



## System Architecture of an Experimental Synthetic Aperture Real-Time Ultrasound System

**Jensen, Jørgen Arendt; Hansen, Martin; Tomov, Borislav Gueorguiev; Nikolov, Svetoslav; Holten-Lund, Hans**

*Published in:*  
IEEE Ultrasonics Symposium

*Link to article, DOI:*  
[10.1109/ULTSYM.2007.165](https://doi.org/10.1109/ULTSYM.2007.165)

*Publication date:*  
2007

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Jensen, J. A., Hansen, M., Tomov, B. G., Nikolov, S., & Holten-Lund, H. (2007). System Architecture of an Experimental Synthetic Aperture Real-Time Ultrasound System. In *IEEE Ultrasonics Symposium* (pp. 636-640). IEEE. <https://doi.org/10.1109/ULTSYM.2007.165>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# System Architecture of an Experimental Synthetic Aperture Real-time Ultrasound System

Jørgen Arendt Jensen<sup>1</sup>, Martin Hansen<sup>2</sup>, Borislav Georgiev Tomov<sup>1</sup>,  
Svetoslav Ivanov Nikolov<sup>1</sup> and Hans Holten-Lund<sup>2</sup>

<sup>1</sup>Ørsted•DTU, Build. 348, Technical University of Denmark, DK-2800 Lyngby, Denmark,

<sup>2</sup>Prevas A/S, Frederikskaj 6, DK-2450 Copenhagen SV

**Abstract**—Synthetic Aperture (SA) ultrasound imaging has many advantages in terms of flexibility and accuracy. One of the major drawbacks is, however, that no system exists, which can implement SA imaging in real time due to the very high number of calculations amounting to roughly 1 billion complex focused samples per second per receive channel. Real time imaging is a key aspect in ultrasound, and to truly demonstrate the many advantages of SA imaging, a system usable in the clinic should be made. The paper describes a system capable of real time SA B-mode and vector flow imaging. The Synthetic Aperture Real-time Ultrasound System (SARUS) has been developed through the last 2 years and can perform real time SA imaging and storage of RF channel data for multiple seconds. SARUS consists of a 1024 channel analog front-end and 64 identical digital boards. Each has 16 transmit channels and 16 receive channels both with a sampling frequency of 70 MHz/12 bits for arbitrary waveform emission and reception. The board holds five Virtex 4FX100 FPGAs, where one houses a PowerPC CPU used for control. The remaining four are used for generation of transmit signals, receive storage and matched filter processing, and focusing and summing of data. Each FPGA can perform 80 billion multiplications/s and the full system can perform 25,600 billion multiplications/s. The FPGAs are connected through multiple 3.2 Gbit Rocket IO links, which makes it possible to send more than 1.6 GBytes/s of data between the FPGAs and between boards. The system can concurrently sample in 1024 channels, thus, generating 140 GBytes/s of data, which also can be processed in real time or stored. The system is controlled over a 1 Gbit/s Ethernet link to each digital board that runs Linux. The control and processing are divided into functional units that are accessed through an IP numbering scheme in a hierarchical order. A single controlling mechanism can, thus, be used to access the whole system from any PC. It is also possible for the controlling PowerPCs to access all other boards, which enables advanced adaptive imaging. The software is written in C++ and runs under Matlab for high level access to the system in a command structure similar to the Field II simulation program. This makes it possible for the user to specify imaging in very few lines of code and the set-up is fast due to the employment of the 64 PowerPCs in parallel. Focusing is done using a parametric beam former. Code synthesized for a Xilinx V4FX100 speed grade 11 FPGA can operate at a maximum clock frequency of 167.8 MHz producing 1 billion I and Q samples/second sufficient for real time SA imaging. The system is currently in production, and all boards have been laid out. VHDL and C++ code for the control has been written and the code for real time beamformation has been made and has obtained a sufficient performance for real-time imaging.

## I. INTRODUCTION

In Synthetic Aperture (SA) imaging data are acquired by sending a spherical wave from part of the aperture to insonify the whole region of interest. The received signals on all

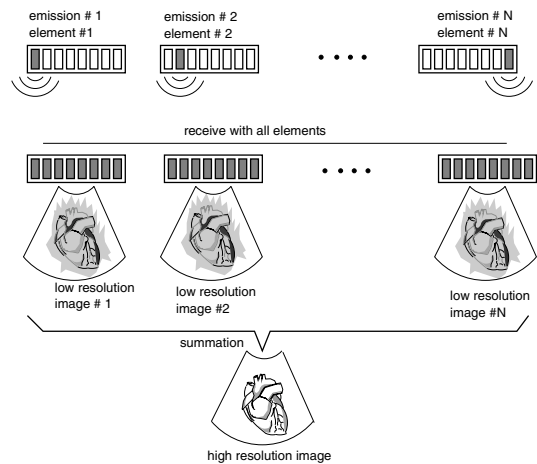


Fig. 1. Acquisition of a synthetic aperture image (from [11]).

transducer elements are then sampled and the process is repeated for a number of transmitting source position. The image can then be dynamically focused in both transmit and receive, thus, yielding the best possible resolution attainable for delay-and-sum beamforming [1], [2], [3], [4].

SA ultrasound imaging has the potential of increasing the resolution and contrast of medical images as has been shown in pre-clinical trials [5]. It has also been shown that problems with penetration [6], [7], [8] and tissue motion can be solved [9], [10]. The major obstacle to implementing SA imaging is, thus, to solve the problem with the high number of calculations to be performed per second. This paper describes the principles behind a research scanner capable of making real-time 2D SA imaging.

## II. SYSTEM DEMANDS

The purpose of the scanner is to acquire and process data for studying advanced ultrasound imaging. This encompasses both conventional B-mode and velocity images in two and three dimensions as well as advanced coded synthetic aperture images for anatomic and vector flow imaging. By far the most demanding application is SA imaging, and the basic demand for data acquisition and processing can therefore be based on this application.

In SA imaging a full low-resolution image is made for every pulse emission as shown in Fig. 1. Typically images

will consist of  $N_l = 200$  lines each containing  $N_s = 1024$  complex samples. Imaging to a depth  $D$  of 15 cm gives a pulse repetition frequency  $f_{prf}$  of 5 kHz. The number of samples to beamform  $N_{bs}$  for each receiving element is then

$$N_{bs} = f_{prf} N_l N_s \quad (1)$$

which for this example amounts to  $1.024 \cdot 10^9$  beamformed complex samples per second. The number of receiving elements range from 64 to 256 in modern scanners, thus, yielding up to  $262.4 \cdot 10^9$  beamformed complex samples per second. This should be compared to the  $1.31 \cdot 10^9$  samples per second in a conventional scanner. The large increase in the number of calculations is due to the beamformation of a full image after each emission.

The calculation demand is also accompanied by a similar requirement on data bandwidth. The amount of data is increased proportionally to  $N_l$  and typically the input data rate to the beamformer, when sampling at 40 MHz, is 20 Gbytes/s and the output rate is 1049 Gbytes/s assuming 4 bytes are used for each complex sample. The low resolution images from the beamformer are summed to yield the final high resolution image. Using an emission sequence of 100 emission will give 50 images a second and the output data rate is here a modest 41 Mbytes/s.

Much of the processing can be done in parallel, but it is crucial that the high resolution image should at least be partly summed within the processing unit to avoid the very high output demand.

Synthetic aperture can also be used for velocity [12] and vector flow imaging [13], [14]. The demands are higher than for anatomic imaging, but they can be scaled sufficiently down to comply with the demands mentioned above, which therefore form the basis for the system design.

The system should be capable of sampling the bandwidth of high-end transducer with a high precision. Currently the limit is around 15 MHz, but it will be increased by cMUT probes. The sampling frequency has, thus, been set at 70 MHz with a precision of 12 bits. This precision matches the signal-to-noise ratio attainable in the current, commercially available TGC chips.

Both array and matrix transducers should be handled. Currently most commercial scanners use 192 element arrays and 3D scanning can be performed with  $32 \times 32 = 1024$  elements often in a sparse configuration. Sampling in 1024 channels should therefore be possible.

A key feature of a research scanner is the capability of storing raw data sets, so that it is possible to study data from the individual transducer elements for every emission. This has e.g. been used in the RASMUS scanner [15] to develop many imaging methods and to perform smaller pre-clinical trials for SA imaging [5] and vector flow imaging [16]. The data storage should be over several cardiac cycles and, thus, a minimum of 3 seconds must be sampled giving a demand for RAM of  $1024 \cdot 70 \cdot 10^6 \cdot 2 \cdot 3 = 400$  Gbytes. The capability of selecting to sample only part of the data can extend the sampling time, so the start and end depth and the lines to sample should

be settable. Also decimation of the data before storage in RAM should be implemented. Fast parallel data storage is also needed. All the data should be stored on disc in less than a minute, if the system should be used for larger clinical studies. A data storage rate of roughly 10 Gbytes/second should, thus, be the target.

The system must be capable of acquiring different imaging modes interleaved at the same time to allow for side by side comparison as done in [5]. The transmitted signal must, thus, be changed from emission to emission and from element to element. The curves should be able to have arbitrary shape and should include long chirp signals [17], [18] as well as conventional sinusoidal signals. Again a precision of 12 bits should be used together with emission focusing and apodization. The TGC curve should also be set on a per emission basis to compensate for the differences in received energy for different emission signals.

After the data is acquired it should be processed in real time. This includes data decimation, matched filtration, beamforming, summation, estimation, and presentation. The matched filter should be selectable based on the emission and can be changed for the different imaging methods.

### III. SYSTEM ARCHITECTURE

The system consist of an analog front-end with 1024 channels and a digital board for transmit generation, sampling, data storage, and real-time processing.

The front-end is based on a commercial scanner and has a linear transmitter, which can amplify arbitrary transmitted signal. The received signal is sent through a TGC amplifier to the digital board.

The central core of the Synthetic Aperture Real-time ultrasound system (SARUS) is the DAUP (Digital Acquisition and Ultrasound Processing) board. Fig. 2 shows a block diagram of the DAUP board. Each board receives and transmits 16 balanced analog signals. The signals go through a number of processing devices that also can communicate with neighboring boards in the rack. This board is the heart of the system and relies on 5 FPGA's to do the processing. The main functions are:

- **Acquisition** The board contains sixteen 12-bits, 70 MHz AD/DA converters and associated control electronics to interface to the analog world. FPGA (2) in Fig. 2 controls the 16 transmit channels, where the emitted signals are taken from the external memory and delayed and apodized before transmission. The signals can have arbitrary shape that change for every channel and emission. FPGA (1) in Fig. 2 controls the 16 receive channels, which can be sampled and captured in external RAM for later disc storage and external processing. This FPGA also houses decimation processing and matched filtration.
- **Signal processing** The focusing and beamformation is performed in FPGA (3) and (4). The processing chain is described in Section V.
- **Communication** The DAUP boards are daisy-chained, with each board doing part of the total processing. Rapid

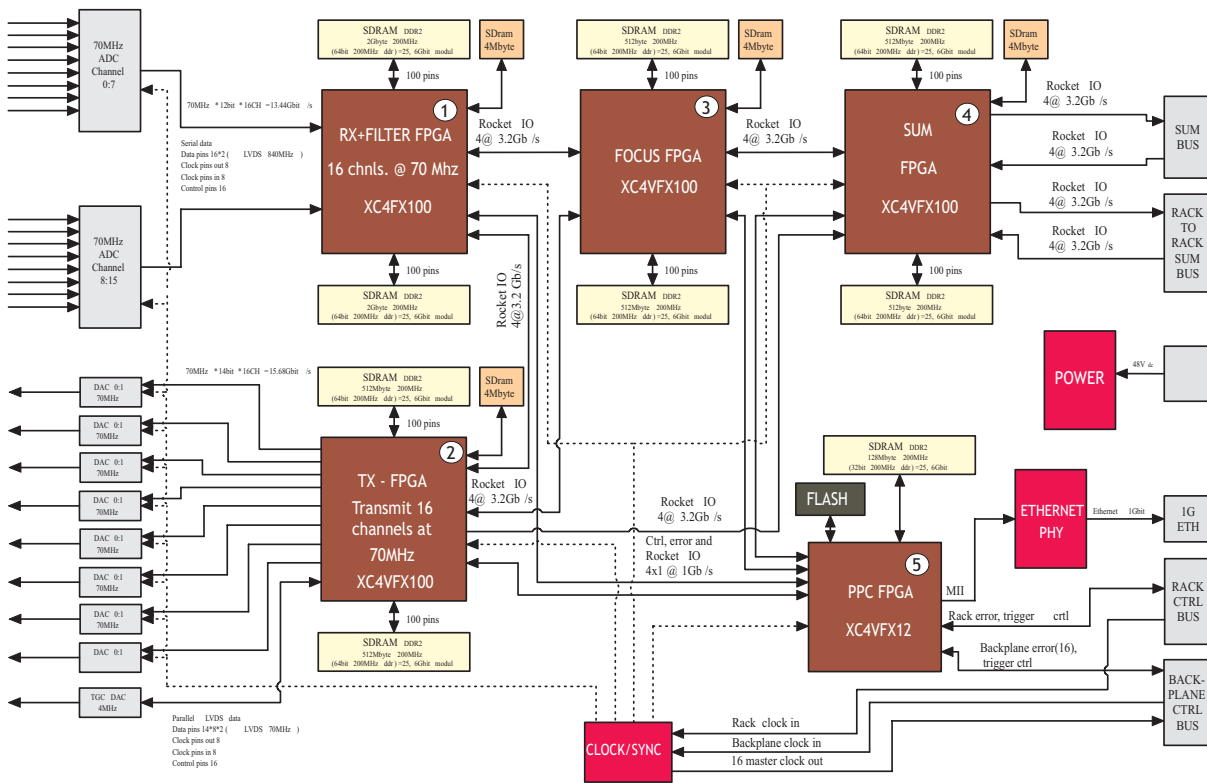


Fig. 2. Block diagram of the DAUP board.

IO communication interfaces to the neighboring cards exist and can send summed and beamformed signal at a rate of more than 12 Gbits/s. Communication between the processing FPGAs can take place at the same rate. Further, a 1 Gbit/s Ethernet connection exists for each board for control of the FPGAs and fast data storage.

- **Synchronization** The sampling has to be precisely synchronized across all 1024 channels for the data to be aligned for later focusing. The sampling clock and other signals are therefore distributed to all cards through the backplane in carefully matched cables and tracks.
- **Support functions** A central PowerPC processor to manage set-up and surveillance of each board is housed in one control FPGA (5). It is responsible for booting the system and runs Linux for a high level communication with the host PC and other boards through the Ethernet.

To simplify the design, it was decided to use the same basic block, based on an FPGA, for each sub-function. Each block is based on an FPGA and some RAM. The key components are:

**FPGA** Due to the high demands on communication speed, the Xilinx Virtex 4FX series was chosen. Based on the demand for processing power and pin-counts, the XC4VFX100-FF1152 device was chosen. It houses 160 signal processing units capable of making 500 million multiplications and additions per second each. The full system with 64 boards and 320 FPGA

can, thus, theoretically perform  $25.6 \cdot 10^{12}$  multiplications/second.

**RAM** Two types of RAM are used: A small fast SDRAM for FPGA parameters that cannot be fitted inside the FPGA and for which fast access is crucial, and some larger blocks of DRAM for sampling. For the DRAM, DDR2 type modules (standard PC types) are used. They have a good price/performance ratio and are suitable for this application. Each FPGA can have associated 4 GBytes of RAM for a total of 1024 Gbytes of RAM, which is sufficient to store more than 7 seconds of data from 1024 channels.

The complete system consist of 64 DAUP boards housed in two rack cabinets. The boards are connected to the analog front-end with 6 transducer plugs for up to 1024 channel transducers. All boards are interconnected through a 1 Gbits/s Ethernet switch. The SARUS system is also connected to a 16 CPU Linux cluster with 8 RAID drives for data storage.

#### IV. COMMUNICATION AND PROGRAMMING

The system houses 320 FPGAs each having numerous functional signal processing blocks inside. A uniform interface to them is, thus, of prime importance.

The communication to each functional unit is handled through the network through a hierarchical numbering scheme patterned after the IP numbering scheme. Each board has an IP address, each FPGA then has a number, and each functional unit with the FPGA has a number. The registers and RAM

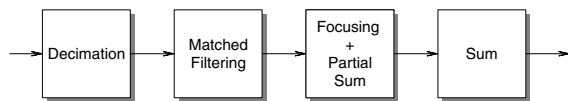


Fig. 3. Processing steps for SA B-mode imaging.

associated with the units also have assigned numbers. The individual registers can then be directly addressed via the Ethernet by any computer through the 1 Gbits/s links.

The registers and RAM for the functional units are defined in XML files. These are translated into VHDL code defining their interface and to C++ code for defining the access by the system library. The assignment of addresses is hereby dynamic and code for individual uses can be loaded and handled dynamically through the system library. The configuration control of all files is handled through the CVS version control system and CRC sums on the codes are used to verify the consistency of the loaded code. CVS also makes it possible to reconstruct any older version of the code used during a measurement. The global communication mechanism also makes it possible to perform adaptive imaging, since all units can communicate to all other units in the system.

The programming of the FPGAs is made in VHDL making the whole system easily configurable. Codes for the FPGAs are loaded by the controlling PowerPC on the main FPGA. The code can, thus, be changed from the controlling PC.

The overall interface to the system is written in C++ and Matlab. A server in C++ is running on each of the 64 PowerPCs and they handle writing of parameters to the individual functional units. The PowerPC receives global commands from a host PC running Matlab and a C++ program linked to Matlab. The commands are here written like in the previous RASMUS scanner [15] and are partly modeled after the Field II program [19]. This makes it possible to implement any kind of imaging in a few lines of Matlab code and then transfer a modest amount of parameters to the PowerPCs. They will then make the calculation of the actual hardware parameters in parallel, thus, making set-up of the system fast and also enabling adaptive imaging.

## V. REAL-TIME PROCESSING

The processing chain for SA B-mode imaging is shown in Fig. 3. After sampling at 70 MHz the individual element signals are decimated to match the sampling frequency to the bandwidth of the transducer. Often a significant data reduction can be attained without loss of information. A simple low pass filter is used in the decimation stage.

The element signals are then matched filtered to optimize signal to noise ratio and to compress e.g. chirp codes. This is implemented using an FFT based scheme. Up to 8192 samples can be input to the FFT with proper zero padding and a new matched filter can be multiplied onto the data for each emission. It can be selected to set the negative frequencies to zero to obtain the complex analytic signal spectrum and it can be shifted in the frequency domain for demodulation purposes.

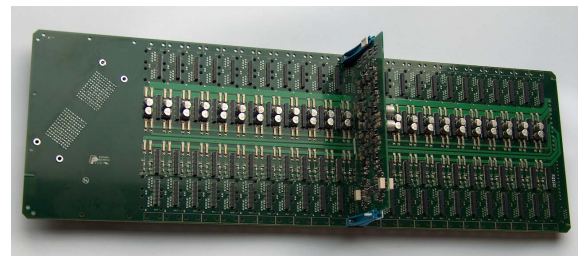


Fig. 5. Analog front-end used in SARUS.

An inverse FFT is performed and 4096 complex samples are output from the block to the focusing unit. This can be done for 4 channels on each DAUP board in real time.

The focusing unit is shown in Fig. 4. It handles four channels simultaneously. It has previously been described in [20] and [21]. It uses a parametric description of both the focusing and apodization, since focusing tables would be too large and demand a too high data bandwidth for the parameters. The delay and apodization values are calculated for the individual channels and common delay and apodization values for the transmitting element is added to this, to give the full delay and apodization. After look-up of the data, a linear interpolation is made and data for all channels are summed. This is repeated for all the emissions in the SA sequence and the data is summed to reveal the high resolution image. The process is then repeated for another set of lines in the image until all high resolution lines have been formed. The final data is transferred to the sum process and is summed coherently through all the 64 boards to yield the final high resolution image. A single beamforming unit can in the Virtex 4 chip process 167.8 million beamformed samples per second for 4 channels. The chips are large enough to house 6 beamforming units, thus, obtaining the performance goal of one billion complex beamformed samples per second.

## VI. CONCLUSION

Currently the first prototype of the system is under fabrication. The analog backplane and one amplifier module with 8 channels is shown in Fig. 5. Such a backplane holds 192 channels and 6 of these boards are placed in the system.

The control VHDL has been made and an initial prototype of the processing VHDL has been made. The latter has been synthesized to verify the performance goal of the system. The SARUS scanner should, thus, be capable of real time SA imaging and can be used for both storing complete 3D RF data sets and for real time processing for clinical studies.

## ACKNOWLEDGMENT

This work was supported by grant 26-04-0024 from the Danish Science Foundation and by B-K Medical Aps.

## REFERENCES

- [1] D. K. Peterson and G. S. Kino. Real-time digital image reconstruction: A description of imaging hardware and an analysis of quantization errors. *IEEE Trans. Son. Ultrason.*, 31:337–351, 1984.

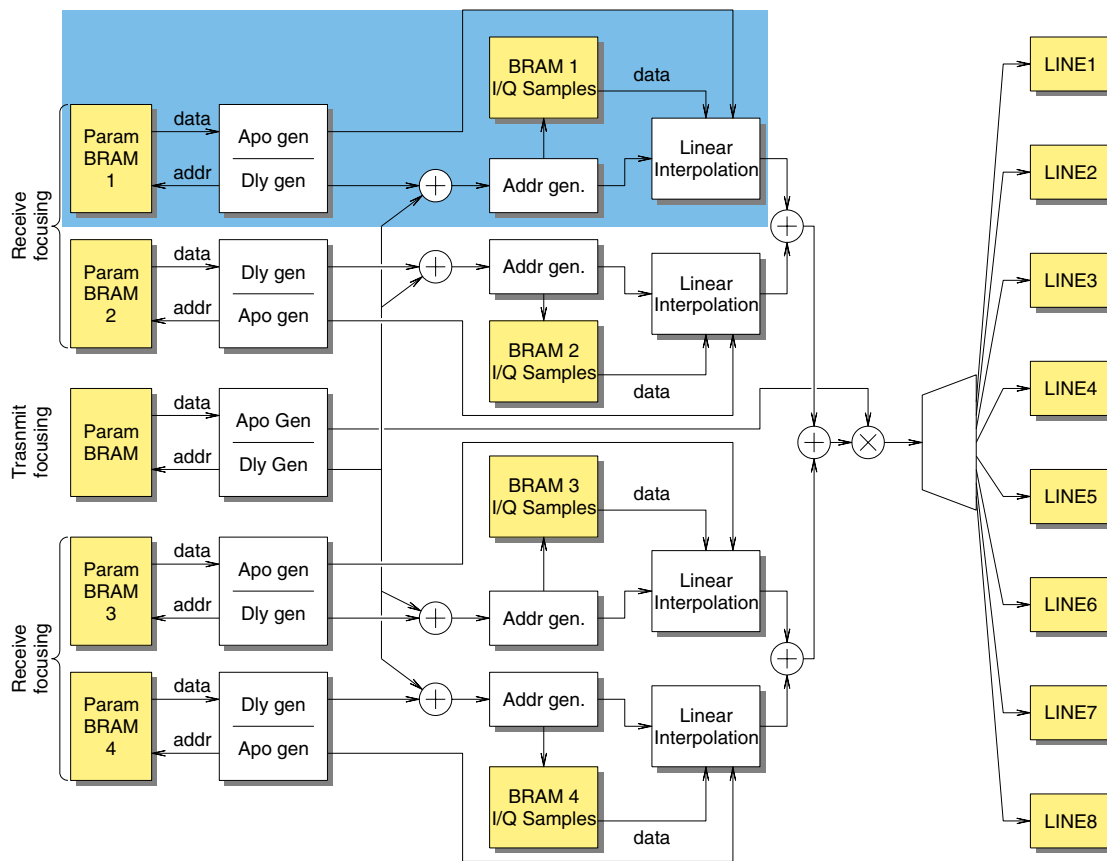


Fig. 4. Parametric beamformer block used in SARUS.

- [2] C. B. Burckhardt, P-A. Grandchamp, and H. Hoffmann. An experimental 2 MHz synthetic aperture sonar system intended for medical use. *IEEE Trans. Son. Ultrason.*, 21(1):1–6, January 1974.
- [3] J. T. Ylitalo and H. Ermert. Ultrasound synthetic aperture imaging: monostatic approach. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 41:333–339, 1994.
- [4] M. H. Bae, M. K. Jeong, T. K. Song, and Y. B. Ahn. Experimental study of transmit synthetic focusing combined with receive dynamic focusing in B-mode ultrasound imaging systems. In *Proc. IEEE Ultrason. Symp.*, pages 1261–1264, 1999.
- [5] M. H. Pedersen, K. L. Gammelmark, and J. A. Jensen. In-vivo evaluation of convex array synthetic aperture imaging. *Ultrasound Med. Biol.*, 33:37–47, 2007.
- [6] M. Karaman, P. C. Li, and M. O'Donnell. Synthetic aperture imaging for small scale systems. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 42:429–442, 1995.
- [7] M. Karaman and M. O'Donnell. Subaperture processing for ultrasonic imaging. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 45:126–135, 1998.
- [8] K. L. Gammelmark and J. A. Jensen. Multielement synthetic transmit aperture imaging using temporal encoding. *IEEE Trans. Med. Imag.*, 22(4):552–563, 2003.
- [9] G. E. Trahey and L. F. Nock. Synthetic receive aperture imaging with phase correction for motion and for tissue inhomogeneities - part II: effects of and correction for motion. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 39:496–501, 1992.
- [10] K. L. Gammelmark and J. A. Jensen. Duplex synthetic aperture imaging with tissue motion compensation. In *Proc. IEEE Ultrason. Symp.*, pages 1569–1573, 2003.
- [11] S. I. Nikolov. *Synthetic aperture tissue and flow ultrasound imaging*. PhD thesis, Ørsted•DTU, Technical University of Denmark, 2800, Lyngby, Denmark, 2001.
- [12] S. I. Nikolov and J. A. Jensen. In-vivo synthetic aperture flow imaging in medical ultrasound. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, pages 848–856, 2003.
- [13] J. A. Jensen and S. I. Nikolov. Directional synthetic aperture flow imaging. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, pages 1107–1118, 2004.
- [14] J. A. Jensen and N. Oddershede. Estimation of velocity vectors in synthetic aperture ultrasound imaging. *IEEE Trans. Med. Imag.*, pages 1637–1644, 2006.
- [15] J. A. Jensen, O. Holm, L. J. Jensen, H. Bendsen, S. I. Nikolov, B. G. Tomov, P. Munk, M. Hansen, K. Salomonsen, J. Hansen, K. Gormsen, H. M. Pedersen, and K. L. Gammelmark. Ultrasound research scanner for real-time synthetic aperture image acquisition. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 52 (5):881–891, May 2005.
- [16] J. Udesen, M. B. Nielsen, K. R. Nielsen, and J. A. Jensen. Examples of in-vivo blood vector velocity estimation. *Ultrasound Med. Biol.*, 33(4):541–548, 2007.
- [17] M. O'Donnell. Coded excitation system for improving the penetration of real-time phased-array imaging systems. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 39:341–351, 1992.
- [18] T. X. Misaridis, K. Gammelmark, C. H. Jørgensen, N. Lindberg, A. H. Thomsen, M. H. Pedersen, and J. A. Jensen. Potential of coded excitation in medical ultrasound imaging. *Ultrasonics*, 38:183–189, 2000.
- [19] J. A. Jensen. Field: A program for simulating ultrasound systems. *Med. Biol. Eng. Comp.*, 10th Nordic-Baltic Conference on Biomedical Imaging, Vol. 4, Supplement 1, Part 1:351–353, 1996b.
- [20] S. I. Nikolov, J. A. Jensen, and B. G. Tomov. Recursive delay calculation unit for parametric beamformer. In *Proc. SPIE - Progress in biomedical optics and imaging*, volume 6147-13, pages 1–12, 2006.
- [21] S. I. Nikolov, J. A. Jensen, and B. G. Tomov. Fast parametric beamformer for synthetic aperture imaging. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, page submitted, 2007.