmitter. It consists of four modules. Two channel spreaders spread the incoming data sequence with specified spreading pilot digital sequences from port 0 or port 1 of this subnet. The spreading sequences are digital of component values of 0 or 1.

IS-95-A Uncoded Coherent Combining AWGN BER generator (uccabe95)

The purpose of this module is to generate Bit Error Probability as a function of signal-to-noise ratio (SNR) for Direct-Sequence CDMA systems. This model is based on the reception of a multipath signal with fixed, but assumed unknown, amplitudes and phases. Delays are assumed to be perfectly compensated, with the signals combined in perfect time alignment. This model also assumes perfect chip matched filter timing and sampling of a QPSK signal. This is a Chernoff bound. This model assumes that the user's signal is orthogonal to the unmodulated pilot signal, implying that the user's spreading sequence is "balanced" (an equal number of 1's and 0's within any symbol). This model also assumes that other users' interference can be modeled as Gaussian noise. The model used here assumes that a pilot tone is used to create a coherent reference to be used for amplitude/phase compensation for the individual path components before addition. The model is parametized by the number of chips over which the unmodulated received pilot is averaged to derive the I/Q amplitude estimates. These I/Q amplitude estimates are used to align phases and weight amplitudes of the Num_Paths_L multipath components.

IS-95-A Uncoded Coherent Combining Rician BER generator (uccrbe95)

The purpose of this module is to generate Bit Error Probability as a function of signal-to-noise ratio (SNR) for Direct-Sequence CDMA systems. This model is based on the reception of a multipath signal with two components per path. The first is a specular component and the second is a Rayleigh fading component, resulting in the classical Rician for each path. The specular component is assumed to have fixed, but unknown, amplitude and phase. The Rayleigh component (sometimes called the diffuse component) is modelled as a random channel magnitude with Rayleigh distribution. The Variance of the distribution is calculated based on the magnitude of the path component and the ratio of the specular to diffuse energy for that path, "Specular_To_Diffuse", which describes the power distribution between the specular and diffuse components. The phase associated with the diffuse component is assumed to be uniformly distributed. Delays are assumed to be perfectly compensated, with the signals combined in perfect time alignment. This model also assumes perfect chip matched filter timing and sampling of a QPSK signal. This is a Chernoff bound. This model assumes that the user's signal is orthogonal to the unmodulated pilot signal, implying that the user's spreading sequence is "balanced" (an equal number of 1's and 0's within any symbol). This model also assumes that other users' interference can be modelled as Gaussian noise. The model used here assumes that a pilot tone is used to create a coherent reference to be used for amplitude/phase compensation for the individual path components before addition. The model is parametized by the number of chips over which the unmodulated received pilot is averaged to derive the I/Q amplitude estimates. These I/Q amplitude estimates are used to align phases and weight amplitudes of the Num_Paths_L multipath components.

IS-95-A 9600 bps Forward channel VITerbi decoder (fvit9695)

This Subnet implements a Viterbi decoder for the rate 1/2, constraint length 9 convolutional code employed in the forward coherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A 4800 bps Forward channel VITerbi decoder (fvit4895)

This Subnet implements a Viterbi decoder for the rate 1/2, constraint length 9 convolutional code employed in the forward coherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A 2400 bps Forward channel VITerbi decoder (fvit2495)

This Subnet implements a Viterbi decoder for the rate 1/2, constraint length 9 convolutional code employed in the forward coherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A 1200 bps Forward channel VITerbi decoder (fvit1295)

This Subnet implements a Viterbi decoder for the rate 1/2, constraint length 9 convolutional code employed in the forward coherent link of the IS-95-A system specification. It operates the General Trellis Decoder Module in a frame-based mode, decoding each frame of the despread and demodulated data stream.

IS-95-A Forward channel WAveform DeModulator 1 (fwadm195)

This Subnet implements an IS-95-A forward channel chip demodulator. This model is responsible for correlating real-valued samples from a sampled-data waveform against stored pulses for the purpose of matched-filter chip demodulation.

IS-95-A Forward channel WAveform MoDulator 1 (fwamd195)

This Subnet implements the IS-95-A forward channel chip modulator. This model is responsible for employing real-valued chip modulating coefficients to weight pulse-like segments of a sampled-data waveform. These overlapping segments are superposed and combined to form a discrete approximation to a continuous-time modulating waveform.

6. Performance Bounds

It is usually prudent to reconcile the empirical results of simulations with analytical benchmarks, when they can be calculated. Often these benchmarks must be made under simplifying assumptions, associated with idealized conditions. Even under these conditions, benchmarks often provide a good rule of thumb or a useful “sanity check” to verify various trends often observed in simulation studies. ICUCOM attempts to incorporate these benchmarks into ACOLADE whenever possible. The Library contains a number of useful performance bounds on coded and uncoded bit error probability and other measures of performance for CDMA systems. These analytical bounds are useful in the analysis of IS-95-A system performance and were used extensively to validate the operation of the models in the Library. These bounds employ combinations of Chernoff and union bounding techniques, which are familiar to most communications engineers. We provide some examples of bounds used to benchmark various types of simulations in the next section.

7. Examples

In this section we provide several examples which demonstrate the use of selected models in the Library for modeling IS-95-A CDMA systems. While by no means exhaustive, these examples provide some insight into the level of detail at which the IS-95-A system is modeled with the Library. The examples also illustrate some interesting issues that arise in the development of simulations of such systems.

This section assumes that the reader is familiar with the operation of the ACOLADE system. A basic understanding of ACOLADE can be acquired by reading the “ACOLADE Overview”.

7.1 Channel Measurement (Sounding)

Before multipath signals can be successfully recombined, the multipath delay profile of the channel must be established. In addition, if this characteristic is changing with time, such variations must be accurately tracked.

Direct sequence spread spectrum signals allow the delay of signals to be estimated with a precision that is approximately equal to the reciprocal of the signal bandwidth. This is approximately equal to the chip duration. In the case of IS-95-A signals, this is 1/64’th of the symbol duration. Thus, signals with arrival times separated by only a small fraction of a symbol may be effectively combined.

Delay estimates are often performed by correlating a received PN sequence against shifted copies of the known transmitted sequence over a time window. Correlation peaks at given values of delay indicate the presence of a multipath component at that delay. In the forward IS-95-A link, there is always an unmodulated pilot transmitted along with the subscriber’s signals that makes this process simple.

Figure 5 shows a block diagram of a coherent Rake sounding loop used to estimate the delays in a received multipath signal. On the far left is an ***IS-95-A Pilot Spreader***, driven by an ***IS-95-A Pilot Constant Generator***. The output of the Pilot Spreader is a complex PN sequence, with independent spreading codes applied to the I and Q sequences, as shown in the bottom of Figure 1.

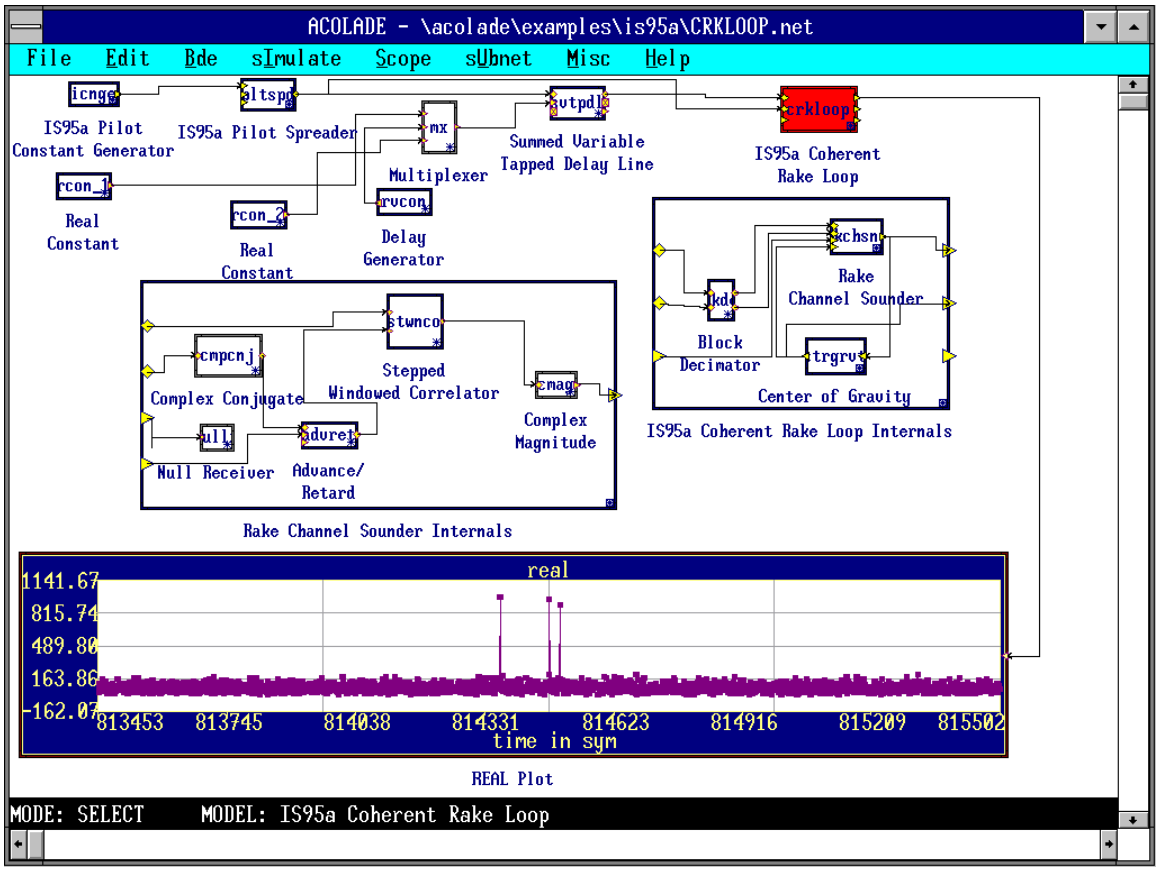

Figure 5. Block diagram of a Coherent Rake Sounding Loop.

The complex sequence output from the Pilot Spreader is fed to the input of a ***Summed Variable Tapped Delay Line*** model which splits the signal into several weighted and delayed components (in this case three) before summing the components together. This model is a key component within the ***Time-varying Multipath Channel,*** but we use it in its elemental form here to illustrate basic concepts. The weights and delays associated with the tapped delay line model may be either fixed in the parameter list of the model or programmed through the input ports. In this case, the three components have constant complex weights of {(-0.707,0.707), (-0.707,-0.707), (0.0,-1.0)}.

The delays for the Tapped Delay Line are programmed through an input port. Three delays, corresponding to the three paths, are combined serially through a ***Multiplexer*** model. The first and third taps are fixed at delays of -7 and 10 chips, respectively, being generated by ***Real Constant*** models. The delay of the second tap is stepped over the sequence (0.0,32.0,64.0,32.0,0.0,-32.0,-64.0,-32.0,0.0) within a period of 640 symbols or 40960 chips. In this simple experiment, the delays are quantized to an integer multiple of the sample times. The tapped delay line model allows interpolation between samples, however, in the general case.

The time-varying multipath signal is then applied to the input of an ***IS-95-A Coherent Rake Loop***. It is the function of the Rake Loop to perform the various correlations necessary to perform the channel sounding. For this purpose, the Rake Loop inputs a copy of the original PN sequence generated by the Pilot Spreader.

The internal composition of the Rake Loop Subnet is shown immediately below the Rake Loop model block. On channels which are slowly time-varying, it is often possible to perform channel measurements only intermittently. This is the reason for the ***Block Decimator*** model at the input of

the Subnet. Its function is to input only every n'th block of data to the ***Rake Channel Sounder***. The Sounder has the function of correlating a reference subsequence of length "*Correlation_Length*" against "*Estimation_Window*" time-shifted versions of the received PN sequence. The Estimation_Window parameter is selected to monitor a window of chip delays large enough to encompass the total multipath delay spread of the channel. In this case, Correlation_Length is set to 512 chips, or 8 symbols. The Estimation_Window is set to 2048 chips, or 32 symbols.

The ***Center of Gravity*** (COG) model is responsible for making long-term adjustments to the PN sequence alignment, and is designed to keep the delayed path estimates near the center of the estimation window as time evolves.

The internals of the Sounder are shown below the Rake Loop internals. The Sounder sends the reference PN sequence through a ***Complex Conjugate*** block before it is applied to a ***Stepped Windowed Correlator*** model that actually performs the correlations. The Correlator performs the Estimation_Window correlations of length Correlation_Length, as described above. Its output is a serial sequence of correlation values, each corresponding to a given value of delay shift. After outputting from the Correlator, the complex correlation values are input to a ***Complex Magnitude*** model block, to calculate the magnitude of the complex correlation, irrespective of any phase shift the signal may have undergone on the channel. This model block could easily be replaced with a magnitude squared model block, if desired.

After the correlation estimates are generated by the Sounder, they are input to the COG model block, which is another Subnet. The internals of the COG are displayed in Figure 6.

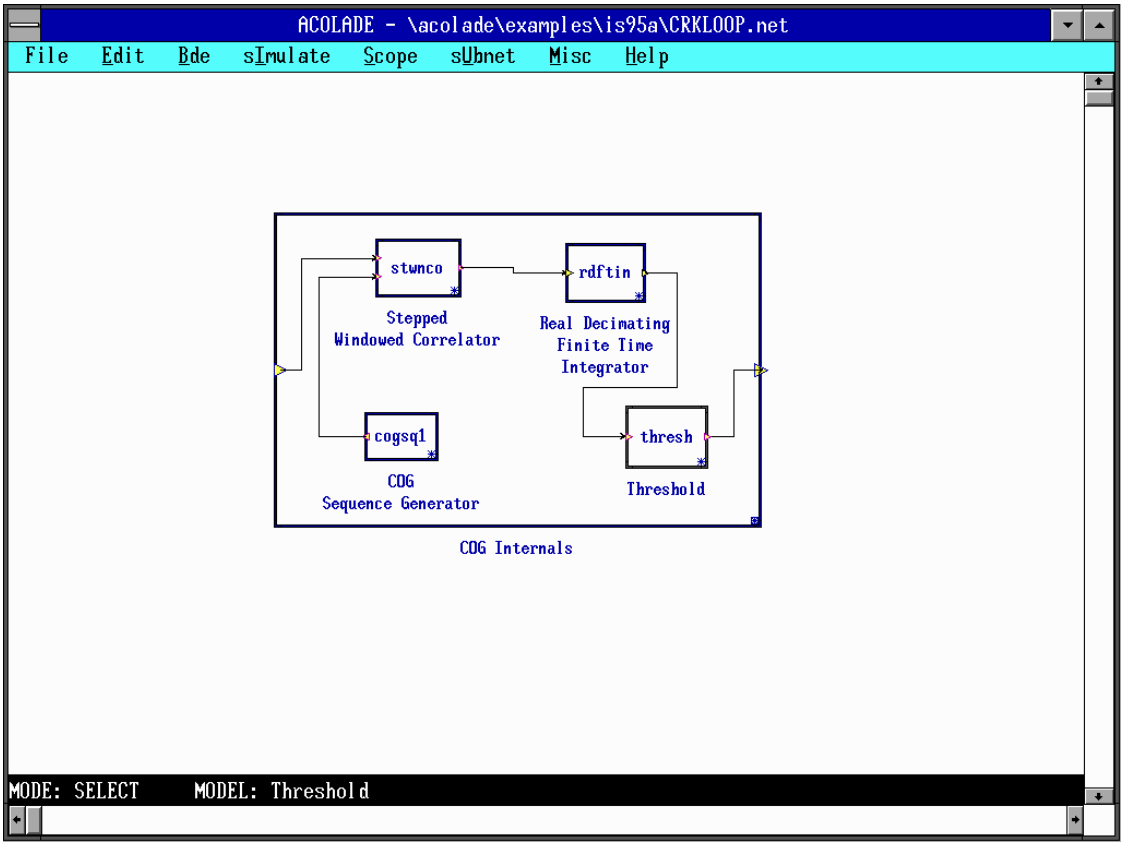

Figure 6. Internals of the COG (Center Of Gravity) Subnet.

The COG circuit consists of another Stepped Windowed Correlator model which is responsible for correlating the sequence of correlation values against a fixed sequence generated by a ***COG Sequence Generator*** that generates a ramp function with zero mean. The correlation outputs from this process are averaged by a ***Real Decimating Finite-Time Integrator*** Module in order to create an error signal which is employed to make long-term adjustments to the reference PN sequence timing.

Finally, the Finite-Time Integrator Module outputs its smoothed version of the error signal to a ***Threshold*** Module which attempts to stabilize the loop with respect to any AGC inaccuracy or short-term received power variations. It is a non-linear element which, in this case, limits the relative shift adjustment to one chip per iteration of the loop. For the frequencies at which the IS-95-A system is designed to operate, multipath conditions vary no faster than this.

The output of the Rake Loop is displayed at the bottom of the screen in Figure 5. The plot shows one sweep of the Rake Loop output. This particular sweep has been taken at a time when the second delay of the multipath channel is at its maximum positive excursion.

A channel measurement system of the form illustrated in this example has certain advantages when implemented in programmable digital hardware. The correlations involved can be implemented with FFT's, thus allowing fairly efficient processing. Additionally, if the chip matched filter outputs are sampled at a sufficiently high rate, the despread outputs may be derived directly from the sounding loop and passed onto a multipath combiner.

For simplicity, we have illustrated the operation of the sounding loop in a scenario that is equivalent to sampling chip matched filter outputs with ideal sampling times occurring once per chip interval. In practice, the sounder could be run on a received sequence consisting of multiple samples per chip to provide more refined delay estimates. Alternatively, sounder outputs are often used as coarse estimates of delay which are used, in turn, as an initial starting point for a separate chip demodulator/bit synchronizer finger that would be assigned to selected delay paths. Examples of both of these scenarios are provided in the example netfiles shipped with the Library.

7.2 Forward Link Uncoded Performance

This example is designed to demonstrate the uncoded error probability performance of a classical 3-element Rake receiver operating over a multi path channel. Several scenarios are presented in which different types of uncertainties about synchronization, power estimation, etc. are assumed. We illustrate the use of Chernoff bounds as a benchmark in these types of simulations.

The example comprises a set of experiments, each with increasing complexity. Our goal is to show how each portion of a system architecture can be verified before the next level of complexity is added, either in the system model or the environment model. The general procedure is to validate and benchmark each layer of the design before another level of complexity is added. In this fashion, simple sources of error may have their effect analyzed before being masked by possible degradation or errors within higher layers of the design. This is a prudent technique for system development in general, not simply for the specific examples shown here.

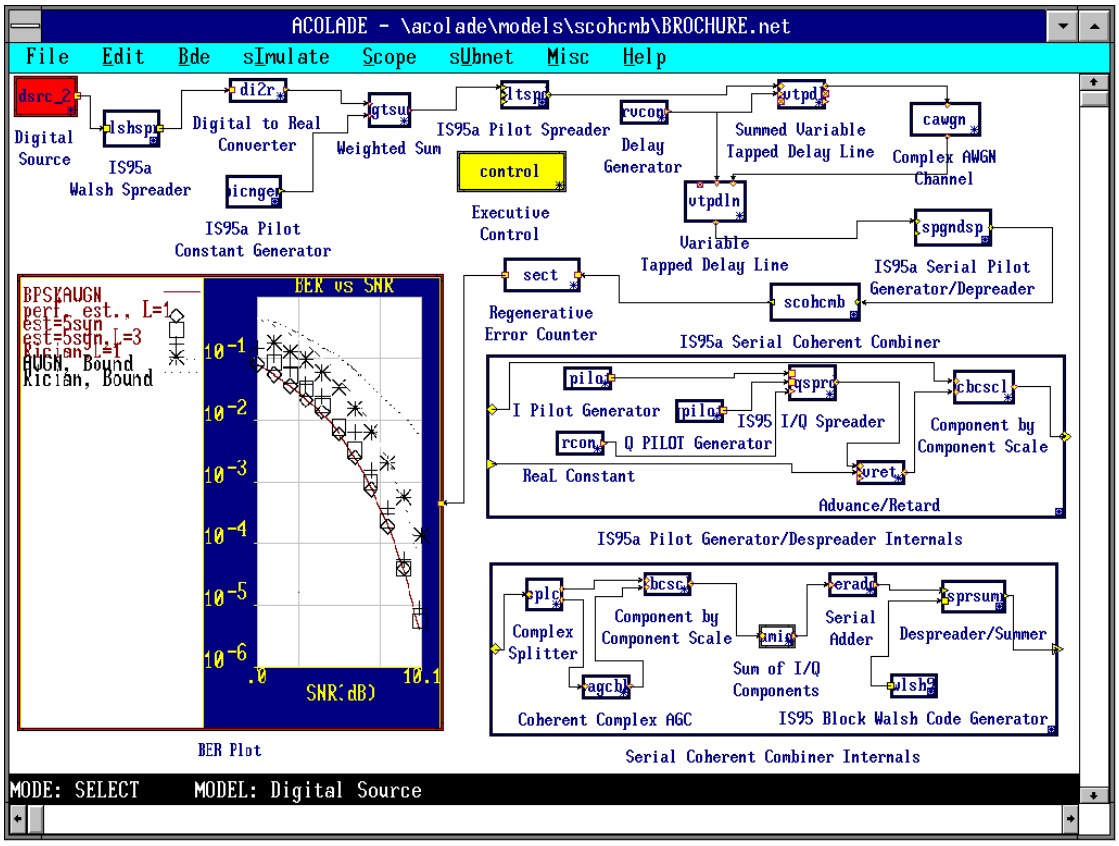

Figure 7. Block Diagram of a Coherent IS-95-A.

Figure 7 shows the block diagram of a coherent IS-95-A link with a pilot (always required) and one subscriber. The pilot signal is generated in this case by PN spreading a constant value generated by the Pilot Constant Generator. The subscriber data sequence is generated with a ***Digital Source*** Module. The source bit stream is passed through a ***Walsh Spreader*** Subnet. The internal details of the ***Walsh Spreader*** subnet are shown in Figure 8. The Walsh Spreader employs a ***Block Walsh Sequence Generator*** and a ***Binary Spreader*** Module to create a {+1.0,-1.0} Walsh code sequence. The internals of the ***Binary Spreader*** subnet are shown also in Figure 8. The pilot signal and the user signals are weighted and combined with a ***Weighted Sum*** Module. In this case the ratio of subscriber to pilot power is 0.0 dB. The output of the Weighted Sum is passed into the Pilot Spreader discussed in the last example. The signal is then passed into a Summed Variable Tapped Delay Line Module for the generation of multipath, as in the last example. In all the experiments in this example we use a fixed multipath channel.

The signal is passed through an AWGN channel, which is responsible for adding Gaussian noise samples to the sampled-data signal.

The output of the Summed Variable Tapped Delay Line Module is passed into a ***Variable Tapped Delay Line*** Module, which has the ability to tap its input sequence at various delays. In this case, the tapped delay line Modules are fed by the same ***Delay Generator***

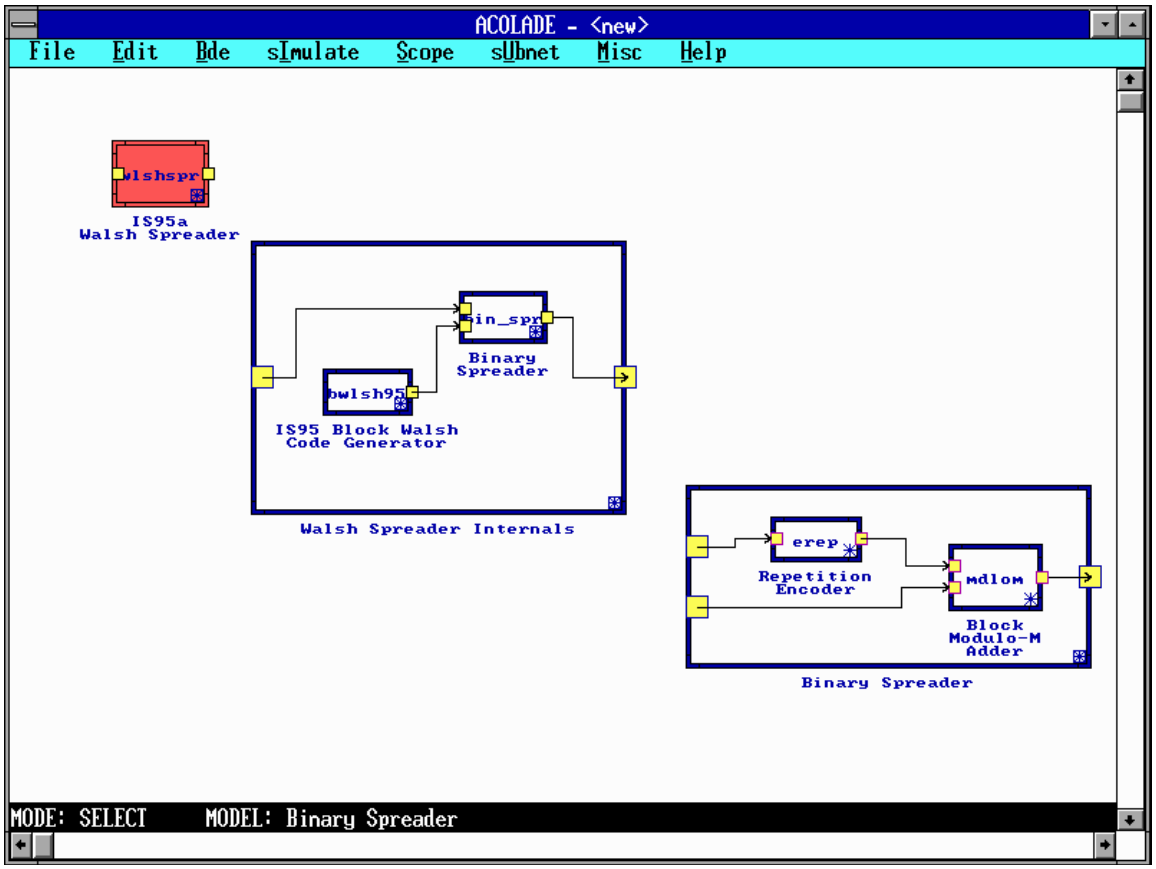

Figure 8. Internals of the Walsh Spreader Subnet.

Module, so that the estimate of the fixed delays is always perfect. Additionally, the paths are exactly aligned at the output of the Variable Tapped Delay Line and output serially. At this point, the signal is processed with an ***IS-95-A Serial Pilot Generator/Despreader*** Subnet to despread the signal by correlating against a time-aligned copy of the pilot sequence. The internals of the Generator/Despreader are shown under the Subnet model block. It consists of two pilot generators for the I/Q components with an ***I/Q Despreader Module.*** The sequence timing may be optionally adjusted. A signal applied to the input ports of the Subnet which are applied to two ***Advance/Retard*** Modules, employed to adjust the timing of the sequence. The spreading sequence is multiplied I/Q component-wise by the respective I/Q components in the data sequence by the ***Component by Component Scale*** Module.

Finally, the despread serial sequences are passed into the ***IS-95-A Serial Coherent Combiner*** Subnet, which is responsible for combining and Walsh despreading of the multipath signal components. The internals of the Coherent Combiner Subnet are shown below the Subnet model block.

After entering the Subnet, the signal is split into two streams, one entering the ***Complex Coherent AGC*** Module and the other directed to the input of the ***Component by Component Scale*** Module. The AGC Module is responsible for averaging the individual I/Q chip streams over a specified number of chips to determine the average values of these components for a particular path. Presence of the pilot signal ensures that these estimates can be used to determine the amplitude and phase of the arriving multipath components. The number of chips to average over is, as usual, a compromise between estimation accuracy and timely tracking of changes in the multipath structure of the channel. In this case, it is set to 4 symbols. The AGC is set to deliver a new estimate at the start of each symbol.

The Component by Component Scale Module is employed to utilize the periodically updated estimates of the I/Q components to de-rotate and amplitude scale the various multipath components.

After appropriate weighting and phase alignment, the I/Q components of each path are added within a ***Sum of I/Q Components Module***, after which they are passed through a ***Serial Adder*** Module to sum them.

Finally, the signal is despread by an ***IS-95-A Despreader/Summer*** Subnet, which is fed by another Walsh code generator. The internals comprise a ***Digital to Real Converter***, a ***Real/Real Multiplier*** and a ***Chip Combiner***, which combines blocks of 64 Walsh chips to produce a final decision statistic.

A ***Regenerative Error Counter*** maintains a copy of the pseudorandom number generator in the Digital Source in order to tally bit errors for the ***Bit Error Rate Plot***.

There are several simulation results overlaid on the BER plot. These experiments begin with very simple scenarios, in order to establish validity of basic operation. The first experiment is conducted with the number of paths L = 1 and perfect I/Q level estimates at the receiver. In this case, the known level estimates were “patched” through to the Component-by-Component Scale Module in the combiner. This was effected by deleting the connection from the output of the AGC and setting the true value in the parameter list of the Scale Module. This simulation result is shown in diamonds. This empirical result is shown overlaid on a solid curve, representing theoretical performance of antipodal signaling in AWGN. The independent variable, in this case, is the ratio of the information bit energy to the energy density of the single-sided AWGN process. Note that the empirical results fall exactly on the theoretical curve, within statistical accuracy. A minimum of 25 errors were collected for each value of SNR in all simulations.

The next simulation was collected with the AGC active, again with L = 1. In this case, inaccurate signal envelope estimates will not affect the performance of the system, since only one path is involved. Inaccuracies in the phase estimates cause a skew in the effective decision boundary, however, and the results are evident for this experiment, shown in stars. The results show a small loss in performance, less than a dB.

The next simulation adds two additional paths, with all path strengths being equal. We always normalize the received SNR with respect to the sum of the energies in all paths. This result is shown in plusses. A Chernoff bound for this case is shown overlaid in the dashed curve. Note that the simulation results are asymptotic to the bound at high SNR, both being displaced to the right of the L = 1 result.

The final experiment we demonstrate is with one path in Rician fading, using AGC estimates. This result is shown in stars. Note that the Rician fading causes a severe degradation, due to the fact that the channel magnitude is very low on certain symbols. A minimum of 100 errors was collected in this experiment. Another Chernoff bound is overlaid in a dotted line.

7.3 Noncoherent Orthogonal Walsh Modulation in Rician Fading

This example demonstrates the use of noncoherent orthogonal modulation in Rician Fading. A Rician channel consists of two components, a so-called “diffuse” component that undergoes Rayleigh fading and a so-called “specular” component, consisting of an attenuated and phase-shifted copy of the original signal. This is a common model for many channels subject to certain fading conditions. Individual path components of multipath fading channels are often modeled as undergoing Rician fading. The simulation shown in this example is simplified to allow us to focus on just the effects of fading associated with a single transmission path. Many other IS-95-A examples demonstrate the detailed operation of the end-to-end Forward and Reverse links.

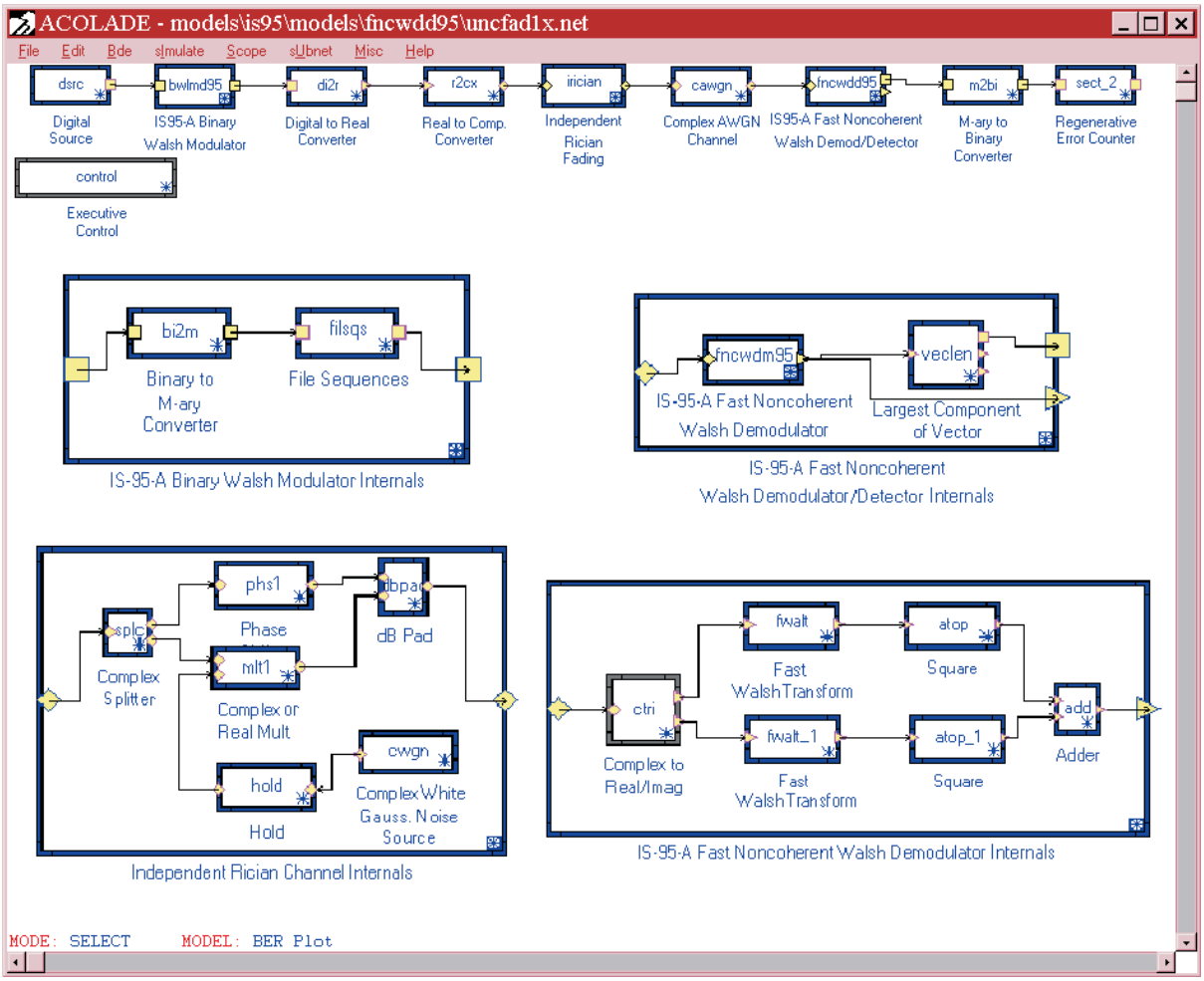

Figure 9. Illustration of Noncoherent Detection of Orthogonal Sequences.

The example network is shown in Figure 9. It contains a ***Digital Source*** model which generates pseudo-random binary digits. The bits are then applied to the input of an **IS-95-A Binary Walsh Modulator**, which is responsible for transforming blocks of six bits into an index which is used to select among the 64 possible orthogonal Walsh sequences of length 64. The internals of the Binary Walsh Modulator are shown overlaid on the main canvas on the top left. A **Binary to M-ary Converter** is employed to group the source bit sequence into sub-sequences of length six before converting to 64-ary digits. These digits are then input to a **File Sequences** model and used as an index to select among the 64 Walsh sequences, which are read from a file at the start of the simulation. The File Sequences model is a very general model which can be used to read an arbitrary set of sequences. It is employed in the coherent Rake receiver example to continuously emit a fixed (but user-selectable) member of the Walsh sequence set for the purpose of orthogonal data-independent spreading of the user's information sequence. In the Reverse channel, the user's information bit sequence is actually used to select the specific sequence to be transmitted. The input port of the File Sequences model provides the index into the array of file sequences to be output. Note that the Binary Walsh Modulator has an overall data rate increase (bits to chips) of 64/6. As we shall soon see, the Walsh modulation can be viewed as a low-rate error-control code, with associated coding rate 6/64 (bits per chip). Elements of the Walsh sequences are {0,1}, representing binary digits.

After the proper Walsh sequence is emitted from the modulator, its elements must be mapped to modulating coefficients which are used to construct the pulse-like analog waveform which will be sent over the channel. The IS-95-A Reverse Channel uses the orthogonal sequences in conjunction

with BPSK, or antipodal, modulation applied to both the I and Q channels. A separate PN sequence is overlaid on the individual I and Q channels, at the Walsh chip rate, before the channel sequences are used as modulating coefficients applied to offset pulse modulators for the I and Q channels. Since we are only interested in evaluating the performance of the noncoherent sequence demodulator, several simplifications can be made. In this example, we simulate BPSK signaling by mapping the bits {0,1} to antipodal modulating coefficients {+1.0,-1.0} with a **Digital to Real Converter**. The modulating coefficients can be thought of as samples of a sampled-data waveform, sampled once per chip signaling interval. These coefficients are then mapped onto the inphase channel of a complex-envelope signal with a **Real to Complex Converter**, corresponding to BPSK, or antipodal signaling. We then corrupt the complex modulated signal by passing it through Rician fading and AWGN. These are important simplification techniques that can shorten simulation time, thereby allowing more comprehensive parameter tradeoff studies to be performed.

The internals of the **Independent Rician Fading** Subnet model are shown on the bottom left of the canvas. It is an abstract channel model designed to simulate the effects of slow Rician fading in the limit as the fading bandwidth becomes a vanishingly small fraction of the symbol rate. This is the limiting case of “flat” fading, but is reasonably well-approximated when the fading bandwidth is on the order of 1/10’th the symbol rate. A **Complex Splitter** model splits the Subnet input into diffuse and specular components. This particular model generates a complex Gaussian variate with the **Complex Additive White Gaussian Noise** model and holds (replicates) it for a period of time before inputting it to a **Complex Multiplier**. A **Hold** model is employed for this purpose. A complex Gaussian variate with independent I and Q components results in Rayleigh-distributed magnitude and uniformly distributed phase. The resulting magnitude scaled and phase-shifted signal is applied to a **dB Pad** model where it is scaled and added to a scaled specular component. The specular component possesses a fixed, but user-specifiable phase. The scaling applied in the dB Pad model achieves a given user-specified specular-to-diffuse (S/D) energy ratio. The result of this process is that the signal undergoes Rician fading that is constant during intervals, but independent from interval-to-interval. This is extremely useful in the simulation of coded systems employing interleaving. If the “hold” period is chosen to be one symbol (or the number of symbols grouped together into an interleaving “cell”), the effects of an arbitrarily large interleaver can be simulated without actually having to use one. In simulating coded systems, this is an appropriate model for channels with a slow fading rate compared to the channel symbol time, but fast with respect to the total interleaving delay. We will employ this model in other examples where we compare the results of the current uncoded system with coded systems employing different interleaving strategies. After processing with the Independent Rician fading model, the signal is corrupted with AWGN with another complex AWGN model.

On the receiving end, the corrupted channel sequence is processed by an **IS-95-A Fast Noncoherent Walsh Demod\Detector** model. Non-coherent demodulation and/or detection are frequently employed on fading channels, since the phase is often varying too rapidly to track. The internals of the Walsh Demod/Detector are shown in the top right of the canvas of Figure 9. It consists of an **IS-95-A Fast Noncoherent Walsh Demodulator**, which is responsible for correlating each 64-ary received sequence against all of the 64 possible contenders. A single output value is emitted by the Demodulator, corresponding to each correlation, resulting in 64 output values for each received Walsh sequence. Each resulting output vector is passed to a **Largest Component of Vector** model to identify and output the index corresponding to the highest correlation value, thereby performing signal detection. The internals of the Demodulator model, another Subnet, are shown in the bottom right of Figure 9. Since the antipodal signal undergoes an unknown phase shift during transmission, it may be received with either positive or negative projections on the I/Q axes. For this reason, a complex non-coherent correlation is performed, the results of which are independent of phase. This procedure is effected by correlating against the I/Q channel signals individually, then squaring and summing the results. A **Fast Walsh Transform** model is employed to perform the correlations on the individual channels. The results are then squared with a **Square** model and summed with an **Adder** model before being delivered to the output.

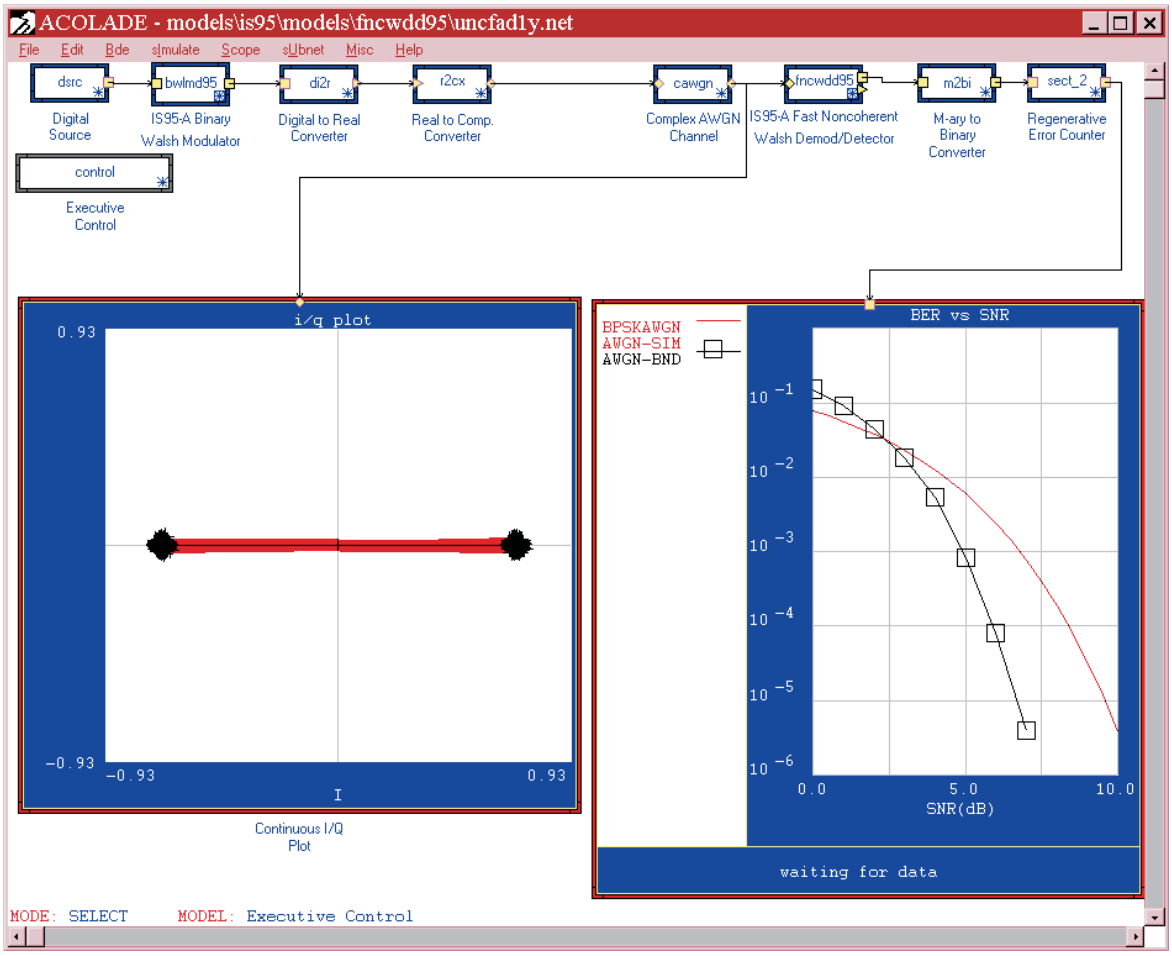

Figure 10. Noncoherent Detection Performance in AWGN.

The Demodulator/Detector delivers its stream of 64-ary output symbols to the input of an **M-ary to Binary Converter**, where each symbol is converted into 6 bits. Finally, the output is delivered to the input of a **Regenerative Error Counter**, used and described in many other examples.

In what follows, we will carefully examine two test cases for different fading parameters, then present results for the additional simulation runs that were performed.

The first test case was run with the Specular to Diffuse power ratio at 100.0 dB, or essentially under AWGN conditions. The results are independent of the fixed phase offset, and it is simply chosen as 0.0. The results of this first test are shown in Figure 10. On the left of the canvas, we have shown a **Continuous I/Q Plot** at the output of the channel. This is a plot of the trajectory of an arbitrary complex signal in the I/Q plane. In this case it is employed to show the magnitude and phase variations of the signal over certain periods of time. In this case, we have set the plot parameters to overlay the output of 32 symbols, or 1024 chips. The I/Q plot displayed here was captured at a chip SNR value of 20.0 dB. With antipodal signaling employing unit-magnitude chip coefficients, the variance of the WGN samples generated by the Complex AWGN Channel is set to a value of -20.0 dB. Because of the rate 6/64 coding implicit in the modulator, this corresponds to a bit SNR of $20.0 + \text{Log}_{10}(64/6) = 30.05$ dB. It will be noticed that the signal points are alternating between 0 and 180 degrees, or +1.0, -1.0 on the inphase axis. The positive and negative projections are due, of course, to the positive and negative chips values of the Walsh sequence elements. Note that the Gaussian noise is causing a circular dispersion about the chip mean values.

The Bit Error Rate (BER) as a function of bit energy-to-noise power ratio (Eb/N0), in decibels, is shown in the bit error rate plot on the right of the canvas. The smooth curve corresponds to antipodal signalling in AWGN. The dashed curve corresponds to the theoretical BER performance for non-coherent reception of 64-ary orthogonal signals in AWGN. The open squares are the simulation results. In all cases, at least 100 bit errors were tabulated for each SNR value by setting the "Minimum Errors Requested" parameter in the error counter to that value. This provides extremely high statistical significance levels to our results. It will be noted that the empirical results fall exactly on the theoretical results for 64-ary modulation. What is interesting to note additionally is the fact that the 64-ary orthogonal modulation provides significant coding "gain" over the BPSK case, even with noncoherent detection. This is due to the fact that orthogonal sequence modulation can be properly considered a form of error-control coding. Using more than one transmission per bit allows the introduction of controlled redundancy into the transmitted chip stream. Alternatively viewed, this allows transmitted chip sequences (codewords) to be differentiated in a number of positions from one another, allowing the probability of ***sequence*** error to be greatly reduced. The BER curve for the coded orthogonal modulation shows the normal threshold behavior associated with coded systems in general. At low SNR, the performance is worse than uncoded, but exceeds it at higher SNR values. For further elaboration of the theory of data transmission by orthogonal functions, see [5].

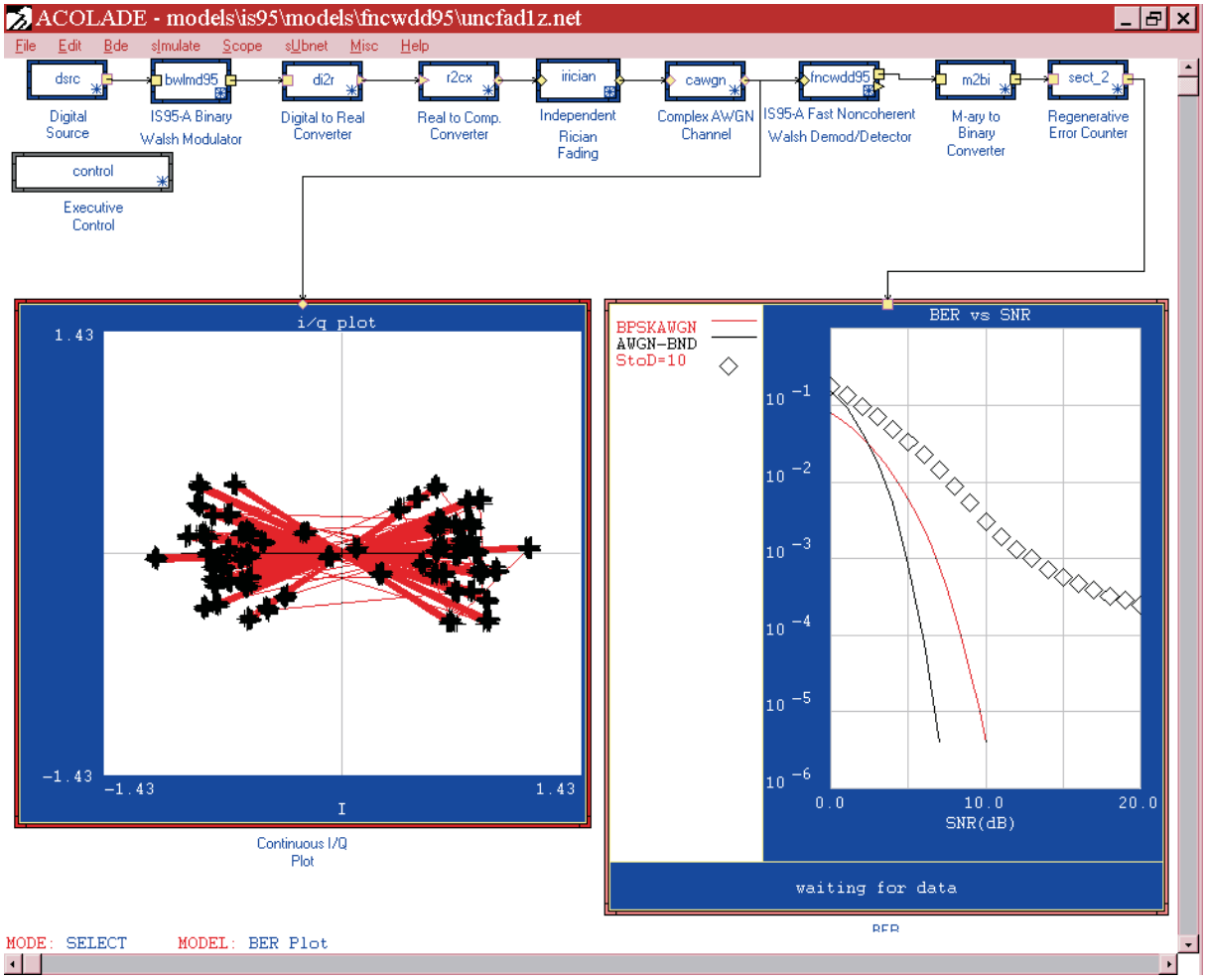

Figure 11. Noncoherent Detection Performance in Rician Fading - S/D = 10 dB.

The second test case we will illustrate is the case of Rician fading at an S/D ratio of 10.0 dB, a value not atypical of many mobile channels. The results of this test are illustrated in Figure 11. The most interesting result is the behavior of the I/Q plot. Note that the BPSK signal constellation is now randomly skewed in phase and appears with random amplitudes, but still clusters somewhat, about the nominal position. On Rician channels, severe degradation in BER performance can occur when the Rayleigh component is approximately equal in magnitude to and opposite in phase from the specular component. In this case, signal cancellation occurs, resulting in an error probability near 0.5 for the corresponding signaling interval. This degradation is noticeable in the BER curve. Note the BER begins to decline at first, but soon reaches a point where the rate of decline stops and actually decreases. This is typical of uncoded system performance on most fading channels. As we will see in other examples, error-control coding can restore most of the performance lost on fading channels.

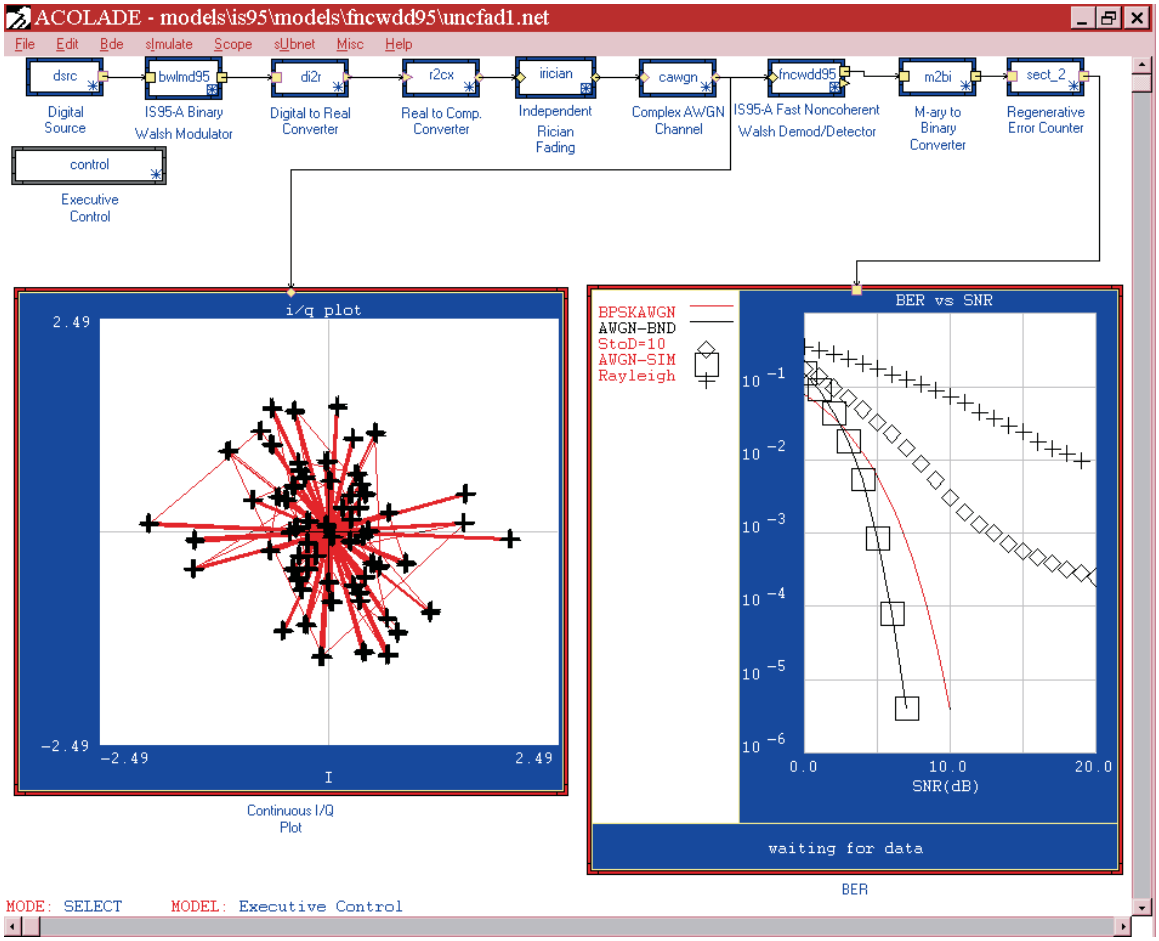

Figure 12. Noncoherent Detection Performance in Rayleigh Fading - Comparison with Other Results.

We finish this example by exhibiting a full set of simulations, performed at various values of S/D. These results are shown in Figure 12. As expected, performance degrades monotonically with decreasing S/D. The limiting case of decreasing S/D is, of course, the Rayleigh channel, which is labeled as such in the legend. In this case, there is no energy at all in the specular component, with the instantaneous received signal power reaching arbitrarily low levels during deep Rayleigh fades. The I/Q plot shows the phase trajectory for Rayleigh fading, which is completely incoherent, due to the absence of any specular component.

There are many more netfiles in the examples directory for uncoded coherent system operation. Some use multiple fading paths and some use path delay estimation with time-varying path delays.

Source code for the models in the IS-95-A library may be obtained under separate license from ICUCOM Corporation.

References

[1] “Mobile-Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System” TIA/EIA/IS-95-A, May 1995.

[2] Viterbi, A. J., “CDMA-Principles of Spread-Spectrum Communication”; Addison-Wesley 1995.

[3] Gibson, J. D., “The Mobile Communication Handbook”; A CRC Handbook published in cooperation with IEEE Press (1996).

[4] Price, R. and Green, P. E., Jr. “A Communication Technique for Multipath Channels” Proc. IRE **46**, (1958) 555-570.

[5] Harmuth, Henning, F., “Transmission of Information by Orthogonal Functions”; Springer-Verlag, New York, (1969).