

بناام خدا

گزارش تمرين ۳

آرشام لؤلؤلهری

۹۹۱۰۲۱۵۶

بخش تئوری:

Aliasing:

این پدیده که بطور عمده در تصویربرداری های Pulsed wave Doppler و Color flow Doppler دیده میشود، مربوط به شرایطی است که میخواهیم سرعت یک flow مثل جریان خون در رگ را (با استفاده از شیفت فرکانسی داپلر) اندازه بگیریم، و این سرعت از حد ماکزیمم توان سیستم اندازه گیری (که بر حسب نرخ نمونه برداری تعیین میشود) فراتر است و در نتیجه ی آن، سرعت جریان، به درستی اندازه گیری نمیشود و حتی میتواند این سرعت در جهت معکوس اندازه گیری شود.

مشکل اینجاست که هنگامی که پالس ها به صورت متوالی به سمت بافت ارسال میشوند، برای تداخل نکردن این پالس ها و زمان دریافتشان با یکدیگر، باید پالس دوم پس از دریافت echo ی پالس اول ارسال شود. این کار *sampling rate* ما را محدود میکند. واضحا هرچه سرعت انتشار موج بیشتر بوده و یا عمق تصویربرداری کمتر باشد، این محدودیت کمتر است چرا که پالس ها سریعتر دریافت میشوند. در اصل کمیتی بنام *pulse repetition frequency* به صورت زیر تعریف میشود:

$$PRF = \frac{\text{Propagation Speed}}{\text{Pulse Spacing}}$$

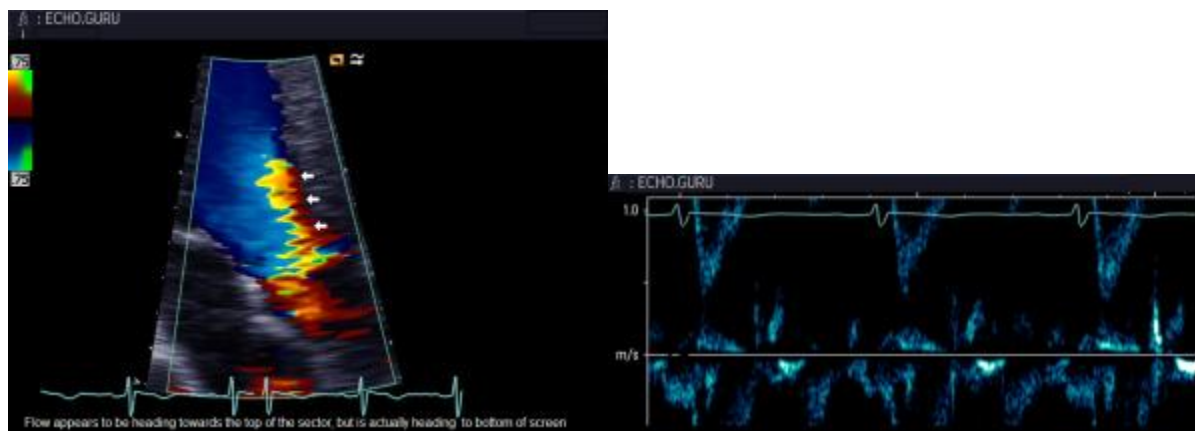
که *Pulse Spacing* مسافت طی شده توسط موج است. حال حد ماکزیمم سرعت قابل اندازه گیری در این روش برابر است با:

$$\text{Nyquist Limit} = \frac{PRF}{2}$$

اگر سرعت جریان از این حد نایکویست بیشتر باشد، با نرخ نمونه برداری محدودی که داریم، نمیتوان سرعت آن را به درستی اندازه گیری کرد. در این شرایط، مشابه چرخ های یک خودرو که

در سرعت های زیاد، جهت چرخششان برعکس بنظر میرسد، سرعت ثبت شده توسط دستگاه، سرعتی است در خلاف جهت اصلی، و عموماً با اندازه ای کمتر از اندازه ی درست.

در تصویر زیر میبینیم که در اثر این پدیده، سرعت خون بجای اینکه به سمت پایین بدست بیاید، به اشتباه به سمت بالا بدست آمده.



Field of View (FOV):

به نواحی ای از بافت هدف (مورد تصویر برداری) در همه ابعاد گفته میشود که در اسکن و تصویربرداری ما، پوشش داده شده است. بسته به دستگاه تصویربرداری و ناحیه ی هدف، FOV میتواند مقادیر مختلفی داشته باشد. مثلاً در تصویربرداری از داخل رگ توسط IVUS (Intravascular Ultrasound)، مقدار FOV تنها چند میلیمتر است. در MR این مقدار در اردر میلیمتر یا سانتی متر است و همچنین FOV میتواند فرم مربعی داشته باشد یا اصلاً متقارن نباشد.

Resolution:

رزولوشن، در واقع میزان واضح بودن (clarity) تصویر ثبت شده میباشد. رزولوشن معمولاً در دو راستای عمقی (lateral) و عرضی (axial) سنجیده میشود که هر دو رزولوشن به فرکانس موج

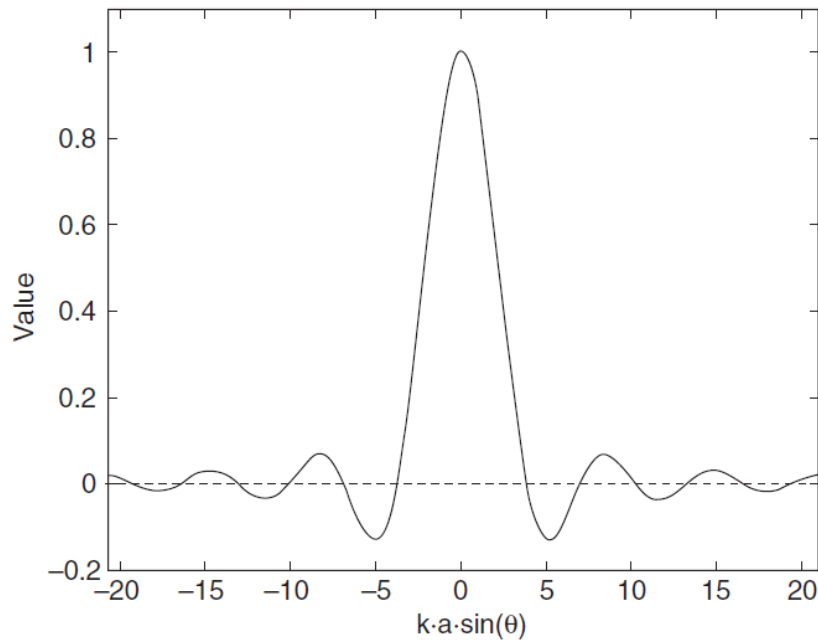
ارسالی وابسته اند. برای داشتن رزولوشن عمقی بهتر، باید پالس ارسالی در زمان بسیار نازک باشد تا هنگام دریافت اکو، زمان دریافت اکو و در نتیجه مکان دقیق مرز مشخص شود. از آنجا که باریک بودن پالس در زمان موجب بیشتر شدن پهنای باند فرکانسی میشود، پهنای باند بزرگتر موجب بهتر شدن رزولوشن عمقی میشود.

رزولوشن عرضی به معنی قابلیت تفکیک دو structure ای است که عمود بر راستای ارسال موج، کنار یکدیگر قرار دارند. برای داشتن رزولوشن بهتر عرضی، باید عرض beam تولیدی بسیار نازک باشد تا به هنگام تصویربرداری از یک ذره‌ی مشخص، اثر structure های کناری در آن ظاهر نشود. از طرفی فرکانس موج با عرض beam نسبت عکس دارد. پس فرکانس های بیشتر موجب افزایش رزولوشن عرضی میشود.

Directivity:

این ویژگی که تحت تاثیر آرایه ی ترنسدیوسرهای ارسالی تعیین میشود، بیانی است از میزان توزیع فشار در اطراف نقطه ی هدف، که تعیین میکند تا چه زوایایی در اطراف نقطه ی هدف، همچنان دامنه ی فشار زیادی داریم. بر این اساس، برحسب اینکه نوع آرایه ترنسدیوسر چه باشد، میتوان تابعی بر حسب زاویه ی انحراف تعریف کرد بنام directivity function. این تابع، بخشی از دامنه ی فشار تولیدی توسط سورس است که تابع زاویه بوده و نشان میدهد beam تولیدی چقدر نازک است. طبیعتاً هرچه این beam نازکتر باشد، فشار روی نقطه هدف (نسبت به نقاط اطراف) متمرکزتر است و directivity بهتری داریم.

برای مثال، برای یک سورس به شکل دیسک که فشار در نقاط far field اندازه گیری شده، تابع directivity function (برحسب زاویه نسبت به مرکز دیسک) به صورت زیر است:



8.10. The profile of a directivity function for a disc-shaped transducer.

هرچه main lobe نازکتری داشته باشیم، directivity بهتر است. با افزایش فرکانس، این تابع فشرده تر شده و directivity بهتر میشود. همچنین با افزایش قطر دیسک (a در مثال بالا) نیز میتوان directivity را بهتر کرد.

DAS:

در این روش بازسازی تصویر، همانطور که از نامش پیداست، برای ساختن مقدار متناظر با هر پیکسل، سیگنال های دریافتی سنسورها به ترتیب خاصی شیفت خورده و سپس با یکدیگر جمع میشوند.

در واقع این تکنیک بر این اساس است که برای محاسبه ی مقدار هر پیکسل در تصویر بازسازی شده، فاصله این پیکسل از تمام سنسورهای گیرنده محاسبه میشود. این فاصله با داشتن مختصات پیکسل و نیز هر سنسور، بدست می آید. سپس با دانستن سرعت انتظار موج در محیط، میتوان تخمین زد که برای هر سنسور، چه مدت زمانی طول کشیده تا اکوی ارسال شده از آن پیکسل به سنسور برسد ($t = \frac{\text{distance}}{\text{sound velocity}}$). اگر این پیکسل مرز بافت موردنظر باشد، اکوی قوی ای

ارسال میکند و در نتیجه در دیتای سنسور، انتظار می‌رود در چنین زمانی، پیک مشخصی دیده شود. حال ایده در DAS این است که تمام سیگنال‌های سنسورهای مختلف را، به اندازه‌ای شیفت دهیم که پیک متناظر با این پیکسل در تمام سیگنال‌ها، همزمان شود و یک جا بیفتد. میتوان برای اینکار، هر سیگنال را به اندازه t محاسبه شده در بالا، شیفت داد. حال اگر تمام این سیگنال‌های شیفت یافته را با هم جمع کنیم، در ناحیه‌ی متناظر با پیکسل مورد نظر، تداخل سازنده (constructive) رخ میدهد و مقدار سیگنال به حد خوبی زیاد میشود در حالی که سایر جاها، تداخل‌ها رندوم هستند و سازنده‌ی مشخصی وجود ندارد. پس از محاسبه‌ی سیگنال مجموع، مقدار متناظر با این زمان در این سیگنال (که ماکزیمم مقدار سیگنال است)، بعنوان ارزش آن پیکسل تعیین میشود. این مقدار بدلیل شیفت‌هایی که انجام میشود، به زمان صفر منتقل میشود و در نتیجه در کد بخش عملی، خواهیم دید که اولین سمپل را برای ارزش پیکسل تعیین میکنیم. این کار باید برای تمام پیکسل‌ها تکرار شود تا تصویر نهایی ساخته شود. در زیر نمونه‌ای از این شیفت‌های سیگنال در ۳ سنسور مختلف را میبینیم، و ضمناً رابطه کلی مورد استفاده در DAS نیز آورده شده است:

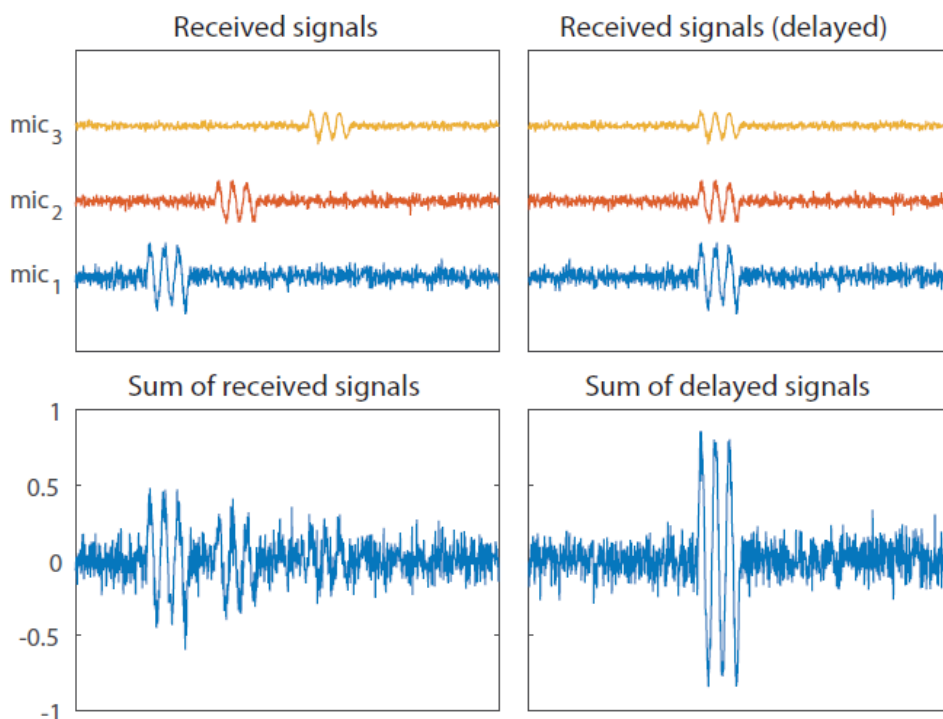


Fig. 1. The wavefield is arriving at individual microphones located at different positions at different times. Summing the signals directly distorts the signal, whereas by delaying the signals the waveforms captured by the individual sensors add constructively.

$$z(t) = \sum_{m=0}^{M-1} w_m \cdot y_m(t - \Delta_m) \quad (1)$$

رابطه بالا برای یک تک پیکسل نوشته شده است. تعداد سنسور ها M ، سیگنال هر سنسور y_m ، تاخیر (شیفت) هر سیگنال Δ_m است و ضمناً فرض شده که میتوان برای هر سنسور، ضریب w_m را در نظر گرفت که روش های متعددی برای محاسبه آن وجود دارد.

(منبع تصاویر و توضیحات فوق در زیر آمده است و در فایل ارسالی نیز قرار دارد)

Beamforming algorithms – beamformers – Jørgen Grythe, Norsonic AS,
Oslo, Norway

بخش عملی:

(توجه: به هنگام ران کردن کد، لطفا تصویر نمونه ای که در کنار کد آپلود شده است را در فولد ری یکسان با کد ارسالی قرار دهید. این تصویر در کد *load* شده است.)

۱:

ابتدا محیط شبیه سازی را تعریف میکنیم. طبق خواسته تمرین، گام های 0.1mm خواهیم داشت ($dx=dy=1e-4$) و کل محیط دو بعدی نیز 2.5cm است ($N_x=N_y=250$). با استفاده از `kwavegrid`، محیط شبیه سازی را تشکیل میدهیم.

سپس محیط شبیه سازی (آب) را با استفاده از سرعت موج در محیط، و نیز چگالی محیط، تعریف میکنیم و آن را در متغیر `medium` نگه میداریم.

سپس باید سورس تولید فشار را تعریف کنیم. طبق گفته سوال، فرض میشود که امواج قبلا به بافت برخورد کرده اند و اکنون صرفا اکوهای آن باید شبیه سازی شود. یعنی عملا سورس ما، همان تصویر اصلی بافت است که در کد به صورت ماتریسی با مقادیر نرمالیزه `load` میشود، و سپس کل این پیکسل ها در یک دامنه فشار که توسط ما تعیین میشود، ضرب میشود (برای شبیه بودن به مثال مذکور در سوال، دامنه فشار را همان ۳ گرفته ایم). سورس تولید فشار را در `p0` طراحی میکنیم و سپس برای هم سایز کردن این سورس با محیط شبیه سازی `2.5*2.5cm` خود، آن را `resize` میکنیم.

حال باید سنسورها را تعریف کنیم. با توجه به دایروی بودن آرایه، `sensor_radius` و `sensor_angle` را تعریف میکنیم تا شعاع دایره و مقدار زاویه چرخش را تعیین کنیم. سپس با

تقسیم شعاع بر dx محیط، میتوان مقدار شعاع را بر حسب تعداد `grid point` ها بدست آورد.

حال با دو روش میتوان آرایه سنسورها را تعریف کرد: باینری یا کارتزین

در روش باینری، با استفاده از `makeCircle` یک ماتریس به ابعاد `grid` ساخته میشود و هر جا که سنسوری داشته باشیم، عدد ۱ و سایر جاها صفر داریم. ورودی های این تابع نیز `Nx, Ny`، مرکز دایره (که برای ما مرکز تصویر است) و شعاع و زاویه دایره هستند.

در روش کارتزین، با استفاده از `makeCartCircle`، ماتریسی با تنها دو سطر ساخته میشود که سطر اول مکان `x` سنسورها و سطر دوم مکان `y` آنهاست. ورودی این تابع شعاع دایره و تعداد سنسورهاست که برای بخش اول، ۱۰۰ تاست.

در اینجا ما از کارتزین استفاده کرده و سنسورها را تعریف میکنیم. سپس با `kspacFirstOrder2D` شبیه سازی را انجام میدهیم.

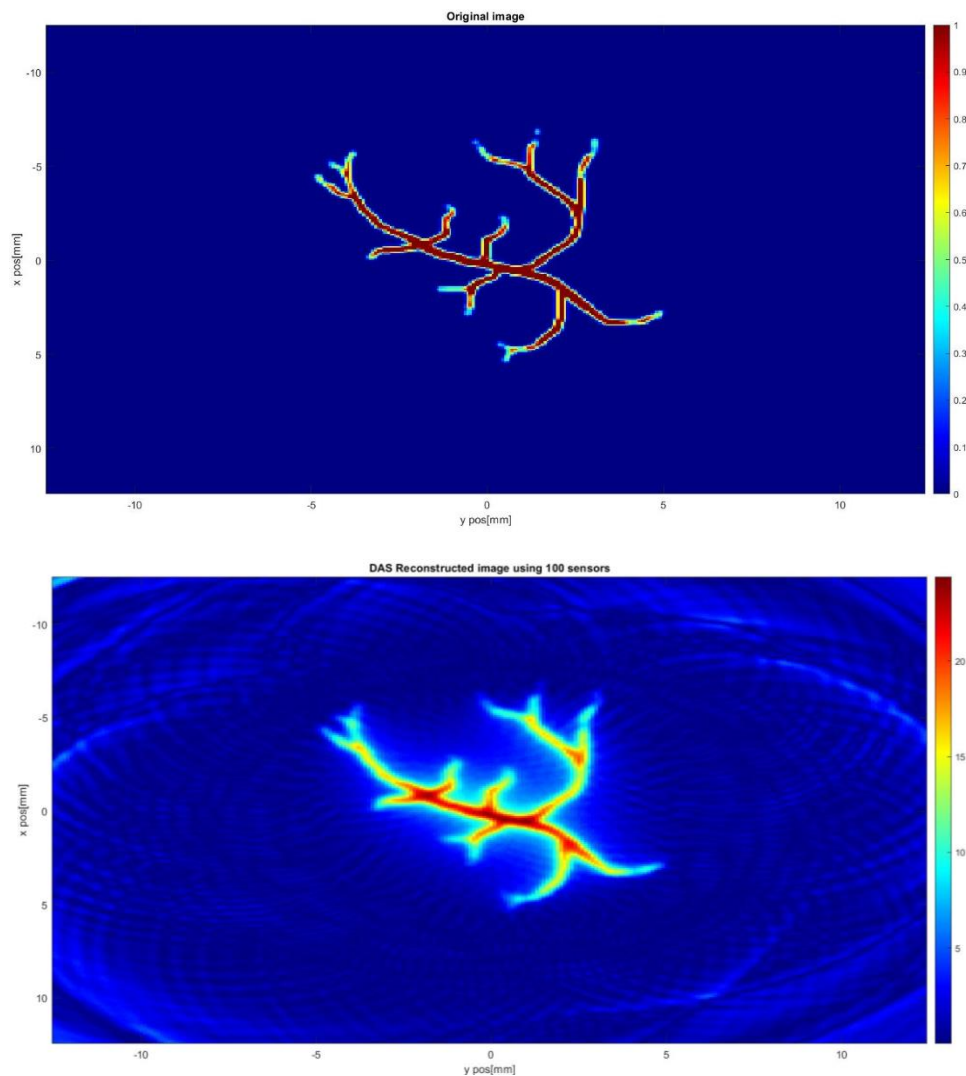
گام های زمانی (`dt`) و تعداد کل سمپل های زمانی (`Nt`) در `kgrid` ذخیره میشوند. (چون مقادیر اتوماتیک تعیین شده، شرایط نایکوئیست را رعایت میکنند و مدت شبیه سازی نیز نسبتاً معقول است، آنها را تغییر نداده ایم).

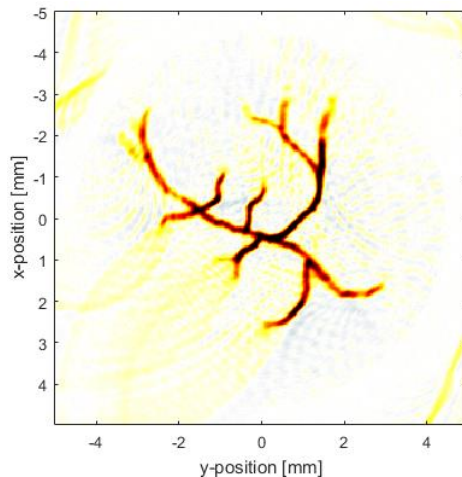
تصویر نهایی بازسازی شده، ماتریس `grid_vals` میباشد.

حال باید توضیحات ذکر شده در بخش تئوری در مورد روش `DAS` را انجام دهیم. مکان `x` و `y` سنسورها را با استفاده از آرایه ی کارتزین، استخراج میکنیم. سپس از دو حلقه ی تودرتو برای تعیین ارزش هر پیکسل که در یک `x, y` مشخص قرار دارد، استفاده میکنیم. مکان این پیکسل را تعیین میکنیم (`point_x, point_y`) و فاصله اقلیدسی آن از هریک از سنسورها را محاسبه میکنیم و در `sensors_dist` میریزیم. همانطور که در توضیحات `DAS` دیدیم، میزان شیفت هر سیگنال برابر است با حاصل تقسیم این فاصله بر سرعت موج (در آب). سپس این مدت زمان شیفت را به تعداد سمپل تبدیل میکنیم و در `sensors_sampleShift` میریزیم. حال با استفاده از حلقه ی `for` سوم، برای هر سنسور، دیتای آن را با استفاده از `circshift` به اندازه ی تعداد سمپل تعیین شده، شیفت میدهیم (به تعداد سمپل های شیفت، انتهای سیگنال صفر قرار

میدهیم چون circshift شیفت دایره ای میدهد اما انتهای سیگنال برای ما مهم نیست). در نهایت همه سیگنال های شیفت خورده را با یکدیگر جمع میکنیم و اولین سمپل را (که مطابق توضیحات DAS، متناظر با اکوی پیکسل مدنظر است)، بعنوان ارزش آن پیکسل قرار میدهیم. بدین ترتیب تمام ماتریس grid_vals ساخته میشود.

در زیر تصویر اصلی ای که برای شبیه سازی انتخاب شده (تصویر اول)، تصویر بازسازی شده با Time Reversal (تصویر دوم) و تصویر بازسازی شده با DAS (تصویر آخر) مشاهده میشود:



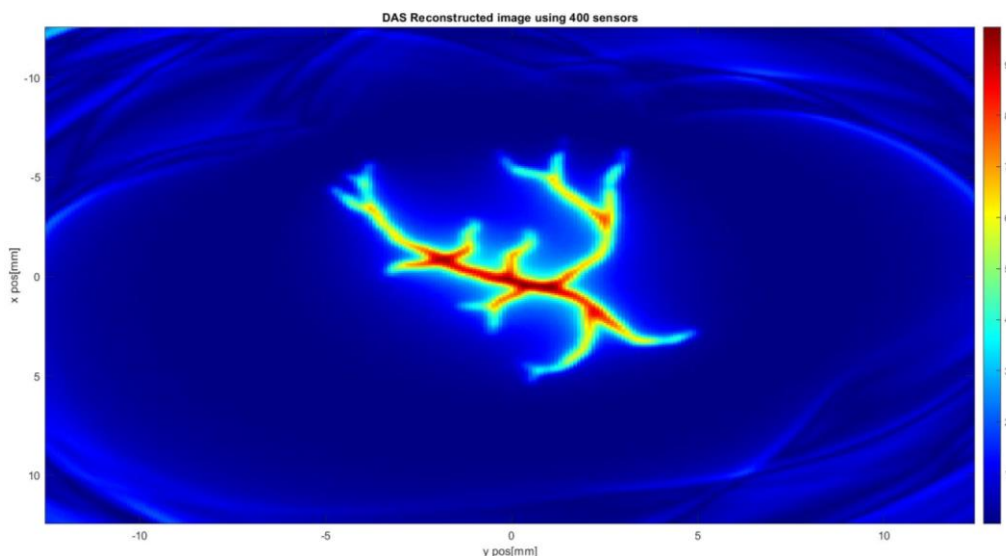


در هر دو تصویر نویز هایی در اطراف بافت اصلی دیده میشود که بخشی از آن به دلیل محدودیت تعداد ترنسدیوسرهاست.

در DAS، علاوه بر کمتر بودن زمان شبیه سازی نسبت به time reversal، مشاهده میشود که لبه های بافت بهتر و واضح تر نمایش داده شده است. در تصویر Time Reversal، حتی با وجود انجام interpolation، لبه ها اندکی حالت blur دارند.

۲:

حال تعداد سنسورها (num_sensor_points) را به ۴۰۰ افزایش میدهیم و مجدداً تصویر را رسم میکنیم:



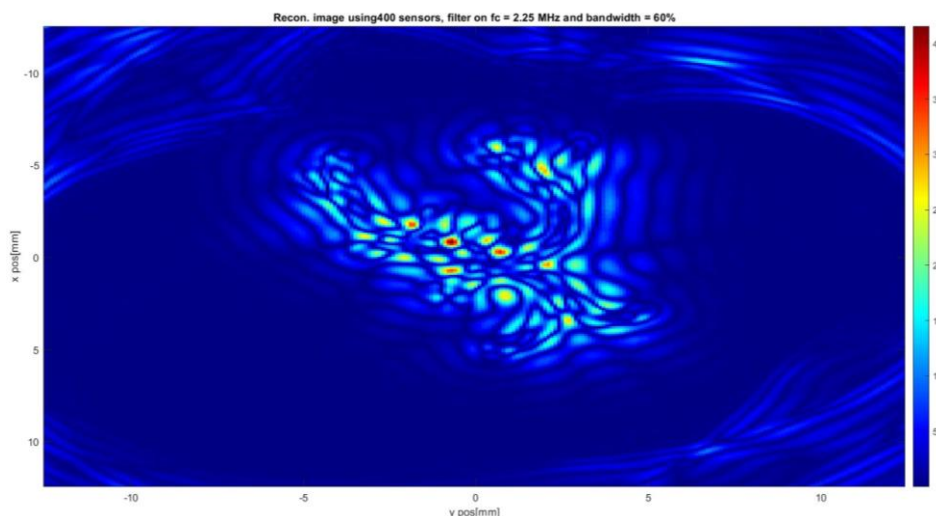
مشاهده میشود با افزایش تعداد سنسورها، تا حد خوبی نویزهای اطراف بافت حذف شده اند. در واقع افزایش تعداد سنسورها باعث میشود هنگام شیفٹ دادن سیگنال سنسورها، بخش هایی که مربوط به پیکل مدنظر نیستند و سیگنالشان فرم تصادفی دارد، بهتر یکدیگر را **cancel out** کنند و اثر **destructive** بهتری روی هم داشته باشند و بدین ترتیب نویز تا حد خوبی رفع میشود.

۳:

کد مربوط به این بخش در **section** بعدی نوشته شده است. همان کار های قبلی انجام شده، با این تفاوت که قبل از دو حلقه ی **for**، دیتای بدست آمده از **sensor_data** را فیلتر کرده و در **filtered_sensor_data** میریزیم. فیلترینگ با تابعی به نام **bandFilt** انجام میشود. داخل این تابع از **designFilt** برای طراحی کردن فیلتر، و سپس از **filter** برای اعمال فیلتر طراحی

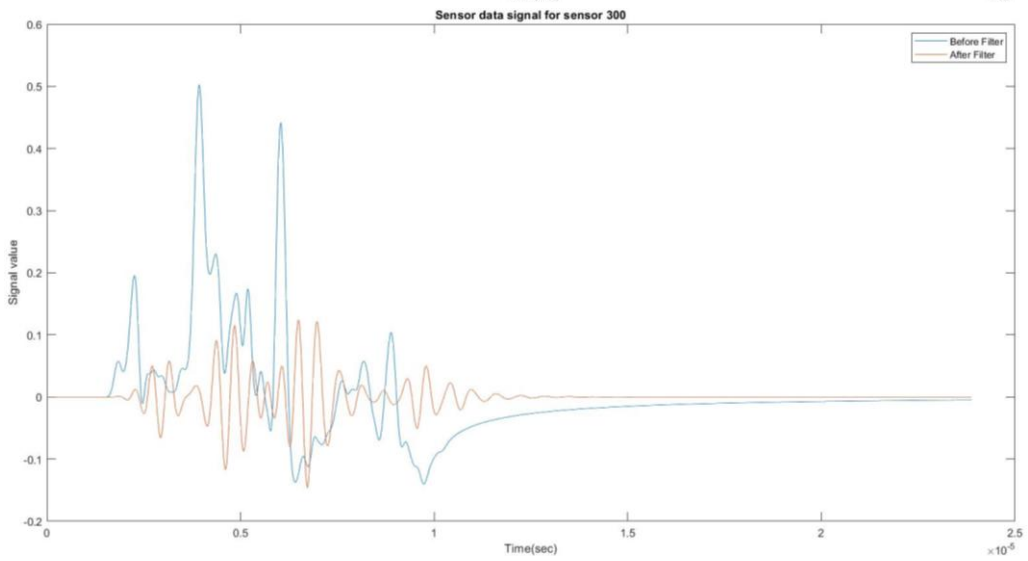
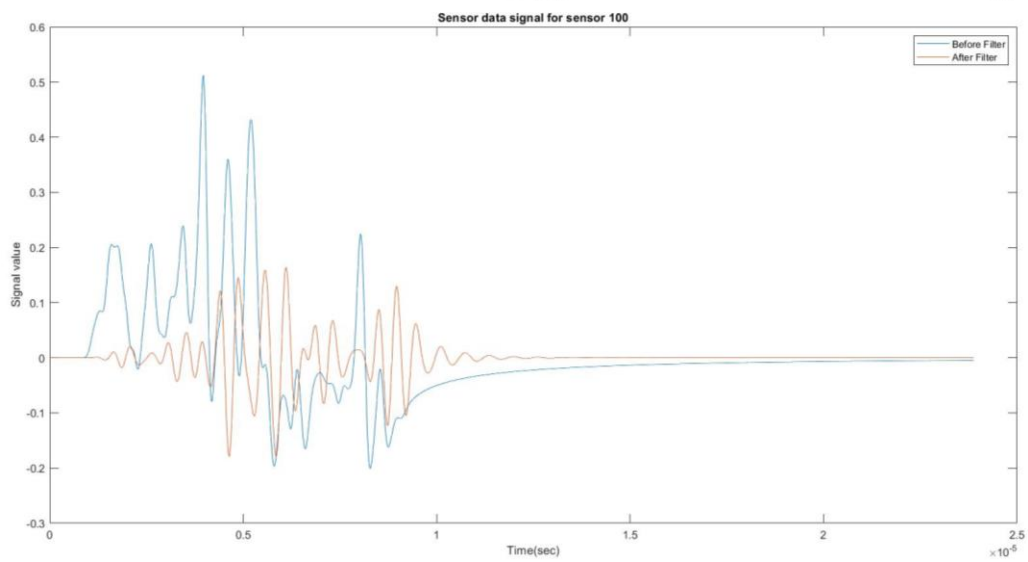
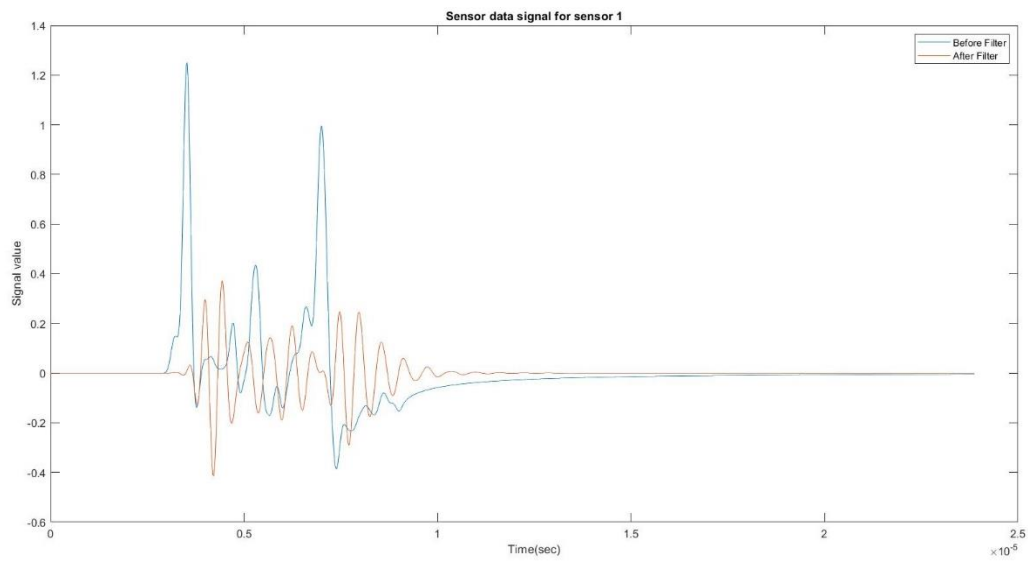
شده بر روی دیتا استفاده میشود. ورودی های این تابع، خود دیتا (که فرض شده که بعد اولش زمان و بعد دومش سنسور ها هستند)، فرکانس مرکزی، پهنای باند و نرخ نمونه برداری هستند.

سپس در حلقه های تودرتو، بجای `sensor_data` از فیلتر شده ی آن، یعنی `filtered_sensor_data` استفاده میکنیم و بقیه ی کد مشابه قبل است. در نهایت تصویر بازسازی شده با ۴۰۰ سنسور و به ازای فرکانس مرکزی 2.25 MHz، به صورت زیر است:



فیلتر سنسورها، محدوده فرکانسی ۶۰ درصد قبل و بعد از فرکانس مرکزی را عبور میدهد که در این حالت، عملاً یک فیلتر میانگذر است که فرکانس های بسیار پایین و بسیار بالا را عبور نمیدهد. سیگنال دریافتی توسط سنسورها، زمانی تغییرات شدید دارد که به محل مرز های بافت میرسیم. در این مکان ها، سیگنال از مقادیر نسبتاً ثابت و کوچک (فرکانس های خیلی پایین)، به یکباره شروع به افزایش میکند (نوسانات سریع و فرکانس های نسبتاً بالاتر که توسط فیلتر عبور داده میشود) و وقتی در میانه ی بافت هستیم نیز، معمولاً مقادیر زیاد با نوسانات اندک و کند داریم (فرکانس های پایین). با این توصیف میتوان گفت فیلتر تا حدی، مرز های سیگنال را که فرکانس های بالایی در محدوده 2.25MHz دارند، حفظ میکند اما خارج بافت و نیز در میانه های بافت، فیلتر دیتا را حذف کرده و تصویر خاصی دیده نمیشود. به همین دلیل است که در تصویر بالا، صرفاً هاله ای از مرز های بافت رگ نمایان است اما داخل رگ نیز مقادیر کوچکی داریم. (در پاسخ سوال ۵ خواهیم دید که اگر فرکانس های بزرگتر را نیز نگه داریم، بهتر میتوان مرز ها را نگه

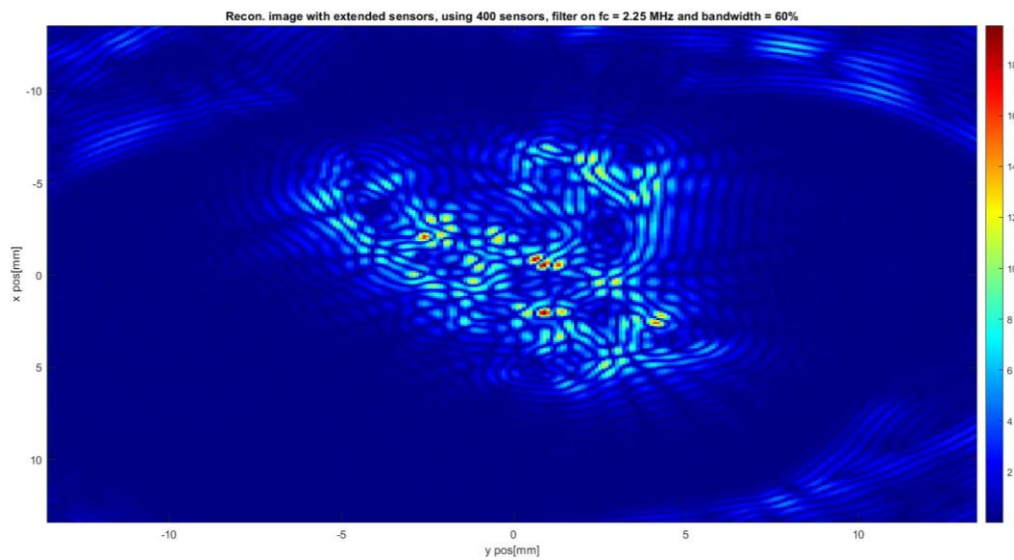
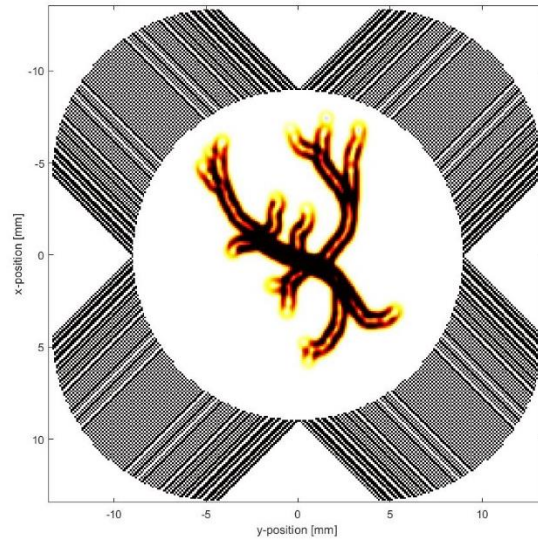
داشت و در نتیجه مرز ها نمایان تر هستند). در زیر سیگنال دریافت شده توسط سنسورهای ۱، ۱۰۰ و ۳۰۰ را، قبل و بعد از فیلتر میبینیم. در این سیگنال ها مشخص است که مقادیر بسیار بزرگ حذف میشوند اما هنگام افزایش (و یا کاهش) ناگهانی سیگنال، به پیک هایی در سیگنال فیلتر شده میرسیم:



برای این بخش باید هریک از سنسورها را به طول ۶ میلیمتر گسترش دهیم. (برای اینکه سنسورها همگی خارج از بافت باشند و هریک طول ۶ میلیمتر داشته باشد، سایز grid را به 2.7cm افزایش میدهیم). مشابه قبل سورس و محیط را تعریف میکنیم و دایره ی اصلی سنسورها را نیز مشابه قبل تعریف میکنیم. حال پارامتر هایی به نام `sensors_dx` و `sensors_dy` تعریف میکنیم که فاصله ی مکان سنسور از مرکز تصویر را نشان میدهند. ایده این است که اگر این فاصله مکانی، مثلاً در راستای X منفی بود، یعنی سنسور روی محور X، عقب تر از مرکز است. در این حالت تمام سنسور های خط ۶ میلیمتری، روی محور X باید پشت این سنسور اولیه قرار بگیرند. به همین دلیل برای این عناصر، `sensors_dx` را به `-dx` تغییر میدهیم. به همین ترتیب اگر این فاصله مثبت باشد، این پارامتر را به `+dx` تغییر میدهیم. برای محور Y نیز این کار را تکرار میکنیم.

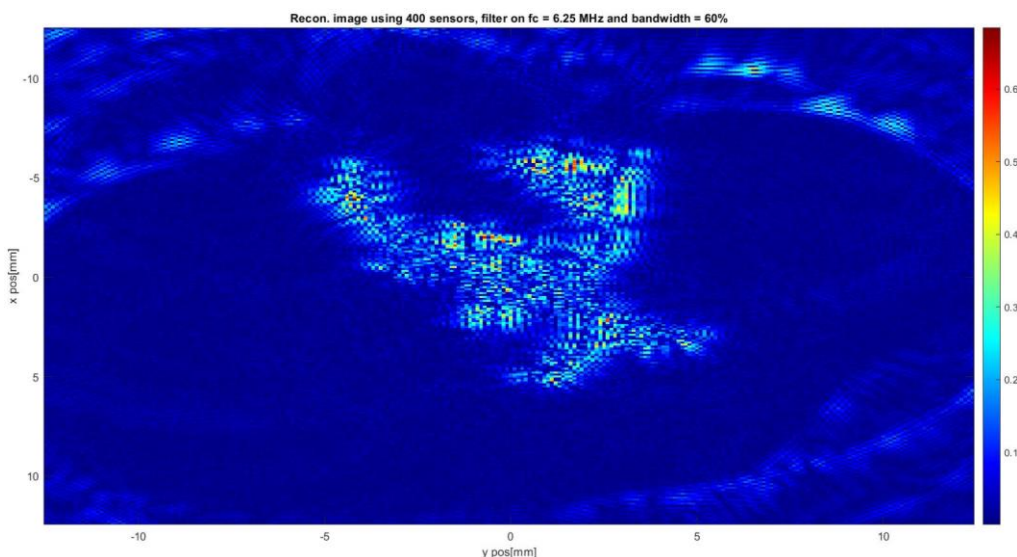
حال میخواهیم `ext_cart_sensor_mask` را بعنوان `mask` جدیدمان تشکیل دهیم (متغیر `cart_sensor_mask`، تنها سنسورهای دایره ی اصلی را نشان میداد). میخواهیم تا زمانی که طول هر خط به ۶ میلیمتر میرسد، سنسور های جدید اضافه کنیم پس از `while` استفاده میکنیم. در هر حلقه، مکان X سنسور جدید (`newSensors_x`)، مکان X سنسور قبلی، بعلاوه ی `sensors_dx` متناظر با خودش است. همینکار را برای Y نیز انجام میدهیم. این مختصات را به ماتریس سنسورهایمان اضافه میکنیم و طول خط جدیدمان را حساب میکنیم تا چک شود که از ۶ میلیمتر بزرگتر نشود. بدین ترتیب ماتریس سنسورهای جدید ساخته میشود و آن را بعنوان `mask` جدیدمان در نظر میگیریم. مابقی کد مشابه بخش ۳ است.

نتیجه چیدمان سنسورها، و تصویر بازسازی شده نهایی پس از فیلتر به صورت زیر است که مشاهده میشود مشابه بخش قبل، صرفاً مرزهای بافت تا حدی قابل تشخیص است و میانه های بافت بدلیل فیلتر شدن حذف شده است:



۵:

بخش ۳ را به ازای فرکانس مرکزی $f_c = 6.25$ MHz تکرار میکنیم. تصویر بازسازی شده به صورت زیر است:



مشاهده میشود که نسبت به فرکانس مرکزی 2.25 MHz، جزئیات مرزها کمی نمایان تر شده اند و شکل کلی مرزهای بافت را بهتر میتوان تشخیص داد. همانطور که در سوال ۳ نیز گفته شد، وقتی فرکانس مرکزی را افزایش میدهیم، تغییرات شدیدتر سیگنال را بهتر میتوانیم حفظ کنیم و در نتیجه مرزها نمایان تر میشوند. البته بدلیل حذف بخش بزرگی از فرکانس های پایین، نویزهایی نیز بطور محسوس در قسمت بافت وجود دارند.

بخش ۴ را نیز با فرکانس مرکزی 6.25 MHz تکرار میکنیم و به نمودار زیر میرسیم که همان تغییرات مذکور را (حفظ بیشتر و بهتر مرزها) نسبت به سوال ۴ دارد:

