

Experiencia en el desarrollo de una herramienta para la enseñanza de comunicaciones digitales en carreras de Informática

J. García-Naya, H. Pérez-Iglesias, D. Iglesia, M. González-López, A. Dapena

Departamento de Electrónica y Sistemas

Universidade da Coruña

Campus de Elviña, s/n

15071 A Coruña

{jagarcia, hperez, dani, mgonzalezlopez, adriana}@udc.es

Resumen

Este trabajo presenta una herramienta de *aprendizaje asistido por banco de pruebas* formada por tres elementos: plataforma hardware, arquitectura de software multicapa y herramienta gráfica (gtTAL). Para el desarrollo de la herramienta se ha optado por una arquitectura multicapa donde el nivel más alto es una interfaz gráfica que permite la interacción con la plataforma sin necesidad de tener conocimientos de programación de hardware a bajo nivel. Por tanto, los estudiantes pueden testear algoritmos fácilmente sin desarrollar un nuevo programa desde cero, reduciendo significativamente el tiempo empleado para las tareas de implementación y optimización.

1. Introducción

Conscientes del papel primordial que juegan las comunicaciones digitales (especialmente las inalámbricas) en el desarrollo de las TIC y de la sociedad, muchas universidades ofertan, tanto dentro de los títulos de Ingeniería Informática como de Máster en Informática, asignaturas relacionadas con los fundamentos de las comunicaciones digitales y aplicaciones de las mismas.

Los autores de la presente ponencia son Ingenieros en Informática que llevan años (algunos más de 10) impartiendo docencia de materias relacionadas con las comunicaciones digitales en titulaciones de Informática. Nuestra experiencia nos indica que en los cursos de comunicaciones digitales inalámbricas es común el uso de simulaciones por ordenador para ilustrar los conceptos teóricos expuestos a los estudiantes. Excepto en los casos donde se llevan a cabo simulaciones muy complejas, las simulaciones más típicas se realizan bajo condiciones controladas e ideales, ignorándose a menudo efectos muy importantes introducidos por los elementos hardware co-

mo, por ejemplo, los causados por las antenas, por los convertidores D/A y A/D o por los amplificadores de radio frecuencia (RF). Algunos de estos efectos sólo pueden ser bien entendidos si los estudiantes experimentan el problema por sí mismos. Por esta razón, las simulaciones realizadas por ordenador sólo son útiles como punto de partida en la comprensión de los conceptos clave de las comunicaciones digitales inalámbricas modernas pero no permiten estudiar y comprender cuestiones de implementación muy importantes.

Durante los últimos años, se han construido diferentes bancos de pruebas multi-antena de propósito general para evaluar el rendimiento de diversas técnicas de procesamiento de señal y/o estándares (por ejemplo, [4, 6]). A primera vista, uno puede pensar que los estudiantes universitarios pueden participar en el desarrollo de bancos de pruebas pero, de hecho, sólo los estudiantes de posgrado con experiencia de alto nivel están involucrados en tales desarrollos. Desafortunadamente, la creación y posterior desarrollo de software que permita transmitir, adquirir y procesar adecuadamente las señales implican engorrosas tareas de programación a bajo nivel para acceder al hardware, lo que hace difícil probar nuevos métodos que permitan a los estudiantes empezar a interactuar con los bancos de pruebas [11]. Por esta razón, es conveniente añadir un mecanismo al banco de pruebas que permita acceder a diferentes niveles de abstracción. Esto significa que un estudiante que comienza a implementar sus primeros algoritmos debería acceder al banco de pruebas a un nivel superior que otro estudiante que está preparado para abordar detalles de más bajo nivel. En consecuencia, este mecanismo permite a los estudiantes centrarse exclusivamente en el desarrollo y las pruebas de las técnicas, liberándolos de la tarea de programación de bajo nivel.

Distintas universidades y centros de investigación

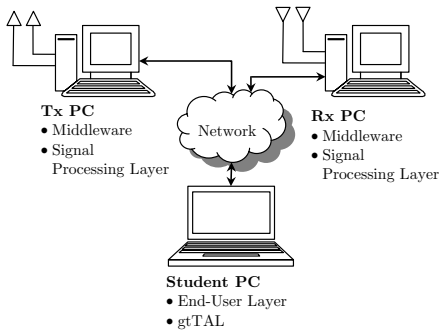


Figura 1: Organización general del banco de pruebas y del PC del estudiante ejecutando gtTAL.

públicos y privados están haciendo un gran esfuerzo para crear bancos de pruebas con fines de investigación pero muy pocos trabajos consideran la posibilidad de aprovechar las posibilidades educativas que ofrecen. En [10] se afirma que los estudiantes que desarrollan implementaciones en un banco de pruebas experimentan un trabajo que les lleva más tiempo, pero que es más gratificante que las tareas de la disciplina teórica. Por otro lado, los estudiantes destacan que el enfoque del banco de pruebas es más detallado y amplio que el enfoque de la teoría y las simulaciones por ordenador. Nosotros estamos completamente de acuerdo con esta afirmación y, por ello, hace 3 años empezamos a trabajar en la idea del “aprendizaje asistido por banco de pruebas”. Como fruto de ese trabajo, en la presente ponencia presentamos una herramienta que permite, incluso a los estudiantes de pregrado, acceder al banco de pruebas en un nivel de abstracción similar al de las simulaciones por ordenador.

El resto del artículo se organiza como sigue. La Sección 2 presenta la arquitectura multi-capas distribuida del software del banco de pruebas. Esta sección describe tanto la parte hardware como las capas de la arquitectura. En la Sección 3 se muestra cómo usar este desarrollo en un tema concreto de comunicaciones digitales. Finalmente, la Sección 4 se dedica a mostrar las principales conclusiones.



Figura 2: Foto del banco de pruebas mostrando el PC Tx, el PC Rx, y gtTAL ejecutándose en un portátil.

2. Descripción general del banco de pruebas

En la figura 1, se muestra la organización general de nuestro banco de pruebas y la figura 2 muestra una foto del equipo actual.

El banco de pruebas se compone de dos ordenadores de sobremesa, uno para el transmisor (denominado PC Tx), y otro para el receptor (llamado PC Rx). La figura 3 muestra el diagrama de bloques de todo el sistema. En ella, se pueden distinguir tres partes principales (de abajo a arriba): el banco de pruebas hardware que nos permite transmitir señales en tiempo discreto en múltiples antenas de 2,4 GHz, la arquitectura software de varias capas que hace al hardware accesible a los usuarios finales (a partir de ahora, consideraremos a los estudiantes también como usuarios finales del banco de pruebas) y, finalmente, la herramienta gráfica gtTAL, que se implementa en la parte superior de la arquitectura software multi-capas.

La capa de nivel más bajo (es decir, el middleware) tiene que estar instalada en los mismos equipos donde este alojado el banco de pruebas hardware, pero las otras dos capas pueden estar en cualquier otro PC disponible. Sin embargo, para simplificar, hemos instalado todo el software necesario en el PC Tx y en el PC Rx. Sólo la capa de usuario está instalada junto con gtTAL en los PCs de los estudiantes (ver figura 1).

La arquitectura software multi-capas presentada anteriormente proporciona un alto nivel de abstracción, lo que permite implementar aplicaciones de

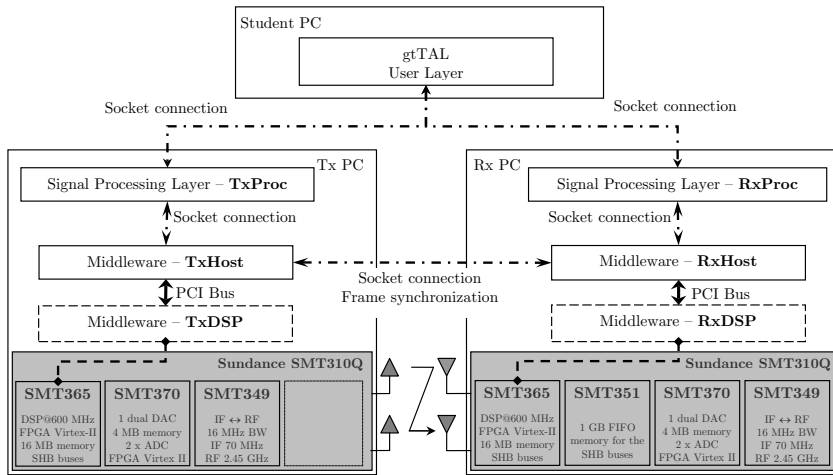


Figura 3: Esquema del banco de pruebas.

usuario sin conocer los detalles hardware del banco de pruebas. Por ejemplo, en este trabajo se presenta gtTAL como herramienta para que el profesor pueda mostrar los principales efectos causados por las transmisiones del mundo real a través del banco de pruebas.

2.1. Descripción del hardware

En la figura 2 se muestra una imagen de los PCs del banco de pruebas (PC Tx y PC Rx). Una descripción más detallada puede ser consultada en [8].

El hardware del banco de pruebas se basa totalmente en el multi-procesador Sundance Ltd (véase la parte inferior de la figura 3). El transmisor se basa en una placa base PCI SMT310Q y el SMT365, un módulo de procesamiento básico equipado con una FPGA, un DSP, buffers de memoria y dos buses capaces de mantener una tasa de transferencia de 400MB/s. El módulo de procesamiento básico se conecta directamente al módulo de adquisición de datos (módulo DAQ), el SMT370. Contiene un convertor D/A dual con memoria dedicada accesible a la misma velocidad que el convertor D/A. El módulo DAQ también tiene dos convertidores A/D. Por último, el módulo DAC está conectado al módulo front-end de RF, el SMT349, realiza las operaciones de conversión desde 70MHz a 2,45GHz con 16MHz de

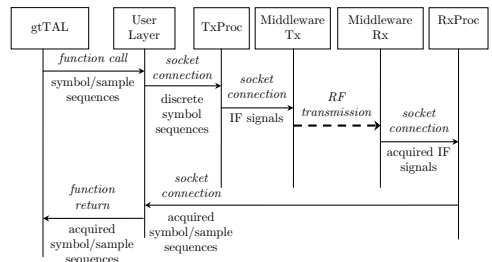


Figura 4: Ejemplo de una transmisión usando gtTAL, la arquitectura software distribuida multi-capas, y el banco de pruebas hardware.

ancho de banda máximo. El receptor utiliza la misma configuración que el transmisor (vea la base de la figura 3), pero incorporando un módulo de memoria buffer, el SMT351, lo que permite almacenar en tiempo real los datos adquiridos por los convertidores A/D para ser posteriormente transferidos al PC Rx.

Con el fin de mostrar las ventajas derivadas de la utilización de una arquitectura software multi-capas, explicaremos paso a paso la transmisión de una trama en la figura 4.

- Una vez que los símbolos a transmitir se han generado en la herramienta gráfica, se llama a una

función pasándole los vectores símbolo correspondientes (un vector por antena transmisora).

- Estos símbolos son enviados a la capa de procesamiento de señal (TxProc) a través de la capa de usuario, donde se convierten a señales pasobanda que son enviadas posteriormente al middleware.
- Cuando tanto el PC Tx y el PC Rx están listos para completar una transmisión, las señales se pasan al banco de pruebas hardware para ser transmitidas a través de las antenas.
- En el lado del receptor, el middleware almacena las señales en los buffers del hardware y luego se envían a la capa de procesamiento de señal del receptor (RxProc).
- Por último, las señales adquiridas se transmiten a la aplicación de usuario a través de la capa de usuario, completando todo el proceso.

La capa de usuario permite configurar varios parámetros. Por ejemplo, la capa de procesamiento de señal puede realizar la sincronización de tiempo y frecuencia o adquirir señales no sincronizadas, datos en bruto o símbolos discretos ya demodulados.

2.2. Descripción de las capas de la arquitectura

Con el objeto de permitir un acceso a alto nivel al banco de pruebas hemos desarrollado una arquitectura formada por tres capas (figura 3): la capa de usuario, la capa de procesamiento de señal y la capa middleware.

Capa de usuario

La capa de usuario interactúa con la aplicación de usuario (la herramienta gráfica, en este caso) mediante una simple función implementada en Matlab® (cualquier otro software que implemente conexiones con sockets también es válido). Su tarea principal consiste en enviar a la capa de procesamiento de señal los símbolos a transmitir junto con los parámetros necesarios, en el lado del transmisor. De la misma manera, la capa de usuario recibe los símbolos adquiridos, siendo notificado si se produce algún error.

El objetivo principal de la capa de usuario es hacer el resto de las capas accesibles a las aplicaciones de alto nivel, teniendo en cuenta el tipo de entorno de desarrollo que utilizan. Por esta razón, la capa de

usuario se ejecuta conjuntamente con esa aplicación (véase figura 3).

Capa de procesamiento de señal

La capa de procesamiento de señal está conectada en red con el usuario y con las capas middleware (véase figura 3). Proporciona acceso remoto y los hace independientes con respecto a las otras capas. Esta capa consiste en dos procesos diferentes que llevan a cabo las operaciones de procesamiento de señal necesarias para conectar al usuario y las capas middleware. El primer proceso (TxProc) recibe los vectores símbolo de la capa de usuario y realiza el up-sampling, el filtrado del pulso, la modulación I/Q y las operaciones de ensamblado de trama a fin de generar las señales IF que se enviarán al middleware. Del mismo modo, el segundo proceso (RxProc) espera por las señales adquiridas en el middleware y realiza las operaciones de sincronización de tiempo y frecuencia seguidas por la demodulación I/Q, filtrado y por el down-sampling. Los vectores resultantes son enviados a la capa de usuario.

Capa middleware

El concepto de middleware constituye un gran salto adelante para la tecnología de bancos de pruebas multi-antena, haciendo al hardware del banco accesible a través de conexiones de red comunes. Esta capa llena el espacio entre el banco de pruebas hardware y la capa de procesamiento de señal, permitiendo transferir señales discretas en tiempo a través del bus PCI y haciendo posible la sincronización entre el PC Tx y el PC Rx usando una conexión de red.

La arquitectura middleware se divide en dos diferentes sub-capas (ver figura 3). La sub-capas superior es responsable de establecer las conexiones de red entre el transmisor y el receptor con la capa más alta (la capa de procesamiento de la señal). La sub-capas inferior se encarga de la configuración hardware del banco de pruebas y del software de control.

El middleware está constituido por cuatro procesos diferentes. Los dos primeros (TxHost y RxHost) implementan la llamada sub-capas superior y se ejecutan, respectivamente, en el PC Tx y en el PC Rx. Están implementadas en lenguaje C++ estándar y usan sockets para establecer las conexiones de red necesarias: una entre los procesos TxHost y RxHost

(utilizada para sincronizar el transmisor y el receptor, por lo que el receptor sabe cuando el proceso de adquisición de señal tiene que empezar); otra, establecida entre el proceso de TxHost y la capa de procesado de señal en Tx y, por último, otra entre el proceso de RxHost y la capa de procesado de señal en Rx. Los otros dos procesos son los del transmisor y receptor que se ejecutan en sus respectivos procesadores digitales de señal (DSPs) disponibles en el hardware del banco de pruebas. Estos implementan la llamada sub-capa inferior. El proceso transmisor DSP (TxDSP) realiza transferencias de datos a través del bus PCI conjuntamente con el proceso de Tx-Host y configura y controla los componentes hardware en el PC Tx. De la misma manera, el proceso RxHost y el proceso receptor DSP (RxDSP) son responsables de transferir los datos a través del bus PCI y, desde el lado DSP, controlar y configurar los componentes hardware del banco de pruebas en el PC Rx.

3. Un ejemplo práctico

Gracias al nivel de abstracción de nuestra arquitectura distribuida, es muy fácil elaborar una herramienta gráfica para el aprendizaje asistido por banco de pruebas. En particular, hemos desarrollado una aplicación gráfica llamada gtTAL (graphic tool Teaching Assistant Learning) orientada a explicar conceptos de modulación y codificación. En la parte de modulación, el estudiante puede trabajar con distintos tipos de modulación (por ejemplo, PAM, PSK, QAM) y estudiar el efecto de variar distintos parámetros como la velocidad de símbolo y el ancho de banda. La parte de codificación es la más significativa de gtTAL porque permite explicar los códigos ortogonales espacio-tiempo (OSTBC) y, es por eso, que la explicaremos con detalle en esta sección.

Los OSTBC consisten en enviar múltiples copias de una señal a través de distintas antenas siguiendo un esquema determinado [12]. Uno de los más conocidos es el código de Alamouti [1] diseñado inicialmente para dos antenas transmisoras y una receptora. El rendimiento de los sistemas de comunicación basados en el esquema de codificación Alamouti dependen en gran medida de la estimación exacta de la matriz de canal. Por esta razón, es interesante estudiar el rendimiento de algoritmos supervisados y no supervisados para estimar el canal. La manera es-

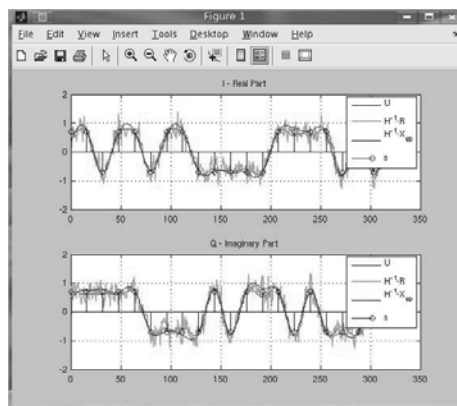


Figura 6: Ventana resultante mostrando los símbolos transmitidos (negro) y señales (rojo), las señales adquiridas después de la demodulación I/Q (verde) y después del filtrado (azul).

tándar para estimar esta matriz consiste en la utilización de símbolos piloto [5] conocidos por transmisor y receptor. Entre los métodos supervisados, el criterio de mínimos cuadrados (LS, del inglés Least Square) [9] constituye un punto de partida frecuente, dada la simplicidad de la técnica resultante.

El principal inconveniente causado por el uso de símbolos piloto es la energía utilizada durante su transmisión. Como los símbolos piloto no contienen datos, el resultado es una pérdida en términos de eficiencia espectral. Este inconveniente puede evitarse mediante el uso de enfoques no supervisados (también conocido como métodos de estimación ciega de canal). Los algoritmos de separación ciega de fuentes (BSS, del inglés Blind Source Separation), pueden estimar el canal. gtTAL tiene implementados distintos algoritmos de este tipo: JADE [7], FastICA [3], técnica de Beres y Adve [2] y otros desarrollados por nosotros.

Utilizando gtTAL, el estudiante puede hacer pruebas, ya sea con secuencias de símbolos discretos de valores complejos o con señales conformadas, tanto en banda base como paso banda. En todos los casos, las señales se transmiten a través del banco de pruebas o por medio de canales aleatorios (es decir, como en una simulación por ordenador convencional). Se pueden modificar varios parámetros, tales como el número de bits a transmitir, o cómo son generados

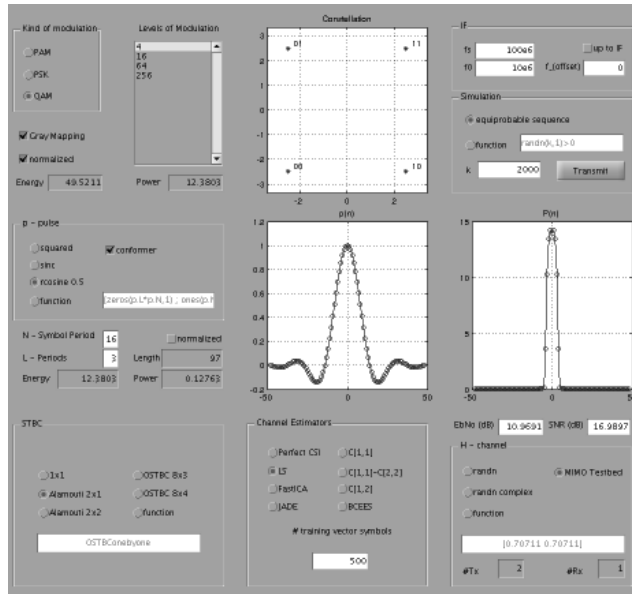


Figura 5: Captura de pantalla de la ventana principal de gtTAL.

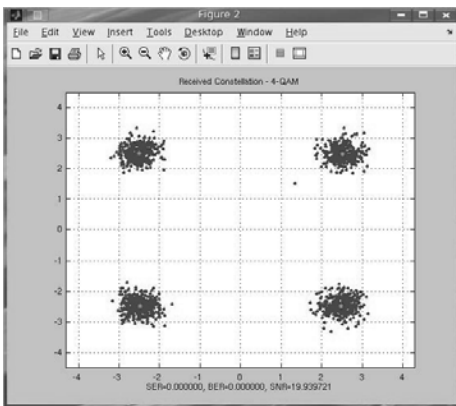


Figura 7: Constelación de símbolos correspondiente a las señales adquiridas.

(fuentes equiprobables o usando una función proporcionada por el usuario), el tipo de modulación (PAM, PSK o QAM), y el número de niveles de la modulación, el número de muestras para cada símbolo o la forma de pulso que se utiliza (rectangular, coseno alzado o definido por el usuario). La figura 5 muestra la ventana principal de gtTAL, usada para introducir estos parámetros. Los estudiantes pueden medir el rendimiento de varios algoritmos de estimación de canal y compararlo con el rendimiento obtenido cuando se dispone en el receptor del conocimiento perfecto del canal.

Por ejemplo, los parámetros introducidos en la ventana principal de gtTAL mostrados en la figura 5 permiten generar $k = 2000$ bits equiprobables mapeados mediante el código Gray. Los símbolos son 4-QAM, filtrados usando un pulso de coseno alzado, y, por último, los símbolos son codificados en espacio-tiempo con el esquema Alamouti. La pantalla principal de gtTAL muestra la constelación de la modulación y la forma del pulso.

Después de pulsar el botón "Transmit", la secuencia se transmite por el banco de pruebas. La figura 6 muestra la señal transmitida s_1 (se utiliza otra fi-

gura para s_2), los valores de los símbolos transmitidos, la señal transmitida después del conformador y las señales adquiridas después del filtro adaptado. La figura 7 muestra la constelación de símbolos recibidos. En la parte inferior de la figura se muestran, los valores estimados para la tasa de error de símbolo (SER), la proporción de error de bit (BER) y la relación señal-ruido (SNR).

4. Conclusiones

En esta ponencia se ha presentado una experiencia en el desarrollo de un sistema de “aprendizaje asistido por banco de pruebas” que está formado por tres módulos: banco de pruebas, arquitectura multi-capas y herramienta gráfica (gtTAL). El módulo gtTAL puede ser utilizado como una interfaz gráfica de usuario convencional para llevar a cabo simulaciones por ordenador, es decir, sin necesidad de ningún componente hardware adicional. Sin embargo, su principal potencial reside en la capacidad de transmitir y adquirir señales mediante un banco de pruebas hardware. Para ello, gtTAL se integra perfectamente con una arquitectura software distribuida multi-capas, lo que permite acceder al banco de pruebas hardware sin programar a bajo nivel. En este sentido, gtTAL representa una herramienta de alto nivel para operar con un banco de pruebas hardware. Así, incluso los estudiantes de pregrado pueden experimentar fácilmente con transmisiones de datos reales en entornos reales.

Aunque sería interesante estudiar la influencia que ha tenido el uso de la herramienta en el rendimiento académico de los estudiantes, en este momento no disponemos de una muestra estadística representativa debido a la reciente implantación de la herramienta. Lo que sí hemos detectado, por comentarios de los alumnos, es su satisfacción al poder experimentar con un sistema real.

Agradecimientos

Este trabajo ha sido parcialmente financiado por Xunta de Galicia, Ministerio de Ciencia e Innovación de España y fondos FEDER (09TIC008105PR, TEC2007-68020-C04-01, CSD2008-00010)

Referencias

- [1] Alamouti, S. “A simple transmit diversity technique for wireless communication”. *IEEE Journal on Selected Areas in Communications*, 16(8). pp. 1451-1458, 1998
- [2] Beres, E. y Adve, R. “Blind channel estimation for orthogonal stbc in miso systems”. *IEEE Transactions on Vehicular Technology*, 56(4). pp. 2042-2050, 2007.
- [3] Bingham, E. y Hyvärinen, A. “A fast fixed-point algorithm for independent component analysis of complex valued signals”. *International Journal of Neural Sys- tems*, 10. pp. 1-8, 2000.
- [4] Borkowski, D., Brühl, L., Degen, C., Keusgen, W., Alirezai, G., Geschewski, F., Oikonomopoulos, C., y Rembold, B. “SABA: A testbed for real-time MIMO system”. *EURASIP Journal on Applied Signal Processing*, 2006.
- [5] Budianu, C. and Tong, L. “Channel estimation for space-time orthogonal block codes”. *IEEE International Conference on Communications, ICC 2001*, 4, pp. 1127-1131, 2001.
- [6] Caban, S., Mehlführer, C., Langwieser, R., Scholtz, A. L., y Rupp, M. “Vienna MIMO Testbed”. *EURASIP Journal on Applied Signal Processing*, Artículo ID 54868. 2006.
- [7] Cardoso, J. y Soughoumias A., “Blind beamforming for non-gaussian signals”. *IEE Proceedings F Radar and Signal Processing*, 140(6), pp. 362-370. 1993.
- [8] García-Naya, J. A., Fernández-Caramés, T. M., Pérez-Iglesias, H., González-López, M., Castedo, L., “A flexible MIMO testbed”, *Proc. CMCS*, 2007.
- [9] Haykin, S. *Adaptive Filter Theory*. Prentice Hall Information and System Sciences Series, 4ta edición. 2001.
- [10] Rao, R., Zhu, W., Lang, S., Oberli, C., Browne, D., Bhatia, J., Frigon, J.-F., Wang, J., Gupta, P., Lee, H., Liu, D., Wong, S., Fitz, M., Daneshrad, B., y Takeshita, O. “Multi-antenna testbeds for research and education in wireless communications”. *IEEE Communications Magazine*, 42(12). pp. 72-81, 2004.
- [11] Rupp, M., Caban, S., and Mehlführer, C. “Challenges in building MIMO testbeds”. *13th European Signal Processing Conference (EUSIPCO'07)*, Polonia, 2007.

- [12] Tarokh V., Jafarkhani H. y Calderbank. A.R. "Space-time block codes from orthogonal designs?". *IEEE Transactions on Information Theory*, 45(5), pp. 1456-1467, 1999.