

METODOLOGÍA PARA IMPLEMENTACIÓN DE ALGORITMOS SOBRE DSP ORIENTADOS AL PROCESO DE BIOSEÑALES

RESUMEN

Se presenta la implementación de algoritmos, empleando técnicas de procesamiento digital de señales, en la estimación de parámetros sobre bioseñales, para dos casos particulares: obtención de frecuencia fundamental (*pitch*) en señales de voz, y detección de complejos QRS en señales electrocardiográficas. Dichos algoritmos, se orientan hacia la ejecución final en una plataforma de hardware (*DSP*), con base en el desarrollo de una metodología de diseño en tiempo real, que genera resultados satisfactorios, en cuanto a tiempos de proceso y valores calculados respecto a modelos de referencia, complementables por procedimientos de optimización, que maximicen el aprovechamiento para recursos potenciales del sistema.

PALABRAS CLAVES: Algoritmo, DSP, Metodología, Pitch, Proceso, QRS, Tiempo real.

ABSTRACT

The implementation of algorithms using digital signal processing techniques, is presented over the estimation of parameters on biosignals, for two particular cases: fundamental frequency (pitch) in voice signals, and QRS complex detection in electrocardiography. Presented algorithms are oriented to the final execution on a hardware platform (DSP), based in the development of a real time design methodology, that give a good performance in terms of process time and results computed with respect to reference models, followed by optimization procedures that yield to take advantage on the potential resources of the system.

KEYWORDS: Algorithm, DSP, Methodology, Pitch, Process, QRS, Real time.

1. INTRODUCCIÓN

Los sistemas en tiempo real (STR), se asocian generalmente con aplicaciones donde el número de procesadores que interactúan, al igual que la cantidad de restricciones impuestas, están en continuo crecimiento [1]. Por tanto, para dichos casos, es importante generar un equilibrio entre tareas realizadas, tiempos de proceso y resultados obtenidos, de manera que se garantice una eficiente operación de conjunto. Adicionalmente, en las últimas décadas, los dispositivos digitales con amplia escala de integración, se vienen convirtiendo en soluciones óptimas y de costo reducido, para aplicaciones con cierta envergadura [2]. Razón por la cual, fabricantes como: Texas Instruments, Motorola y Analog Devices, entre otros, han desarrollado soluciones a nivel de procesadores digitales de señal (DSP), que permiten la ejecución para tareas de cálculo complejas, con alta relación entre velocidades de proceso y capacidades de memoria, ajustables a STR. De otro lado, las técnicas de análisis automatizado, cobran cada vez más fuerza dentro del campo de la medicina aplicada, debido principalmente a su capacidad para efectuar valoración clínica, optimizada en términos de reducción para tiempos de diagnóstico, y disminución de errores, característicos en procedimientos subjetivos [3].

Con respecto a este último punto, se presenta la aplicación para técnicas de procesamiento digital sobre bioseñales, a partir de una metodología enfocada hacia la implementación en un dispositivo de hardware (DSP), para el caso particular de procedimientos de: cálculo de frecuencia fundamental (*pitch*) [4], en señales de voz y detección de complejos QRS [5], en señales electrocardiográficas, tal y como se describe, en las secciones posteriores¹.

2. ESTRUCTURAS EN TIEMPO REAL

Un sistema que interactúa con el ambiente en instantes precisos, y cuyo tiempo de procesamiento sea menor o igual a la duración de una ventana de análisis (apertura), se denomina sistema en tiempo real (STR) [6]. Dichos sistemas, se utilizan en procesos que requieren un estricto cumplimiento de tiempos, alta confiabilidad, manejo intenso de datos, al igual que alto grado de predictibilidad, y adaptabilidad. Los STR se han integrado a una amplia variedad de aplicaciones que incluyen: sistemas médicos, de control, de manufactura, de robótica y de multimedia, entre otros, adquiriendo

LUIS ENRIQUE AVENDAÑO

Ingeniero Electrónico, Ms.C
Profesor Titular
Universidad Tecnológica de Pereira
leavenda@utp.edu.co

GERMÁN CASTELLANOS

Ingeniero Electrónico, Ph.D.
Docente de Planta
Universidad Nacional de Colombia
– sede Manizales
gcastell@ieee.org

RICARDO ALZATE

Ingeniero Electrónico.
Universidad Nacional de Colombia
– sede Manizales
ricardoalz@unalmlz.edu.co

ANDRES MORALES

Ingeniero Electrónico.
Universidad Nacional de Colombia
– sede Manizales
anmora32@hotmail.com

¹ Este trabajo es financiado por COLCIENCIAS según contrato 11191412867.

distintas formas, que van desde simples dispositivos de monitoreo y control, hasta sistemas altamente complejos y de características críticas [6]. Debido a esto, los STR demandan una alta confiabilidad, con resultados correctos, predecibles y a tiempo, incluso en presencia de fallos [7]. Luego, puede verse a un STR, como una entidad de cómputo y control, compuesta por unidades planificables (tareas), correspondientes a procedimientos que abarcan: algoritmos en control de procesos [8], ejecución de transacciones en bases de datos [9], o transmisión de paquetes en redes de comunicación [10], que al ejecutarse de manera concurrente, y tras cumplir con requerimientos estrictos para tiempos de respuesta, garantizan su correcta operación.

3. DISEÑO DE SISTEMA EN TIEMPO REAL

En la metodología de diseño para STR, existen una serie de pasos que deben ser tenidos en cuenta [11]:

- La primera parte del diseño de un STR consiste en identificar las restricciones (principalmente de tiempo) que deben ser satisfechas por el sistema.
- Se realiza un diagrama de bloques, indicando las partes funcionales que integrarán el STR.
- Selección de los componentes individuales, tomando en cuenta la arquitectura propuesta y respetando siempre las restricciones de tiempo que se imponen.
- Realización de un diagrama de conexiones físicas necesarias entre dispositivos, al igual que un conjunto de pruebas de escritorio (simulaciones, cálculo de tiempos internos y de respuesta, entre otras). Si las pruebas realizadas acreditan un comportamiento adecuado, se procede a la realización del prototipo, junto con pruebas reales, para conocer el desempeño final del sistema.
- Las pruebas reales (mediciones de voltaje, frecuencia, tiempos, entre otras) aplicadas al sistema bajo condiciones normales de operación, determinan si el sistema desarrollado cumple con las especificaciones requeridas.

Por tanto, se realiza la división modular, aplicada a los procedimientos de: estimación del pitch en señales de voz, y detección para complejos QRS en señales electrocardiográficas, con base en el desarrollo de una metodología para especificación de sistemas en tiempo real, a manera de etapas constitutivas fundamentales [1], que cubriendo los respectivos procesos de partición [2] y análisis (ó verificación [12]), permita adecuar el modelo obtenido, a una estructura de hardware disponible [13].

4. IMPLEMENTACIÓN DEL SISTEMA

Utilizando como referencia, el algoritmo descrito en [14], se realizó la ejecución sobre un procesador digital de señales [13], para los procedimientos de estimación del pitch en señales de voz, y detección de complejos QRS en señales electrocardiográficas, haciendo uso de la herramienta de software “Code Composer Studio” [15], creada para interactuar con tales dispositivos.

Dicha interfaz (figura 1), se desarrolla sobre una plataforma operativa *Windows*, al tiempo que utiliza como lenguaje de base, el conjunto de comandos *Ansi C*, permitiendo desarrollar aplicaciones en un entorno más amigable, que los convencionales, basados en lenguaje ensamblador [16].

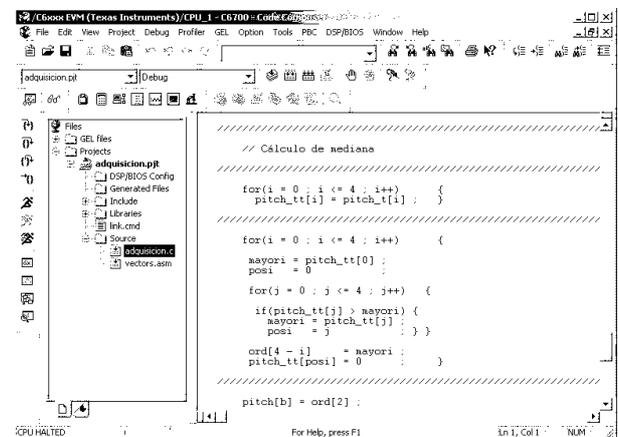


Figura 1. Ventana para ejecución de código.

El algoritmo implementado contiene la siguiente secuencia de etapas (figura 2):

- a. En primera instancia, es necesario adecuar el sistema para su posterior ejecución, a partir de procedimientos de inicialización y configuración, aplicados sobre aquellos elementos implicados en desarrollo del proceso, correspondientes a los módulos: codificador / decodificador de audio (*Codec*), y puerto serial multicanal (*McBSP*). Para terminar en un ciclo infinito, en espera de ser anticipado por procesos de mayor prioridad (*interrupciones*). Todo lo anterior incluido en una rutina principal, denominada *Main*.
- b. Cada vez que el registro de recepción de datos (*DRR*), activa un servicio de interrupción desde el *McBSP*, el ciclo infinito en *Main*, se interrumpe para dar paso a las instrucciones contenidas en la función *Hookint*, efectuando: inicialización en vector de interrupciones, mapeo entre la interrupción 15 del procesador utilizado y la petición del *DRR*,

activación de interrupción, y asignación para rutina de proceso (*McBSPcvISR*).

- c. Por último, cuando la petición de interrupción *Hookint*, asigna ejecución para el procedimiento *McBSPcvISR*, se realiza lectura del valor proveniente desde el puerto de entrada, equivalente a una muestra de señal actual, asignada posteriormente a la posición respectiva en el buffer de datos, que luego de alcanzar la cantidad de muestras considerada por ventana (128), permite ejecución para la rutina de cálculo, contenida en la función *parameter*. Dicha función, realiza estimación para el parámetro en consideración, utilizando alguno de los métodos descritos en la sección 5.

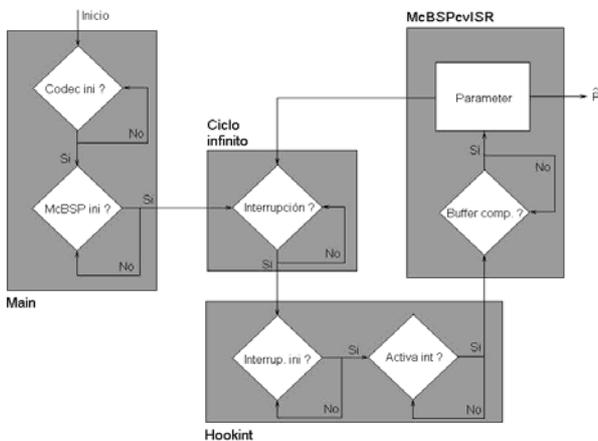


Figura 2. Algoritmo implementado.

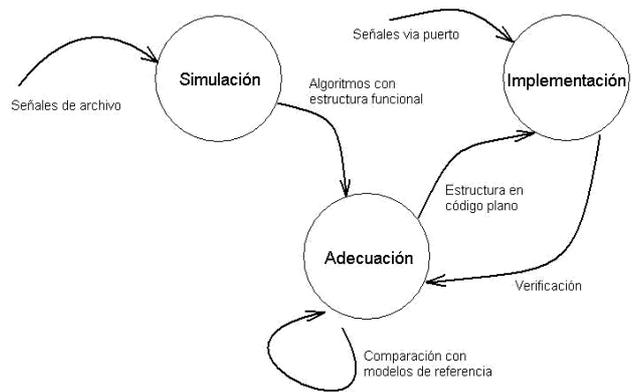
5. DESARROLLO DE ALGORITMOS

La ejecución para los algoritmos implementados, se obtuvo a partir de las siguientes fases (figura 3):

- Realización de códigos en un entorno de programación simulado (empleando Matlab), aplicando señales de prueba, correspondientes a: fonemas vocálicos segmentados pertenecientes a la base de datos MIRLA (propiedad de la Universidad Nacional de Colombia, sede Manizales), para el caso de análisis de pitch, al tiempo que señales electrocardiográficas contenidas en la base de datos del MIT, para obtención de complejos QRS.
- Realización de comparación, en términos de desempeño, sobre resultados obtenidos por sistemas de referencia: algoritmo Childers [17], como método de validación respecto a contornos del pitch, al igual que criterios de evaluación recomendados por la AAMI (Association for the Advancement of Medical

Instrumentation [18]), en el caso de patrones para ECG.

- Implementación de algoritmos, en código orientado al procesador de DSP disponible (referencia TMS320C6701 [19], [15]).
- Adquisición de señales analógicas vía puertos, para crear un intercambio de datos entre las tarjetas de



procesador (DSP), y de audio (PC).

Figura 3. Diagrama de flujo de datos (DFG), para metodología.

5.1 Algoritmo para cálculo del pitch

Haciendo uso de un método basado en análisis de dominio frecuencial (*HPS - Harmonic Product Spectrum* [20]), se desarrolla estimación para la frecuencia fundamental en señales de voz [4], a partir de los siguientes pasos (figura 4):

- Ventaneo tipo Hanning, con traslape de 64 muestras, en el segmento de señal adquirido (cuya apertura es de 128 datos), para reducir sensibilidad a los cambios presentados en extremos de ventana.
- Posteriormente, se aplica transformada rápida de Fourier (FFT), con resolución de 2048 puntos, sobre dicho segmento.
- Luego, se define un vector de posiciones para candidatos a frecuencias sub-armónicas, conformado por aquellos valores superiores al 30% del mayor armónico obtenido.
- Finalmente, se ejecuta un juicio de decisión, respecto a las distancias relativas entre posiciones para tales armónicos, de cuyo máximo común divisor, se obtiene el valor de periodo fundamental.

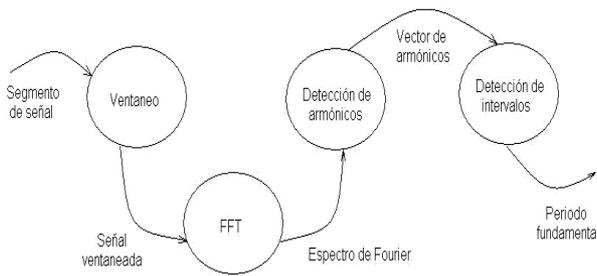


Figura 4. Algoritmo para cálculo del pitch.

5.2 Algoritmo para detección de complejos QRS

Análogamente, el procedimiento para detección de complejos *QRS* en señales electrocardiográfica [5], se obtuvo a partir de (figura 5):

- Realización de submuestreo (reducción de 10 kHz a 125 Hz), sobre el segmento de entrada (ventana de 128 datos).
- Obtención para coeficientes de aproximación y de detalle, aplicando transformada Wavelet (DTW), a partir de un procedimiento de filtrado digital en cuadratura (paso alto / paso bajo).
- Búsqueda del mayor valor, presente en el conjunto de datos aplicados como entrada.
- Búsqueda de máximos relativos al mayor (con criterio del 80%), y de cruces por cero, sobre los datos generados con transformada Wavelet.
- Comparación entre máximos posibles y cruces por cero, para detección de la onda R.
- Detección para la onda Q, partiendo de la posición para la onda R confirmada, y retrocediendo hasta hallar el mínimo valor en un rango definido (100 ms). Si es necesario, se busca en los datos de la ventana anterior.
- Detección para la onda S, partiendo de la posición para la onda R confirmada, y avanzando hasta encontrar el mínimo en un rango definido (100 ms). Si es necesario, la búsqueda se pospone hasta la próxima ventana de datos.
- Cálculo de la frecuencia cardiaca, con base en la distancia entre las posiciones para ondas R actual y anterior.

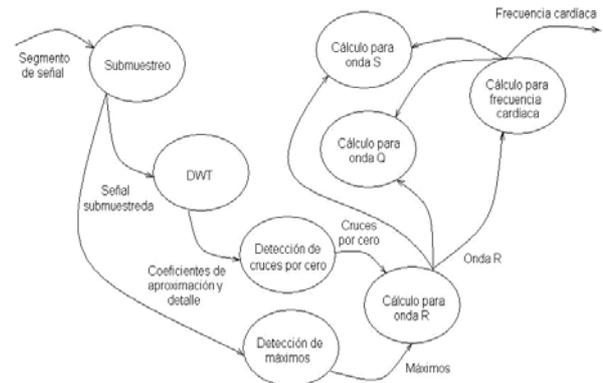


Figura 5. Algoritmo para detección de complejo QRS.

6. RESULTADOS

La implementación a nivel de hardware (DSP), para los procedimientos anteriores, permitió obtener lo siguiente:

6.1 Cálculo del pitch

El algoritmo para detección de la frecuencia fundamental en señales de voz, genera una envolvente con valor promedio muy cercano al patrón de referencia, como se observa en la figura 6, tomada para un segmento de voz equivalente al fonema 'a', con valor promedio de 227 Hz y tasa de muestreo de 10kHz. Resultados adicionales, respecto a valores promedio obtenidos sobre otras señales de prueba, se visualizan en la tabla 1.

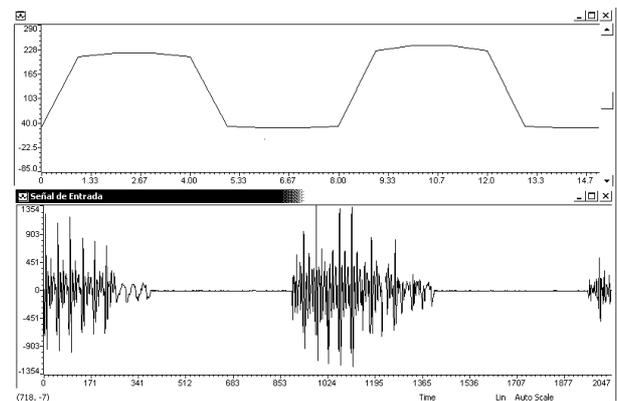


Figura 6. Señal de entrada y contorno respectivo en DSP.

Análogamente, se cita información acerca de indicadores de desempeño para el sistema en tiempo real, con base en especificaciones técnicas agrupadas en la tabla 2.

Tabla 1. Error porcentual para diferentes señales de prueba.

Señal Referencia Praat (Hz)	Promedio medido (Hz)	Error porcentual (%)
227	220	-3,08
114	126	10,52
171	186	8,77
210	200	-4,76
233	245	5,15
199	196	-1,5
194	201	-3,6
144	159	10,41
140	131	-6,42

Tono de calibración (Hz)	Promedio medido (Hz)	Error porcentual (%)
100	103	3
125	131	4,8
250	253	1,2
300	304	1,3
500	501	0,2
1000	1003	0,3
2000	2001	0,05

Tabla 2. Indicadores de desempeño para sistema implementado.

Criterio	Valor
Tiempo de ciclo por instrucción	40 ns
Duración promedio para rutina de proceso	435 ms
Longitud promedio para rutina de proceso	6884 Bytes
Duración promedio para rutina de adquisición	6.52 us
Longitud promedio para rutina de adquisición	2160 Bytes
Nivel de optimización	para máxima velocidad

6.2 Detección de complejos QRS

Utilizando criterios de evaluación AAMI, correspondientes a: *TP* (verdaderos positivos), *FP* (falsos positivos), y *FN* (falsos negativos), se determinan mediciones para rendimiento del detector, en términos de: *+P* (valor predictivo positivo), y *S* (sensibilidad), aplicados al caso de una señal de prueba, correspondiente a un electrocardiograma (ECG) constante, con 1200 complejos QRS, tal y como se observa en la tabla 3.

En la misma medida, se obtuvo información respecto a variabilidad del algoritmo para detección de valores de frecuencia cardíaca, validados con un dispositivo de medición (escopómetro), y cuyos resultados se agrupan en la tabla 4.

Tabla 3. Criterios de evaluación AAMI.

Criterio	Valor
TP	1184
FP	0
FN	16
P	1
S	0,986

Tabla 4. Desempeño de algoritmo señal ECG constante (MIT).

Medida	Distancia Q-R (seg)	Distancia R-S (seg)	Distancia R-R (seg)
Varianza	2.2298 * 10e-5	2,2341 * 10e-5	2,4246 * 10e-2
Media	0.0187	0,0374	0,966
Referencia	0.0226	0,036	1

Finalmente, tras utilizar una señal ECG normal obtenida de la base de datos QT [21], en la cual, habían sido marcadas con anterioridad (por un grupo de expertos), las posiciones para las ondas Q, R y S, se efectuó una comparación, entre estas posiciones y las generadas en el procesador digital, obteniendo sobre 2400 complejos QRS analizados, los errores promedio que se muestran en la tabla 5.

Tabla 5. Desempeño de algoritmo señal ECG constante (QT).

Medida	Distancia Q-R (seg)	Distancia R-S (seg)	Distancia R-R (seg)
Error promedio	0,0016	0,0018	0,0021

7. CONCLUSIONES

Con base en los resultados obtenidos, puede concluirse que:

- La metodología propuesta, permite ejecutar de manera directa y eficiente, la estimación en línea de parámetros requeridos en las señales analizadas, facilitando implementación de tareas en tiempo real, con base en el uso de herramientas tanto de hardware, como de software, especializadas para tal fin.
- En la misma medida, se obtuvo una respuesta satisfactoria en cuanto a valores calculados respecto a los modelos de referencia.
- Respecto a velocidades de proceso, se resalta el hecho, que para el caso de procesamiento en señales electrocardiográficas, los tiempos de respuesta se

hagan menos críticos que en su contraparte de voz, debido a la naturaleza inherente de la señal manipulada (ancho de banda), y que deriva en relativa facilidad para cumplir con cuotas de proceso, evitando pérdida de muestras. Razón por la cual, se hace necesario un procedimiento de optimización, que permita maximizar la capacidad de aprovechamiento sobre los recursos del sistema.

- Luego, puede considerarse a este, como un primer paso dentro de la realización de sistemas más robustos, que incluyan conjuntos ampliados de características, enfocadas hacia la ejecución de tareas particulares aplicadas en análisis automatizado de bioseñales.

8. BIBLIOGRAFÍA

- [1]. M. Joseph, "Real-Time Systems: Specification, Verification, and Analysis". p 98 - 122. Prentice Hall. New York. 1996.
- [2]. A. Jhon, "Real-Time Signal Processing: The design and implementation of signal processing systems". p 151 - 240. Prentice Hall. New York. 1999.
- [3]. C. G. Omar Danilo, "*Identificación de la Voz Normal y Disfuncional, a partir de su Análisis Acústico Objetivo*". Departamento de Ingeniería Electrónica. Universidad Politécnica de Valencia. Valencia. 2002.
- [4]. A. Ricardo, C. Germán, "*Estimación de contornos del pitch en línea sobre DSP*". Universidad Nacional de Colombia - Sede Manizales. 2003.
- [5]. M. Andrés, C. Germán, A. Luis E., "*Segmentación de ECG normal con Wavelets en tiempo real*". Universidad Nacional de Colombia - Sede Manizales. 2003.
- [6]. P. D. Lawrence, K. Maunch, "Real-Time Microcomputer System Design". McGraw Hill, New York, 1987.
- [7]. J. Lala, R. Harper, "*Architectural Principles for Safety-Critical Real-Time Application*". Proceedings of the IEEE, 1994.
- [8]. K. G. Shin, H. Kim, "*Derivation and application of hard deadlines for Real-Time Control Systems*". IEEE Transactions on Systems, Manufacturing, and Cybernetics, Vol 22, November 1992.
- [9]. K. G. Shin, P. Ramanathan, "*Real-Time Computing: A New Discipline of Computer Science and Engineering*". Proceedings of the IEEE, Vol 82, 1994.
- [10]. J. Stankovic, K. Ramanritham, "*Tutorial: Hard Real-Time Systems*". IEEE Computer Society, 1988.
- [11]. A. L. Chau, J. J. Medel, P. Guevara, "*Sistemas Dedicados de Tiempo-Real*". <http://aleph.cs.buap.mx>.
- [12]. B. Stuart, "Real-Time Computer Control: An introduction". p 129 - 172. Prentice Hall. New York. 1988.
- [13]. Texas Instruments, "*TMS320C6201/6701 Evaluation Module Technical Reference*".
- [14]. K. Hoover, "*TMS320C62x/C67x C Familiarization and Audio Sampling*". Rose-Hulman Institute of Technology.
- [15]. Texas Instruments, "*Code Composer Studio Getting Started Guide*".
- [16]. N. Seshan, "*DSPs vs. FPGAs in Signal Processing System Design*". Texas Instruments.
- [17]. C. G. Donald, "Speech Processing and Synthesis Toolboxes". p 320-324. John Wiley & Sons, Inc. New York. USA. 2000.
- [18]. AAMI, "*Association for the Advancement of Medical Instrumentation*". <http://www.aami.org>.
- [19]. Texas Instruments, "*TMS320C6201/6701 Evaluation Module User's Guide*".
- [20]. Q. Holger, S. Olaf, S. Manfred R., "*Robust pitch Tracking in the Car Environment*". Drittes Physikalisches Institute, Universität Göttingen and Robert Bosch Research & Development Daimler Chrysler Research and Technology AT&T Bell Labs.
- [21]. Physionet, "*The QT Database*". <http://www.physionet.org/physiobank/database/qtddb/doc/node1.html>.