Experiments in Real-Time MRI with RT-Hawk and Medusa

P. Stang[1], J. Santos[1], J. Pauly[1], and G. Scott[1]

[1]Electrical Engineering, Stanford University, Stanford, CA, United States

INTRODUCTION:

Real-time MRI is indispensable for visualizing fast-evolving processes such as cardiac function. Successful real-time operation also imposes stiff requirements on software and hardware, including the need for efficient pulse sequences, optimized reconstruction routines, tight synchronization between software and scanner, high-speed low-latency data transfer, and an interactive user interface. To explore these requirements, we assembled a hardware and software suite using RT-Hawk, a real-time MRI software engine [1], and Medusa, an open-architecture scalable MRI console [2]. Using this system, we successfully demonstrated high-performance real-time MRI at 1.5T. We are investigating the limitations of the Medusa system architecture and Universal Serial Bus (USB) data transports and their resulting impact on real-time performance in MRI systems.

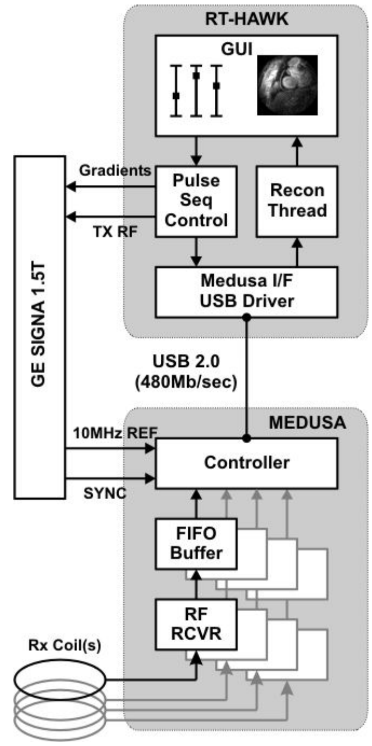


Figure 1: Medusa receives the MR signal, passing the data to RT-Hawk over USB 2.0 for real-time reconstruction. SIGNA handles gradient and Tx RF playback.

METHODS:

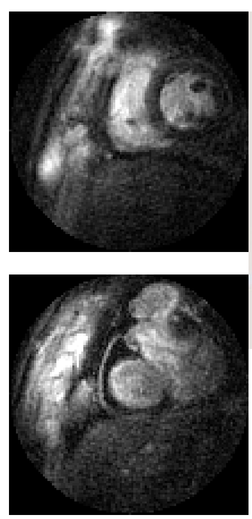
Our real-time MRI system is composed of three major components: the RT-Hawk software engine, a Medusa Console, and a GE SIGNA 1.5T scanner. RT-Hawk serves as the scan controller, providing the user interface, pulse sequence synthesis, and image reconstruction. Based on interactive user input, RT-Hawk generates gradient and RF pulse sequences on-the-fly and sends them to the SIGNA scanner for immediate playback at each TR. The Medusa RF Module acquires the MR RF signal from the receive coils, down-converts, and stores the complex baseband data in a local FIFO memory. Sampling and TR synchronization is implemented directly in an Altera MAX-II EPM1270 Complex Programmable Logic Device (CPLD). This allows each RF Module to transmit and receive RF with no CPU overhead, ensuring that acquisition always occurs on-time, regardless of CPU loading. Only a TR synchronization pulse and a 10MHz reference are required to maintain frequency and sampling alignment between SIGNA and the Medusa console. The Medusa Controller is responsible for processing and packetizing RF data streams for transfer via USB to the upstream host PC. To make efficient use of the USB bus, we developed a low-latency operating mode for the Medusa Controller. In this mode, received RF data is automatically aggregated from the FIFO memories and forwarded over the USB bus immediately following the active readout interval of the pulse sequence. This asynchronous approach minimizes latency from the time data is acquired to the moment it reaches RT-Hawk. Upon receipt of each RF data set RT-Hawk performs reconstruction and updates the real-time display.

RESULTS:

We have demonstrated our real-time acquisition system using both spiral and 2DFT pulse sequences. Figure 2 shows frame-captures from in-vivo imaging of the heart at 50 frames per second using an interleaved 3072-pt spiral sequence with TR=20ms and an RF bandwidth of 250kHz. Incoming spiral data is reconstructed with a 4-TR sliding-window to maintain high image quality while contributing minimal motion blur. In a more strenuous test of communication latency, RT-Hawk and Medusa successfully kept pace with a 128 x 128 2DFT sequence at TR=4.5ms. In bench tests, Medusa delivers continuous real-time operation on 4 receiver channels sampled at 250kHz for a total data rate of 4.2 MBytes/sec. In comparison, low-latency is not required, each Medusa Controller can support up to 9.8 MBytes/sec usable transfer rate.

DISCUSSION:

The distributed parallel architecture of the Medusa Console is especially well-suited to real-time applications. The use of local buffering and dedicated acquisition control logic for every RF channel permits received data to be read out even while the next TR is already in progress. However, real-time performance of the USB 2.0 connection is disappointing, especially when transferring small packets. The 1ms frame interval of USB 2.0 imposes a latency often 1ms or longer for any data transfer. Our experiments confirm that the majority of this delay occurs in the host PC USB driver, and we expect that improvements could be made by pre-allocating bus bandwidth using USB Isochronous Mode, or by writing a real-time driver. Alternately, USB 3.0 (4.8Gb/sec) has already been demonstrated in industry and promises universally better performance, but the standard is not expected to be ratified until late 2008.



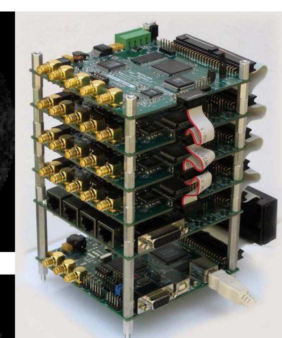


Figure 2: A 4-ch Medusa Console with USB (above). Still-frames from real-time cardiac imaging at 50fps acquired using Medusa and RT-Hawk. (left)

CONCLUSIONS:

We have demonstrated real-time operation of a Medusa Console in single-channel receive mode with RT-Hawk, and 4-channel tests are being prepared. Although our USB data link currently limits real-time performance to 4 receiver channels per Medusa Controller, this is already sufficient for a wide variety of applications. Furthermore, Medusa's scalable architecture allows for multiple controllers to be used in parallel to overcome data bottlenecks. Medusa is also offers native control of transmit RF and gradients, and we are developing experiments to test these features in a real-time regime.

REFERENCES:

[1] P. Stang et al., Proc 15th ISMRM, p925, 2007.

[2] Santos JM, et al. 26th IEEE EMBS, 1048, 2004.