

2018 IEEE IUS SA-VFI Challenge

Authors: Jørgen Arendt Jensen and Carlos Armando Villagómez-Hoyos
 Reviewers: Hervé Liebgott, Olivier Bernard, and Alfonso Rodríguez-Molares
 Version 1.0, November 14, 2017

CONTENTS

I	Introduction	2
II	Imaging sequence	2
II-A	Probe	2
II-B	Excitation	3
II-C	Duplex SA sequence	3
III	Supplied Data	5
III-A	Data format	6
III-B	Data sets	8
IV	Evaluation in the competition	8
IV-A	Global measurements	9
IV-B	Local measurements	9
IV-C	Validation code	9
V	Processing examples	10
V-A	Beamforming example	10
V-B	Velocity estimation example	11
VI	Evaluation examples	12
VI-A	Straight vessel	12
VI-B	Spinning disk	13
VI-C	CFD	14
	References	14
	Appendix	15
A	Supported MATLAB and Toolboxes	15

I. INTRODUCTION

Given the positive response to the IUS 2017 challenge on plane wave imaging (PICMUS)[1], we propose to organize a challenge on synthetic aperture (SA) 2-D vector flow imaging (VFI). The challenge consist on estimating blood flow velocities from both simulated and measured ultrasound RF element data sets. The challenge will be deployed using the same web platform used for the PICMUS challenge. The results will be presented during the next IEEE International Ultrasonics Symposium 2018 in Kobe (Japan).

During the challenge, the participants will estimate 2-D blood velocities at a pre-defined spatio-temporal grid imposed by the organizers, and upload their final estimates together with a validation code to the platform. The data sets are designed to enable the evaluation of the following two global metrics: bias and standard deviation; as well as the peak velocity in selected data sets. The data sets consist on pre-beamformed RF element signals using a pre-selected SA sequence on flow phantoms, both simulations and experiments, which are provided to the challengers through the Zenodo platform ([10.5281/zenodo.1045795](https://doi.org/10.5281/zenodo.1045795)). A reference wire phantom data set is also be provided to give the participant the possibility to ensure that the correct beamforming process is in place.

In November 2017, the participants are granted access to the 1st stage data sets and the example code material for the challenge. This will allow them to start working with the data before the platform opens. A forum for the participants, where they can receive additional input from organizers and other users is available at the [SA-VFI challenge website](#). As soon as the platform opens in March 2018, participant must create an account. The participant must then upload their results from the first stage data set in the exact same format as specified in Section IV, together with their validation MATLAB code. After uploading their result, the participant will receive a second stage double blinded data set from which they will estimate the velocities. For each participant, the results from the metrics described in Section IV are used to allocate points. These points serve to classify and rank the challengers and designate a winner.

The challenge is organized in two stages. In a first stage, global metrics from the first stage data sets are used for attributing points and rank the participants. In this stage, a correction factor attributed to the number of emissions is used for the calculation of the number of points. In the second stage, both global metrics and local metrics from the second stage data sets are used and participants is free to use any number of emissions.

This document describes the data provided for the IEEE IUS 2018 SA VFI challenge. The data is acquired using a synthetic aperture sequence on a linear array as described in Section II. The data sets provided to the contestants in the different stages are described in Section III-B. The evaluation criteria and data format for the results are detailed in Section IV. Reference code examples from beamformed and fixed-angle velocity estimation are detailed in Section V.

II. IMAGING SEQUENCE

A. Probe

A 128-element linear array probe is selected for the challenge. The transducer coordinate system is defined following the convention defined in Field II [2], [3], where the z coordinate increases with depth, x is along the elements, and a right-handed coordinate system is used. The array has a measured center frequency of 8 MHz and 60% bandwidth. The transmit impulse response is shown in Fig. 1, as measured with an Onda HGL-0400 hydrophone [4]. The transducer parameters and variables are given in Table I.

TABLE I
PARAMETERS AND VARIABLES FOR THE TRANSDUCER USED.

Parameter	Value	Unit	Matlab variable
Center frequency	8	MHz	xdc.f0
Number of elements	128	elements	xdc.n_elements
Pitch	0.3	mm	xdc.pitch
Element height	4	mm	xdc.height
Elevation focus	20	mm	xdc.elevation_focus
Element positions	[x y z]	m	xdc.element_positions(128 x 3)
Impulse response	[samples]	$Pa/(Vs^2)$	para.xdc.impulse_response

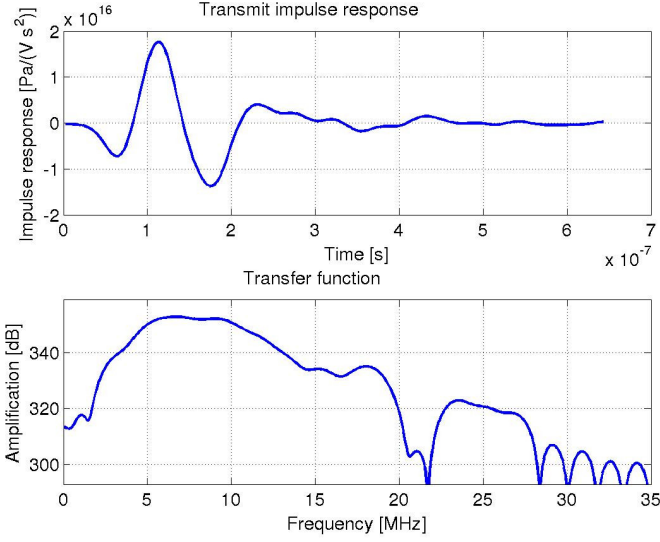


Fig. 1. Impulse response of the probe.

B. Excitation

The pulse waveform consist of a tapered 3 cycles sinusoidal waveform. The excitation waveform, sampled at 70 MHz is shown in Fig. 2 and is also included in the parameter structure described in Section III-A.

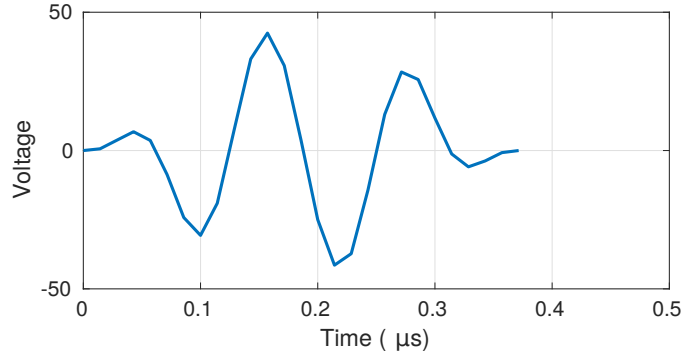


Fig. 2. Excitation waveform.

C. Duplex SA sequence

The provided data is acquired with a linear array probe using a duplex sequence (both B-mode and flow). The sequence is repeated to generate continuous data, so that the data can be beamformed throughout the whole imaging region, and data can be made continuously.

The imaging sequence is as follows:

Emission number:	1	2	3	4	5	6	7	8-11	12	13	14 - 767	768
Emission type:	V_1	V_2	V_3	V_4	V_5	B_1	V_1	... V_5	B_2	V_1	... V_5	B_{128}

where V_1 is a velocity emission for virtual source 1 and V_5 is the emission equal to the number of velocity emissions. B_2 is a B-mode emissions for virtual source 2. The inter-spaced B-mode sequence is 128 emissions long and starts again with B_1 , when the last B-mode emission has been made, thus, having a frame size of 768 emissions. A summary of the transmit sequence parameters are listed in Table II. The precise location of the specific

virtual sources are found in the parameter file *parameters_matlab.mat* and are described in section III-A. It should be noted that all parameters are in SI units, e.g. length in meters and time in seconds.

TABLE II
ACQUISITION/SIMULATION PARAMETERS

Transmit Parameters		
Parameter	B-mode	Flow
Number of emitting elements	16	64
Tx Apodization window		Hanning
F-number	-1	-3.5
Number of distinct beams	128	5
Excitation signal	3 sinusoids with a 50% Tukey window	
Pulse repetition frequency (PRF)	5 kHz	

The placement of the various virtual emission centers are shown in Fig. 3. Note that some of the emissions for the B-mode image is only shown as a dot, and only the first 5 velocity emissions are shown. The following ones are placed at exactly the same locations.

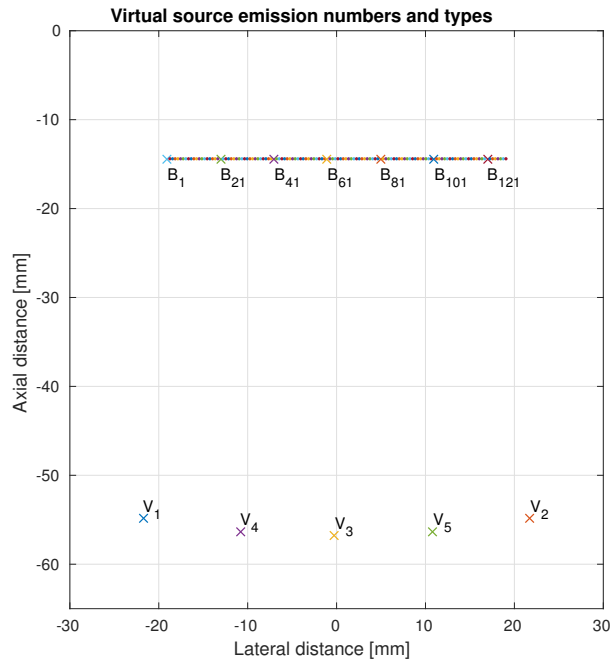


Fig. 3. Placement of virtual sources in the imaging sequence.

The insonified area of virtual source 51, used for B-mode imaging, is shown in Fig. 4 to the left. The placement of the transducer elements are shown in blue and the active elements for the emission are colored red. The two lines roughly denote the acceptance angle for the virtual source, so that the emitted energy is present within the two lines. Similarly the virtual source 4 for blood flow velocity estimation is shown to the right in Fig. 4.

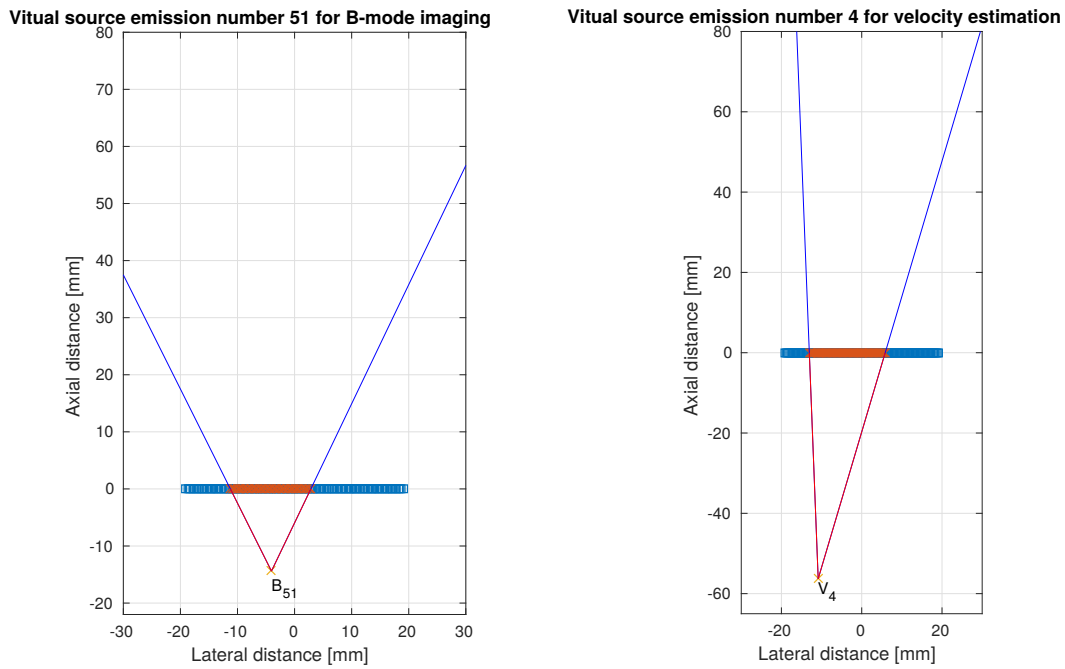


Fig. 4. Virtual source for B-mode imaging (left). Virtual source for velocity estimation (right). The inactive transducer elements are indicated in blue and the active elements are shown in red.

III. SUPPLIED DATA

The supplied data file is found through the Zenodo platform ([10.5281/zenodo.1045795](https://zenodo.org/record/1045795)) and the folder organization is as follows:

- /database
 - wire_phantom
 - straight_vessel_90deg
 - straight_vessel_75deg
 - spinning_disk
 - CFD
- /example_results
 - straight_vessel_90deg
 - straight_vessel_75deg
 - spinning_disk
 - CFD
- /code
 - evaluation
 - image_beamformation
 - src
 - velocity_estimation
- /evaluation

The pre-beamformed data sets are in */database*. The data is stored in MATLAB files and contains all the information needed to beamform the data sets, including the transducer geometry and virtual sources position. The data format and folder structure is fully described in section III-A.

The folder */example_results* provides a set of estimated velocity results for each of the phantoms stored in */database*. The results submitted by the participants must replicate the folder organization and data format provided

in this folder to be accepted by the platform. The provided example results contain the required spatial-temporal grids for each of the data sets.

The example and evaluation code are located in the folder */code*. The example code consist on a simple beamformer and a directional velocity estimator[5]. The evaluation code contains the algorithms used by the platform to evaluate the participants, which are describe in Section IV. The evaluation code reads, by default, the results stored in the folder */example_results*, and outputs the processed results in the folder */evaluation*.

A. Data format

The sampled element RF data are stored at */database/data/PHANTOM_NAME/element_data*. Data from both the B-mode and flow sequence are provided, and are located under their respective folder with the same name. The data consist of 10 frames, each frame is stored in the folder *seq_xxxx*.

The sampled element RF data are stored for each individual emission as an *elem_data_em0000.mat* file, where 0000 indicates the emission index. The file contains a single variable sampled array with dimension (number of samples) x (number of transducer elements). The data is stored as uint16, where 2048 corresponds to zero in the measurements. It should be noted that in the measurements there can be a slight DC offset, and this should be removed, either by subtraction or match filtration.

All parameters for a given data file are found in the file *parameters_matlab.mat* in the main directory for the data set. It should be noted that all parameters are in SI units, e.g. length in meters and time in seconds. The top level content of the structures are defined in Table III. The individual structures are defined in the following Tables.

TABLE III
STRUCTURES IN THE *parameters_matlab.mat* FILE.

Structure name	Content
para.sys	Overall variables for the image.
para.xdc	Definition of the transducer.
para.tx	Transmissions used in the image.
para.rx	Reception variables for the image.

TABLE IV
PARAMETERS AND VARIABLES FOR THE SYSTEM.

Parameter	Value	Unit	Matlab variable
Sampling frequency	35	MHz	para.sys.sampling_frequency
Speed of sound	1540-1480	m/s	para.sys.c
Number of frames	10		para.sys.no_acquired_frames
Pulse repetition frequency	5000	Hz	para.sys.fprf

The different transmissions are described in the *para.tx* structure, which is listed in Table V. Note here that the excitation waveform is sampled at 70 MHz, twice the receive sampling frequency.

The reception of the data is described in *para.rx*. This gives the start and stop time for sampling of the received data after excitation of the first element in the active aperture.

TABLE V
PARAMETERS AND VARIABLES FOR THE TRANSMISSIONS.

Parameter	Value	Unit	Matlab variable
Emission number in sequence			para.tx.emission_number
Definition of the virtual transmit source			para.tx.virt_source(n)
Active elements used in virtual source			para.tx.virt_source(n).elements()
Delays used in virtual source		s	para.tx.virt_source(n).delays()
Apodization used in virtual source			para.tx.virt_source(n).weights()
Center for virtual source	[x y z]	m	para.tx.virt_source(n).apert_center(1:3)
Transmit delay for virtual source			para.tx.virt_source(n).tx_delay
Wave type for virtual source	spherical		para.tx.virt_source(n).wave_type
Excitation for this emission	Waveform samples		para.tx.excitation{n}
Type of emission	flow/bmode		para.tx.type{n}
Excitation waveform used for this emission	flow/bmode		para.tx.excitation{n}

TABLE VI
PARAMETERS AND VARIABLES FOR THE RECEPTION.

Parameter	Value	Unit	Matlab variable
Start time for the sampling		s	para.rx.record(n).start_time
Stop time for the sampling		s	para.rx.record(n).end_time

The first sample in the data has a starting time that can be calculated as:

```
% Find which virtual source to use
virt_source_index=flow.emission_placement(i);

% Find the correct starting time relative to the center of the active aperture
virt_pos=para.tx.virt_source(virt_source_index).apert_center;

% Find the center element for emission and calculate the distance to this
Ne=size(para.tx.virt_source(virt_source_index).elements,2);
center_element = para.tx.virt_source(virt_source_index).elements(floor(Ne/2));
ele_pos = (para.xdc.element_positions(center_element,:) ...
    + para.xdc.element_positions(center_element+1,:))*0.5;
delay_index=floor(Ne/2);
phase_delay_value=(para.tx.virt_source(virt_source_index).delays(delay_index)...
    +para.tx.virt_source(virt_source_index).delays(delay_index+1))*0.5;
data_start_time = sqrt(sum((virt_pos-ele_pos).^2))/para.sys.c+...
    para.rx.record(virt_source_index).start_time-phase_delay_value;
```

This calculation find the center position of the emitting active aperture and calculates the distance from the virtual source to this element. It then compensates for the transmission focusing delay and the starting time of the sampling in the SARUS system. This starting time can then be used in the focusing of the data. To get full alignment between a point scatterer position and the beamformed data, it is also needed to compensate for the length of the ultrasound convolved with the matched filter employed.

B. Data sets

The competitors are provided with 9 data sets. The first stage data sets are released together with the reference data sets through the Zenodo platform([10.5281/zenodo.1045795](https://zenodo.org/record/1045795)). The participants are also provided with processing examples scripts. This allow the participants to start working with the data and get started before the platform opens. The data sets released in this first stage must be submitted to gain access to the second stage data. The second stage data is randomized from a set of available CFD simulations for each participant. The exact reference velocities are only be revealed to the automatic evaluation scripts and at the end of the challenge. The evaluation scripts follow the ones present in the folder `/code/evaluation` and detailed in Section IV. The participant will be able to update the results at any time before the platform closes.

The data sets consist of 9 sets: two beamforming reference phantoms, a CFD model example, four straight vessel phantoms, one spinning disk and a randomized CFD model. A description of the data sets are given in Table VII

TABLE VII
DATA SETS PROVIDED IN THE COMPETITION.

No	Content	Description
Reference		
1	Simulated point spread phantom	A wire phantom containing 5 wires at depths of 35, 60, 85, 110, and 135 mm are simulated in Field II using the SA sequence. This is used to test the basic beamforming algorithm.
2	Measured point spread phantom	Wire phantom measurements are also given to the participants for validation of their beamforming.
3	Simulated CFD model	Simulated pulsatile flow in a carotid bifurcation based on CFD flow model developed by Swillens et al[6]. No noise is added.
1st stage		
4-5	Simulated parabolic profile	Data set simulated using Field II, where all parameters and the ground truth are known. An example of axial beamformation and axial velocity estimation for this data set are also posted on the web-site. The beam-to-flow angles are 75 and 90 degrees. Stationary echoes are included from the vessel boundary and are 40 dB larger than the blood scattering. The signal-to-noise ratio in the vessel is 15 dB.
6-7	Measured parabolic profile	A set of data measured on a flow rig with a stationary, parabolic profile. Two data sets at angles of 75 and 90 degrees are given. The ground truth is given in the form of a parabolic profile and its spatial position.
8	Simulated spinning disk	Data set simulated using Field II, where all parameters and the ground truth are known. The phantom rotates clock-wise, having a diameter of 1.5 cm, and a maximum velocity $v_{max} = 0.25ms^{-1}$ at the outer edge. The center is placed at a depth of 2.5 cm below the center line of the array. The elevation extent of the spinning disk is 5 mm. White noise is added to obtain a SNR of 10 in dB.
2nd stage		
9	Randomized CFD model	Finite Element simulated complex flow in a carotid bifurcation based on CFD flow model. A section of the cardiac cycle is simulated including peak systole and vortices. Field II is used for simulating the received ultrasound signal from both blood and tissue. Stationary echoes are included from the vessel boundary and are 40 dB larger than the blood scattering. Noise is added to the signal and the signal-to-noise ratio for individual RF element signals in the vessel are 15 dB. Part of a cardiac cycle around the peak systole is given to the contestants. The depth position of the vessel and its orientation (rotation) is not revealed. The data is blinded to both the contestants and the committee by randomizing the exact position, orientation and inlet velocity magnitude within a set of values defined by the committee. The ground truth is only revealed to the participants after the platform closes.

IV. EVALUATION IN THE COMPETITION

The participants uploads their results using the web platform as a zip file, following the exact same folder structure as present in the folder `/example_results`. Velocity estimates are uploaded as x-y-z components, following the spatio-temporal grid defined in each `velocities.mat` using the `flow_grid` variable. The participants must store the estimated velocities components as a 4-Dimensional matrices with the same size as the provided grid. The

participant are referred to the folder */example_results*, for the velocity structure format and the specific grid for each phantom.

In general, the performance is evaluated using two metrics: 1) Global metrics, where the overall performance is evaluated by calculating the averaged standard deviation and the bias across a data set. 2) Locally, where the performance is evaluated using the local metric of peak velocity for data set #9. The different data sets are evaluated individually with respect to their global or local metric error, and zero being the best score achievable. The overall results will be the weighted average of the normalized metrics from the distinct phantoms. The weight distribution is considered as: 25% for vessel phantoms, 25% for the spinning disk, and 50% for the randomized CFD phantom.

A validation MATLAB code must also be submitted together with the results, and shall be able to reproduce the submitted results. The validation code will be executed at the discretion of the challenge committee to corroborate the results provided from the participants and to verify that the rules were followed, e.g. that the specified number of emissions used were correct. The validation will be done retrospectively, after the platform closes and its main purpose is to discourage cheating. All the results must be generated using the exact same velocity estimation algorithm (code), and the code must not include case/switch with phantom specific parameters. The code can be published once the competition is over, if the participant has given permission to code publication.

A. Global measurements

The performance is evaluated by calculating the averaged standard deviation and bias across a data set. The provided velocity components are first transformed to be represented as the velocity magnitudes and velocity angles. The magnitude is always positive and measured in ms^{-1} , while the angle is in degrees and have range of 0-360°.

The metrics are individually evaluated by using relative standard deviation and bias defined as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (\hat{m}(i) - m_{true}(i))^2}{Nm_{peak}^2}}; \quad m_{bias} = \frac{\sum_{i=1}^N (\hat{m}(i) - m_{true}(i))}{Nm_{peak}} \quad (1)$$

Here $\hat{m}(i)$ is the estimated metric by the participant, $m_{true}(i)$ is the ground truth, and m_{peak} is the maximum value attainable by the metric. The estimates are ordered in a vector that can be converted to a spatial and temporal position. The performance metrics are found for both the magnitude and angle. For the angle circular statistics are used to avoid aliasing around 360° angle. The participants are ranked according to an average of all four metrics. For the first stage data sets, the evaluation would also weight the indices by the number of emission sequences used for finding the velocities

$$\sigma_{weighted} = \sigma \sqrt{1 + \frac{N_{emissions}}{5}} \quad (2)$$

where $N_{emissions}$ is the number of flow emission used for calculating the estimate (averaging). The number of emissions used must be provided by the participant and will be verified using the validation code.

B. Local measurements

The peak velocity is evaluated only on data set #9, the maximum velocity from the simulated plane is used as the evaluation metric and the bias percentage is estimated as

$$v_{peak_{bias}} = \frac{\hat{v}_{peak} - v_{peak}}{v_{peak}}, \quad (3)$$

where \hat{v}_{peak} is the measured peak velocity of all estimated velocities.

C. Validation code

A validation code must be submitted together with the results. The code must be self-sufficient, meaning that it must contain all the supporting code needed to generate the estimates with a standard version of MATLAB with the Toolboxes listed in Appendix A. The code must be able to generate a velocity estimate at an arbitrary point in time and space from any specified data set path. The selection of the point and path must be clearly identified at the beginning of the *main.m* script, which will be the runnable script. Any results that are not verifiable using this code will be disqualified.

V. PROCESSING EXAMPLES

The processing examples are in the `/code` folder, where beamforming and velocity estimation examples are described. This code is meant to help participants get started with the data.

A. Beamforming example

A code example for beamforming the reference wire data sets 1 and 2 are provided at `/code/image_beamformation`. The code produces beamformed contour images for wires present at a depth of 31 and 56 mm as the ones shown in Fig. 5. The images shows contour plots with 6 dB between contours for both simulated and measured data sets. The measured wires on the right consists of a pair of twisted copper wires, which are difficult to align exactly. They are therefore off-set laterally and in depths, and two peaks are seen from the twisted wires. The dataset and sequence to beamform is selectable in the code example.

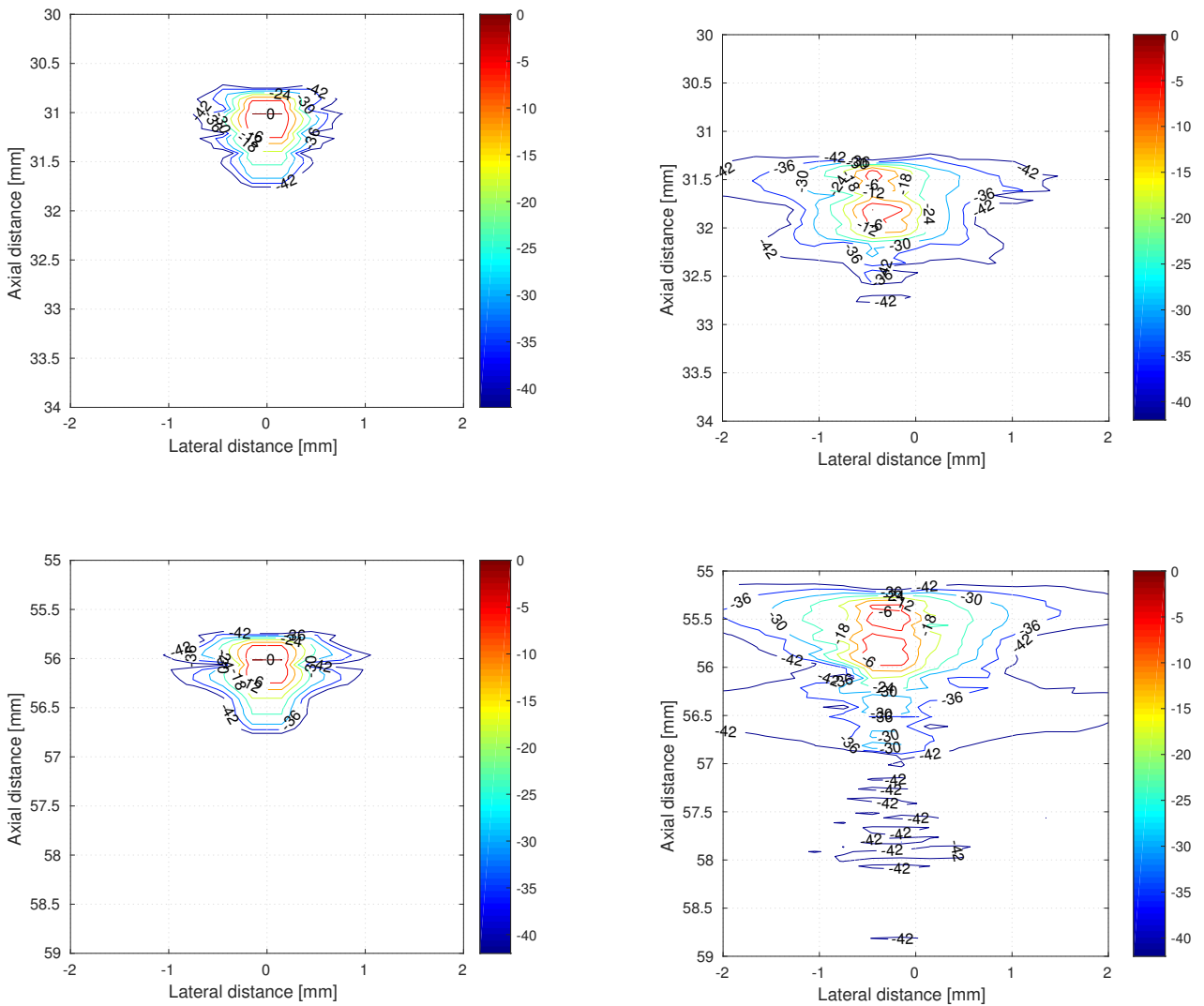


Fig. 5. Beamformed images of the wires in data #1 and #2 for the B-mode emissions. The contour plots for the simulated data set are shown on the left and for the measured data sets on the right. There are 6 dB between the contours.

The beamformed responses for the flow sequence are shown in Fig. 6. Higher side lobes are seen since fewer emissions are used.

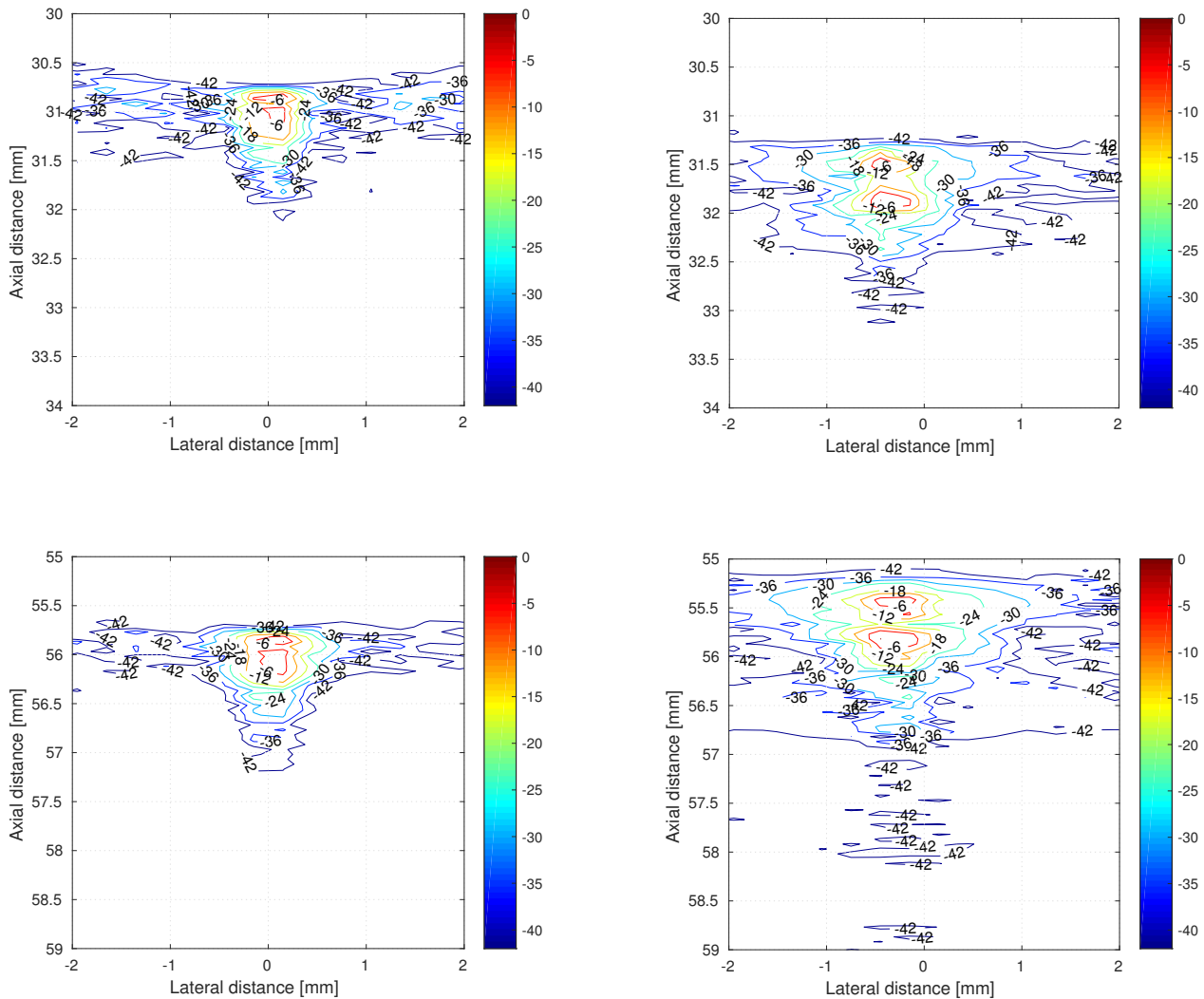


Fig. 6. Beamformed images of the wires in data #1 and #2 for the five flow emissions. The contour plots for the simulated data set are shown on the left and for the measured data sets on the right. There are 6 dB between the contours.

B. Velocity estimation example

A code example for estimating velocity magnitude using directional beamforming and cross-correlation, as described in [5], is found at `/code/velocity_estimation`. The code estimates the velocity magnitude provided the given angle at the selected points. The code is just meant as an example, and does not include an angle estimator. The participant shall estimate both the angle and magnitude from the provided data. An example of the generated flow profile with data set #4 is shown in Fig. 7

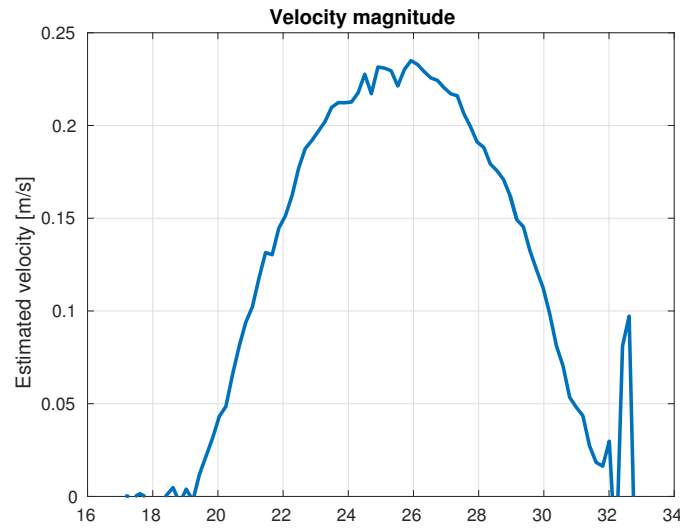


Fig. 7. Profile plots for straight vessel phantom #4, single magnitude profile is generated with example code.

VI. EVALUATION EXAMPLES

The evaluation code are in the folder `/code/evaluation`, where scripts are provided for each phantom type. This code is meant to provide the participants a stand-alone method to evaluate their results before the platform opens. The code generate the metrics and additional plots to show the participant the overall performance of their method. A set of example results are also provided for the flow data sets in folder `/example_results`. By default, the evaluation scripts read these results and produces both a numerical and a graphical result that is stored at `/evaluation`.

A. Straight vessel

An example plot for the `/example_results` of data set #4 is shown in Fig. 8. The plot provides a graphical results for the mean, standard deviation and true value for the evaluated magnitude and angle; while the numerical results used by the platform to score the participants is written in a text file.

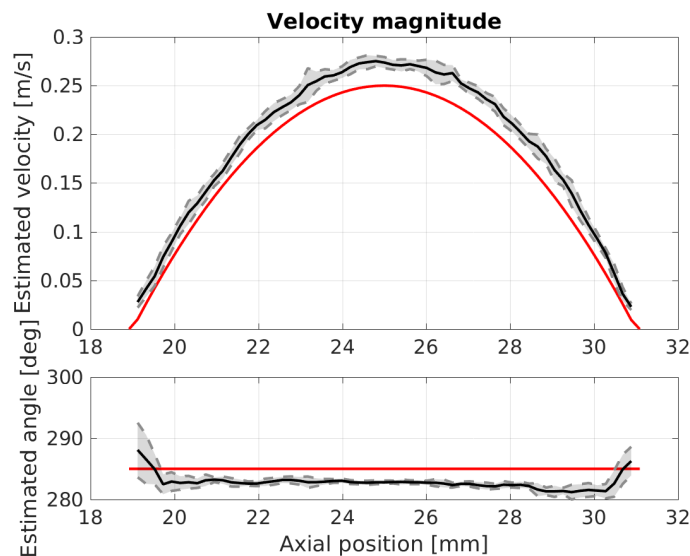


Fig. 8. Profile plots for the provided example result for the straight vessel phantom #4. The true profiles are shown as the red line, while the mean and standard deviation are shown in black and grey, respectively.

B. Spinning disk

An example evaluation plot for data set #8 in */example_results* is shown in Fig. 9. The angles and magnitude errors are plotted to provide a better insight on the method performance. However, the numerical metrics are estimated as an overall performance along the whole phantom.

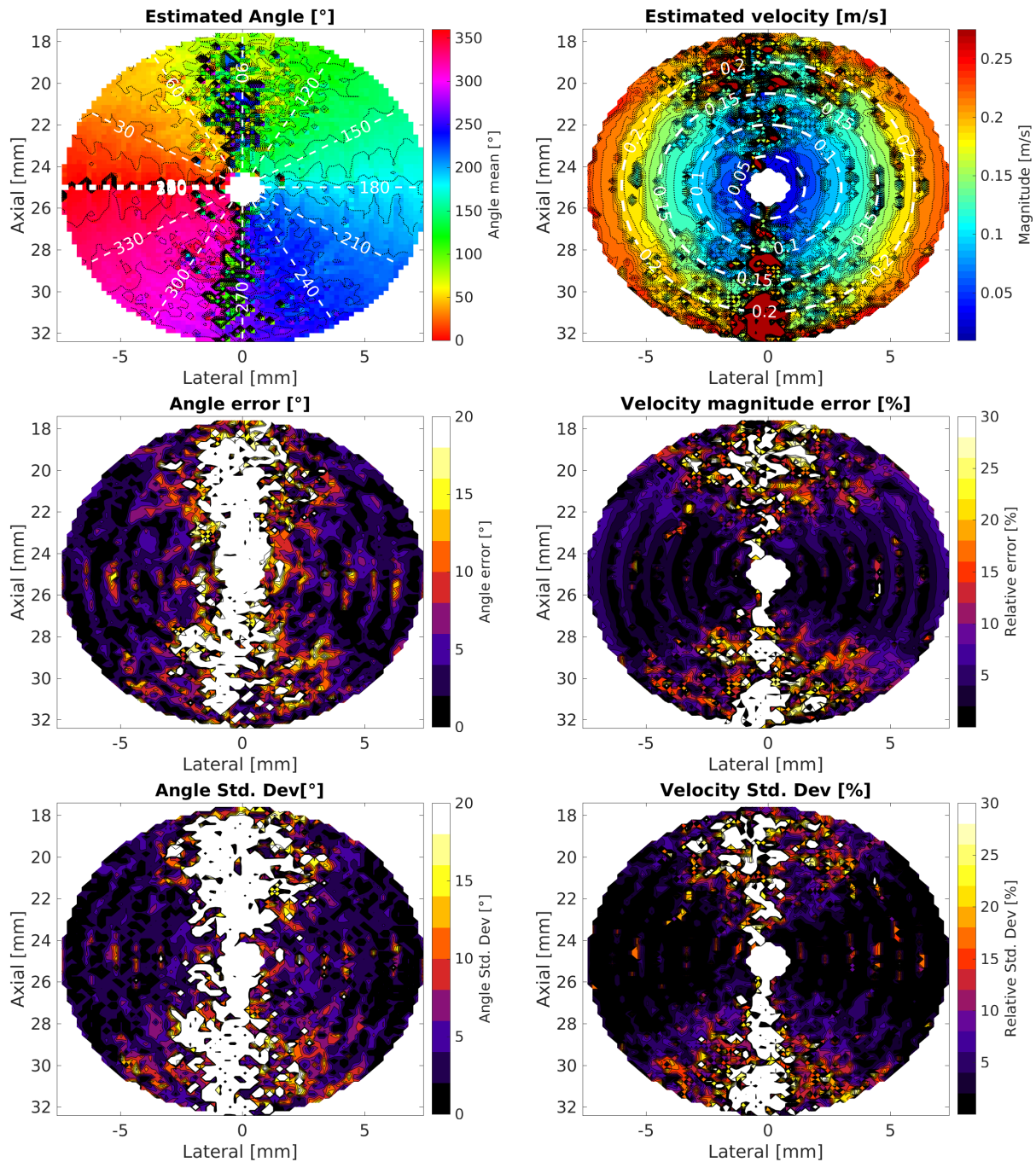


Fig. 9. Plots generated using the evaluation code on the spinning disk example results (data set #8). On the left column: The estimated angle, angle error, and angle standard deviation. On the right column: The velocity magnitude, relative velocity error, and relative velocity standard deviation.

C. CFD

An example evaluation plot for reference data set #3 in */example_results* is shown in Fig. 10. Data set #3 is not used for evaluation in the challenge, however, results from the data provided in the second stage will be evaluated using the same metrics. The results are evaluated against the ground truth provided by the velocity field from the CFD model.

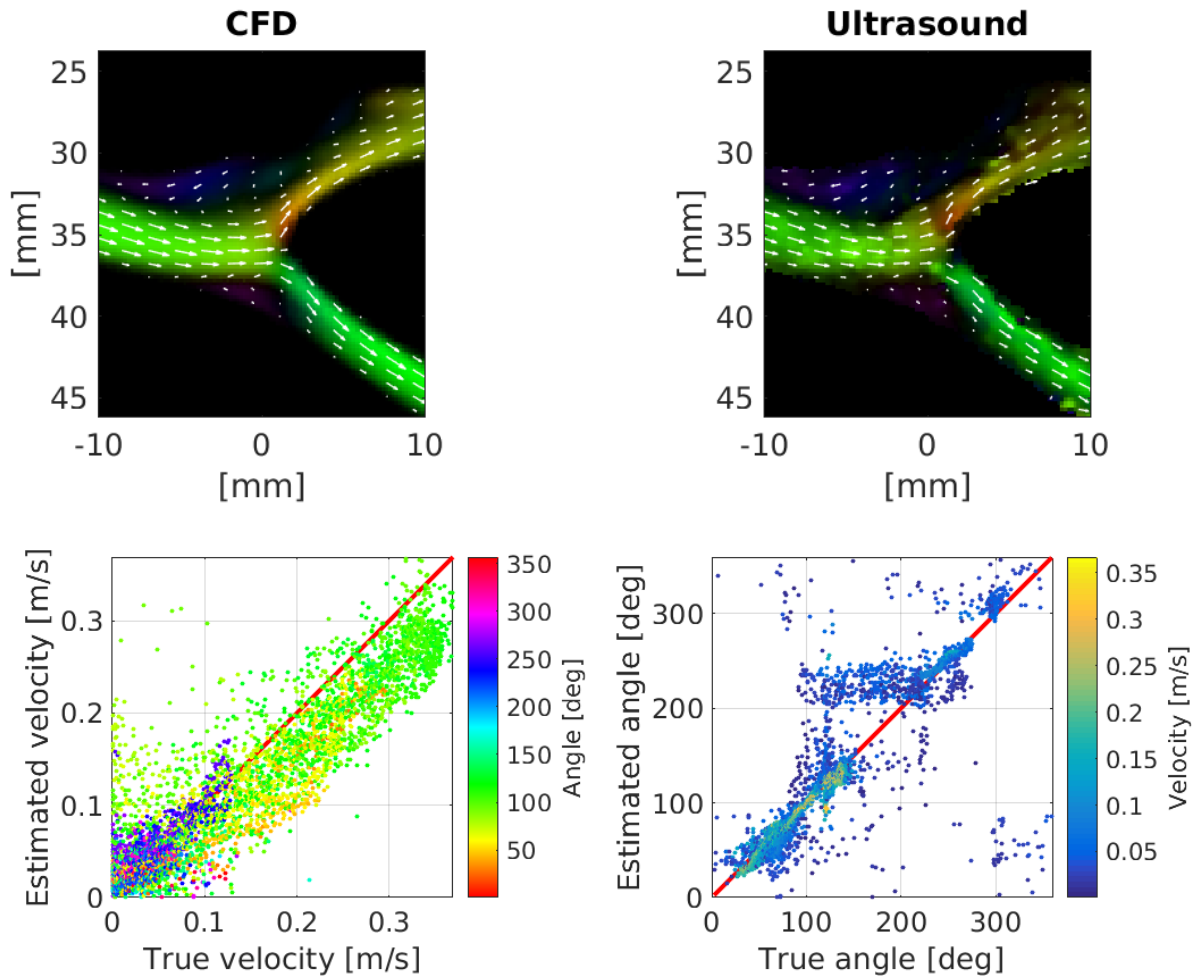


Fig. 10. Plots generated using the evaluation code on CFD example results. [Top] Vector flow imaging (VFI) frames from the reference CFD model (left), and the estimated velocity fields from ultrasound (right) during late systole. [Bottom] Scatter plots comparing the reference CFD values to the ultrasound estimates. (Left) The scatter plot of the estimated velocities color coded with the estimated angle. (Right)

REFERENCES

- [1] H. Liebgott, A. Rodriguez-Molares, F. Cervenansky, J.A. Jensen, and O. Bernard. Plane-wave imaging challenge in medical ultrasound. In *Proc. IEEE Ultrason. Symp.*, pages 1–4, 2016.
- [2] J. A. Jensen and N. B. Svendsen. Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 39:262–267, 1992.
- [3] 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, 1996.
- [4] J. A. Jensen. Safety assessment of advanced imaging sequences, II: Simulations. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 63(1):120–127, 2016.
- [5] J. A. Jensen. Directional velocity estimation using focusing along the flow direction: I: Theory and simulation. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 50:857–872, 2003.
- [6] A. Swillens, L. Løvstakken, J. Kips, H. Torp, and P. Segers. Ultrasound simulation of complex flow velocity fields based on computational fluid dynamics. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 56(3):546–556, 2009.

APPENDIX

A. Supported MATLAB and Toolboxes

MATLAB Version 8.6 (R2015b)

Computer Vision Toolbox Version Version 7.0 (R2015b)

Image Processing Toolbox Version 9.3 (R2015b)

Signal Processing Toolbox Version 7.1 (R2015b)

Wavelet Toolbox Version 4.15 (R2015b)