

Real-Time Multi-Channel Modem Testbed

Author (s)

Mr. Aditya Chopra

Mr. Yousof Mortazavi

Prof. Brian L. Evans

The University of Texas at Austin, Austin, Texas

Product(s) Used : LabVIEW, LabVIEW RT, PXI-1045, PXI-8106, PXI-8186, PXI-5421, PXI-5122

Industry: Education, Research, Wireline communications, DSL, Multi-channel communication

Campaign:

Challenge

The goal of this case study is to build a prototype of a multi-channel wired communication system utilizing Discrete Multitone Modulation (DMT) using commercial off-the-shelf (COTS) components. This would allow a system designer to evaluate communication performance and computational complexity tradeoffs and explore the design space in order to obtain more efficient and cost-effective solutions while still meeting the product specifications. A flexible prototype also allows the designer to validate results and evaluate system design tradeoffs without iterating through the long design cycle associated with custom hardware. The use of commercial off-the-shelf (COTS) components reduces time to market also simplify the modification of system parameters and algorithms.

Solution

The testbed was built using COTS components from National Instruments and various other vendors. The transceiver algorithms execute on a PXI – 8106 embedded processor which was upgraded from PXI – 8186. PXI – 8106 consists of a dual-core 2.2 GHz Intel processor. The transceiver for a 2×2 system is especially suited to a multi-threaded implementation. The A/D converter for the receiver and D/A converter for the transmitter were PXI – 5421 and PXI – 5122, respectively. The embedded processor and data converters were plugged into NI's PXI – 1045 chassis, which provided a synchronized backplane for all components and allowed easy access to the data converters within the LabVIEW environment. The analog front-end of the system included line-driver/hybrid circuits from Texas Instruments and lowpass anti-aliasing passive filters from Analog Devices. A desktop PC running LabVIEW was used to provide a visual interface to the testbed.

Body

The Testbed

In high-speed wired communication systems, a single transmitter-receiver setup may not be able to provide the desired rate of information exchange. Multichannel communication systems, in which multiple transceivers communicate in parallel, are becoming increasingly popular. They are also known as multiple-input multiple-output (MIMO) systems. Another trend is to use multiple carrier frequencies to transmit data, a.k.a. discrete multitone (DMT) modulation. DSL is an example.

We have developed a real-time MIMO DMT testbed with two transceivers, as shown in Figure 1. Due to electromagnetic coupling between the wires, parallel transmission creates a new host of impairments. These impairments include near-end crosstalk (NEXT) and far-end crosstalk (FEXT). The testbed enables a designer to evaluate the tradeoffs in communication performance vs. implementation complexity

tradeoffs in algorithms for mitigating NEXT, FEXT and other impairments. The system has been developed on NI hardware and uses LabVIEW software.

The software solution

The software solution running on the testbed was split into two sections:

Real-Time Components: The real-time software components are the transceiver algorithms and their interfaces to hardware, and were implemented in LabVIEW RT on an embedded controller to meet the hard real-time constraints. The core transceiver algorithms include generation of the transmission signal, decoding the received signal, and various signal processing techniques used to increase the data rate with a low bit error probability. These algorithms were developed in C++ and executed as part of a DLL. The interface to hardware was tasked with maintaining buffers of transmitted and received samples and sending these samples to the data converters. These interfaces were implemented as LabVIEW RT VIs.

Non Real-Time Components: The non real-time components of the testbed include the user interface related sections of the software. These allow the user to control the parameters of the communication link pertaining to hardware (transmission voltage, sampling rate), software (buffer size) and system (equalizer, bit-allocation settings) sections. This interface is executed as LabVIEW VIs on a desktop PC connected via Ethernet to the PXI chassis.

Improvement in computational performance using NI products

The testbed was initially developed using the PXI – 8186 embedded PC and LabVIEW 8.0. Both LabVIEW 8.0 and PXI – 8186 support single threaded implementations, and our solution was barely able to execute within the real-time limitations imposed by the streaming high-speed communication system. Some algorithms such as non-linear equalization and channel tracking were switched off and the sampling rate was lowered to ease the computational requirements of running this testbed. We upgraded our system to PXI – 8106 which is a multi-core processor and LabVIEW 8.5 which allows VI's to execute with multiple threads to take advantage of our multi-core system. Through each upgrade we were able to achieve a significant increase in computational performance, which allowed us to execute the testbed with its full complement of signal processing algorithms at the desired sampling rate.

Figure 3 shows the increase in computational performance measured as the percentage of the received signal buffer in use. A higher percentage implies that the time taken to process a frame is high, which leads to higher buffer usage as data is kept in the buffer for a longer duration. There is significant improvement in computational performance from both individual upgrades i.e. PXI – 8186 to PXI – 8106 and LabVIEW 8.0 to LabVIEW 8.5, respectively. Moreover, this increase in performance was achieved just by plugging in a new processor to the PXI chassis and installing LabVIEW 8.5, in other words, no changes were made to the LabVIEW or C++ code. Thus our LabVIEW based solution was not tied to a particular component of hardware and we were able to quickly adapt our code to take advantage of multi-core processors. This increase in computational power made available to us through our LabVIEW based design opens the possibility to extending our testbed to a 3×3 communication system.

Contact Information

Prof. Brian L. Evans
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712
+1-512-232-1457
bevans@ece.utexas.edu
<http://users.ece.utexas.edu/~bevans/>

Additional Material

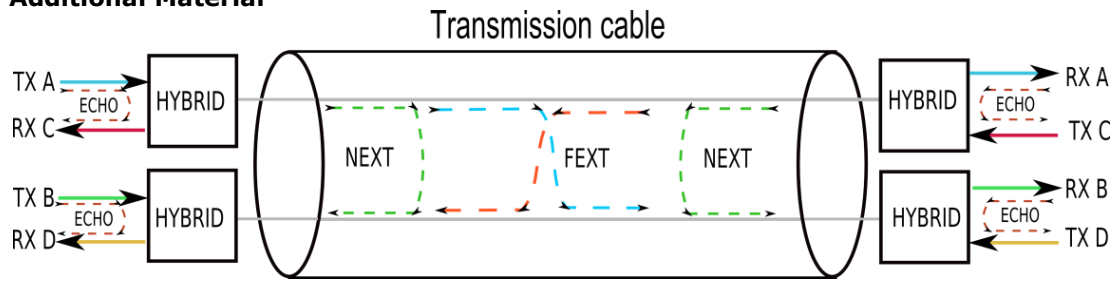


Figure 1 Block Diagram of a two-channel wired communication system

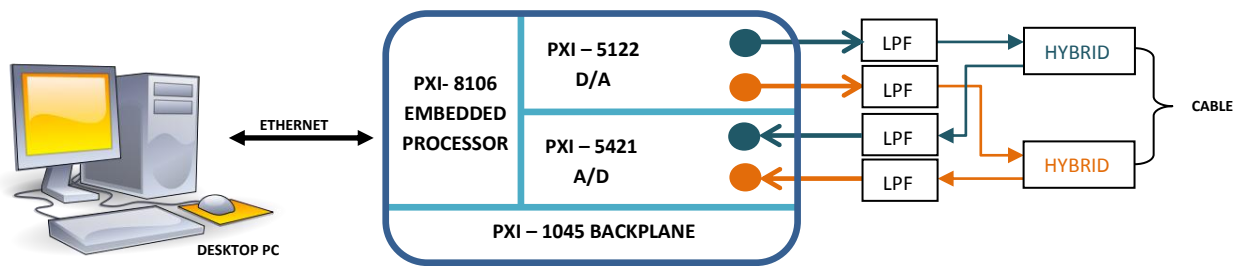


Figure 2 Hardware block diagram of the 2x2 MIMO DMT testbed

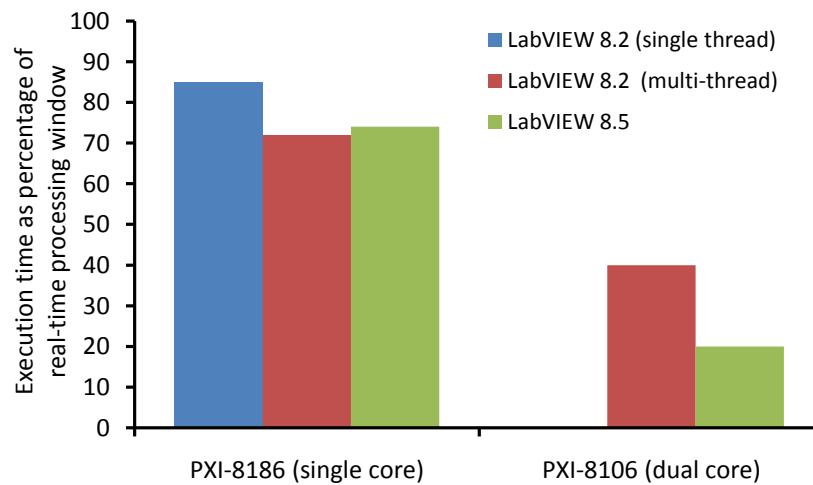


Figure 3 Real-time computational performance of the testbed vs hardware/software solutions