# 2018 IEEE IUS SA-VFI Challenge

Authors: Jørgen Arendt Jensen and Carlos Armando Villagómez-Hoyos
Reviewers: Hervé Liebgott, Olivier Bernard, and Alfonso Rodriguez-Molares
Version 1.2, December 12, 2017

## CONTENTS

# I. INTRODUCTION

Given the positive response to the IUS 2016 challenge on plane wave imaging (PICMUS)[1], we propose to organize a challenge on synthetic aperture (SA) 2-D vector flow imaging (VFI). Synthetic aperture, as plane wave imaging, is a parallel system technique that consist on utilizing broader beams to insonify a larger region, e.g spherical waves. The advantages of utilizing such parallel systems for estimating blood flow are many, foremost, they break the tie between frame rate, region of interest (ROI), and precision of the estimates. In the literature, there exist a variety of parallel systems techniques that are capable of estimating the velocity of the blood flow[2]. The purpose of the challenge is to provide a framework where the techniques can be easily compared given a synthetic aperture sequence and a set of data sets. The challenge consist on estimating blood flow velocities from both simulated and measured ultrasound RF element data. 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 participants results verification 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 SA-VFI challenge download page. A reference wire phantom data set is also provided to give the participant the possibility to check that the correct beamforming process is in place.

In January 2018, 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 is available at the SA-VFI challenge website, where they can receive additional input from organizers and other users. As soon as the platform opens in February 2018, participants must create an account. The participant must then upload their results from the first stage data set as specified in Section IV, together with their verification 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 are 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. 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 [3], [4], 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 [5]. 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 |

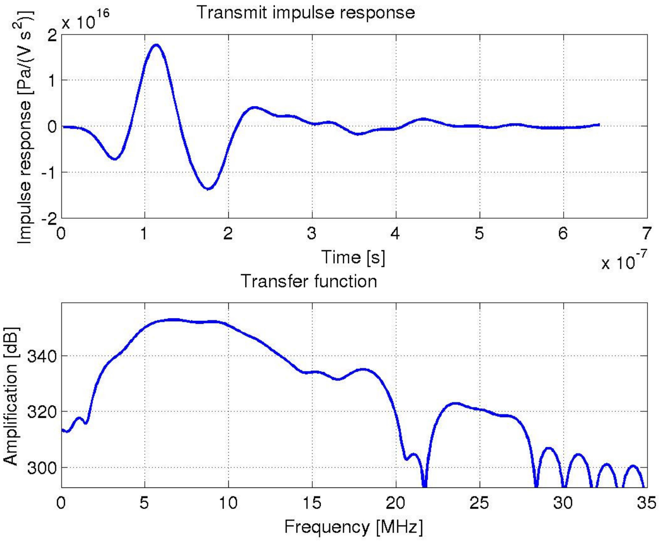

Fig. 1. Impulse response of the probe.

### 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.

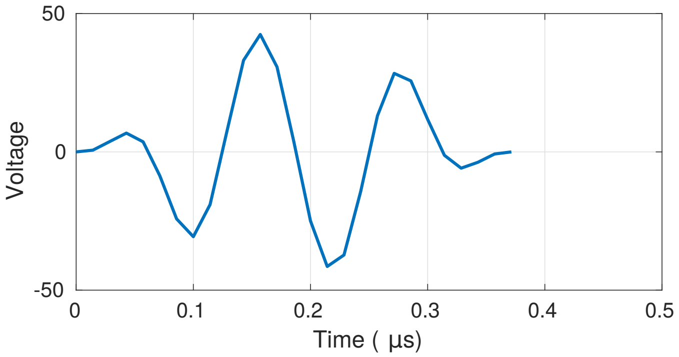

Fig. 2. Excitation waveform.

### *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 location of the 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 are only shown as a dot, and only the first 5 velocity emissions are shown. The following ones are placed at the same locations.

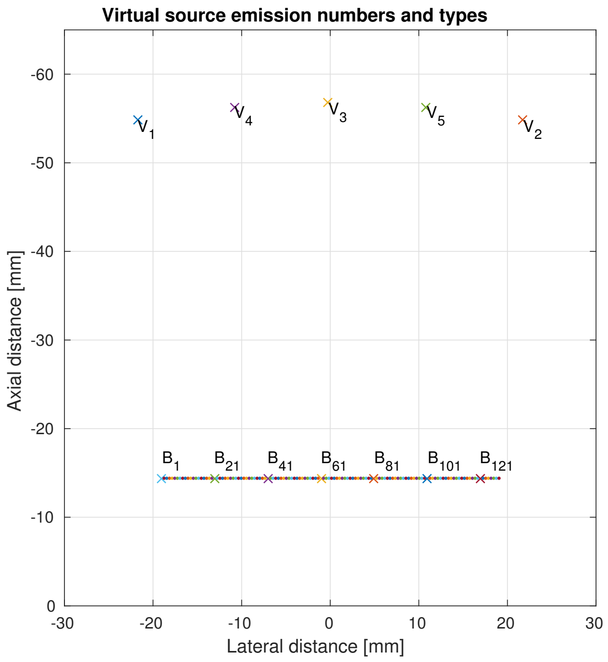

Fig. 3. Placement of virtual sources in the imaging sequence.

The insonified area of virtual source #51, used for B-mode imaging, is shown in Fig. 4 (left). The placement of the transducer elements are shown in blue and the active elements for the emission are colored red. The two lines delineates the acceptance angle for the virtual source, so that the insonified area is present within them. Similarly the virtual source #1 for blood flow velocity estimation is shown in Fig. 4 (right).

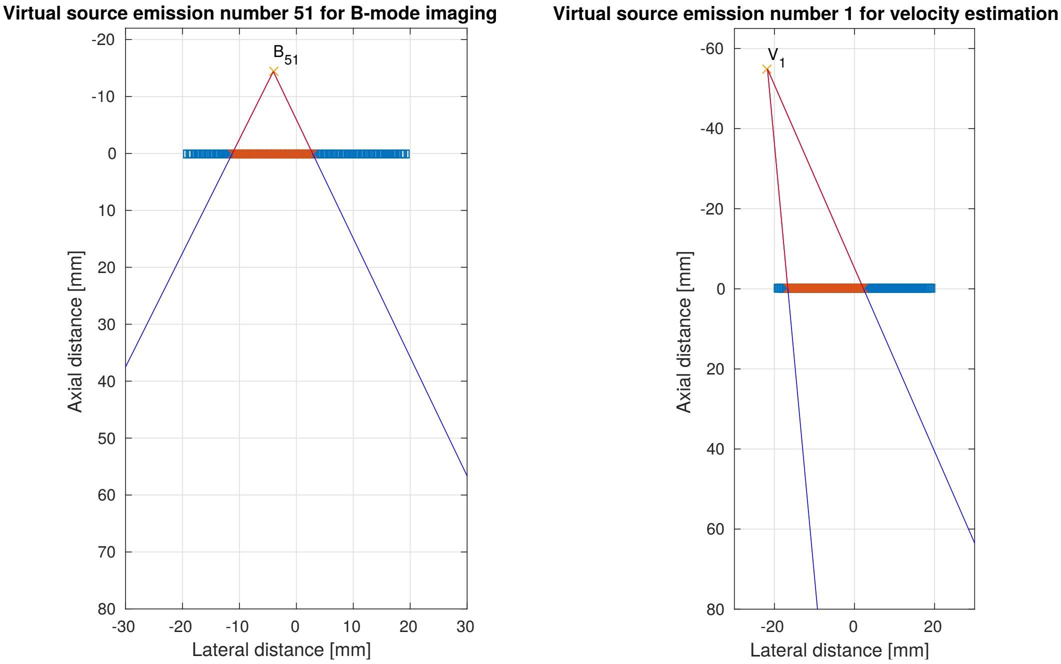

Fig. 4. Virtual source for B-mode imaging (left). Virtual source for velocity estimation (right). The inactive transducer elements are indicated in blue and the active elements are shown in red.

## III. SUPPLIED DATA

The supplied data sets are found through the SA-VFI challenge website. The participants are provided with 8 data sets distributed in 5 zip files, as listed in TableIII. The ninth data set is provided by the platform during the challenge and is from a randomized set of available CFD simulations models. To start working with the data and get started, the participants can also download a set of processing examples scripts from the same web-sites. The use of the example code is detailed in Section V.

TABLE III
DATA SETS PROVIDED IN THE COMPETITION.

| No | Content | Description |
|---|---|---|
| **Reference & 1st stage** | | |
| 1-2 | Wire phantom | A wire phantom containing 5 wires at depths of 31, 56, 81, 106, and 131 mm are both simulated in Field II, and measured with using the SA sequence. The phantoms are meant to be used as reference to verify beamforming algorithms. |
| 3 | Carotid bifurcation model | Simulated pulsatile flow in a carotid bifurcation based on CFD flow model developed by Swillens et al[7]. No noise is added. The phantom is meant as reference for the 2nd stage of the challenge. |
| 4-5 | Straight vessel at **105**° | Parabolic profile straight vessel phantoms with a beam-to-flow angle of **105**°are both simulated and measured. The simulated data set is obtained using Field II, where 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. The measured data set is obtained from a flow rig circulating blood-mimicking fluid. The length of the entrance tube is long enough to ensure fully-developed laminar flow with a parabolic profile. |
| 6-7 | Straight vessel at **90**° | A set of simulated and measured data for a straight vessel phantom with a beam-to-flow angle of **90**°. The same characteristics as data sets # 4-5 are used. |
| 8 | Spinning disk | Data set simulated using Field II. 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.<br>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. |

### A. Data format

The RF channel data are stored for each individual emission as a single *elem_data_em0000.mat* file, where *0000* indicates the emission number. The file contains a single variable sampled array with dimension ($N_{samples}$ x $N_{elements}$). The data is stored as uint16, where 2048 corresponds to zero in the measurements. It should be noted that the channel signals might come with a slight DC offset, and this should be removed, either by subtraction, band pass filtering or match filtration.

The RF channel data are stored at */SAVFI-data/PHANTOM_NAME/element_data*. The data are structured as follows:

- **/bmode**
    - seq_0001
        - elem_data_em0001.mat
        - ...
        - elem_data_em0128.mat
    - ...
    - seq_0005
- **/flow**
    - seq_0001
        - elem_data_em0001.mat
        - ...
        - elem_data_em0640.mat
    - ...
    - seq_0005

The data are stored in a per frame basis, where each frame is stored in the their respective folder *seq_0000*. A frame consist of 768 emissions as detailed in Sec. II-C, from which 128 are B-mode and 640 are flow emissions. The B-mode and flow sequence are stored separately under their respective folder, where *elem_data_em0001.mat* under the B-mode folder corresponds to the sixth emission of the frame. The five emissions on the SA flow sequence are repeated 128 times during the frame to complete a frame, hereby, 640 emissions are stored under the folder *seq_0000*. The frames are repeated after each other without any delay between frames, effectively acquiring continuous flow data.

### *B. Data parameters*

All parameters for a given data file are found in the file *parameters_matlab.mat* in the root of each 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 IV. The individual structures are defined in the following Tables.

TABLE IV
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 V
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 VI. 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 VI
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 VII
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 pulse convolved with the matched filter employed.

## IV. EVALUATION IN THE COMPETITION

The participants uploads their results to the web platform as a MAT and PNG files. The required result files are generated by using the scripts located inside the folder */SAVFI-code/evaluation* and are stored by default in the folder */evaluation_results*. Velocity estimates are stored as a MATLAB structure with x-y-z components as the field names. The velocity components are arranged as 4-Dimensional matrices following the pre-defined spatio-temporal grid. The reference grids are obtained from their respective phantom at the folder */SAVFI-code/reference_grids*, each contains a flow_grid variable inside the *"ref_grid.mat"* file. The participants are also provided with a set of example results at the folder */SAVFI-code/example_results*, which should help on understanding the format.

In general, 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 data sets are evaluated individually with respect to their global or local metric error, and zero being the best score achievable. The overall result for the first stage will be the average of the normalized metrics from the distinct phantoms. The second stage will be solely evaluated on the randomized CFD data set metrics. Recognition will be awarded for both stages, however, contestants should use the same code for both challenge stages.

### 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 meters per second, 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:

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

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

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

where $N_{emissions}$ is the number of flow emission used for calculating the estimate (averaging or filtering). The number of emissions used must be provided by the participant and must be of the size of the larger temporal filter applied to the estimates. The $N_{emissions}$ will be verified using the verification code after the challenge has closed.

### 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

$$Bias_{peak} = \frac{\hat{v}_{peak} - v_{peak}}{v_{peak}}, \tag{3}$$

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

### C. *Verification code*

A verification MATLAB code must also be submitted together with the results to reproduce the submitted results. The verification code will be executed at the discretion of the challenge committee to corroborate the provided results and verify that it complies with the challenge rules, e.g. that the specified number of emissions used were correct. The validation will be done retrospectively, after the platform closes. All the results must be generated using the exact same velocity estimation algorithm (code) with the exact same parameters. It is not allowed to tweak the algorithm for each specific phantom. The code can be published once the competition is over, if the participant has given permission for it.

The verification code must be submitted together with the results. The code must be stand-alone, meaning that it must contain all the supporting code needed to generate the estimates. The verification should be able to run on a 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. The code can be submitted as a MEX file if it is compiled to run on a Linux operating system.

## V. PROCESSING EXAMPLES

The processing examples code are located in the folder */SAVFI-code*, it contains a beamformation exampled and velocity estimation examples. The code is meant to help participants get started with the data.

### A. *Beamforming example*

A code example for beamforming the data sets is provided at */SAVFI-code/image_beamformation*. The beamformed B-mode straight vessel with a beam-to-flow angle of 90°for both simulated and measured data sets are shown in the top Fig. 5. In the same manner the B-mode for the spinning disk and carotid bifurcation model are shown in the bottom Fig. 5

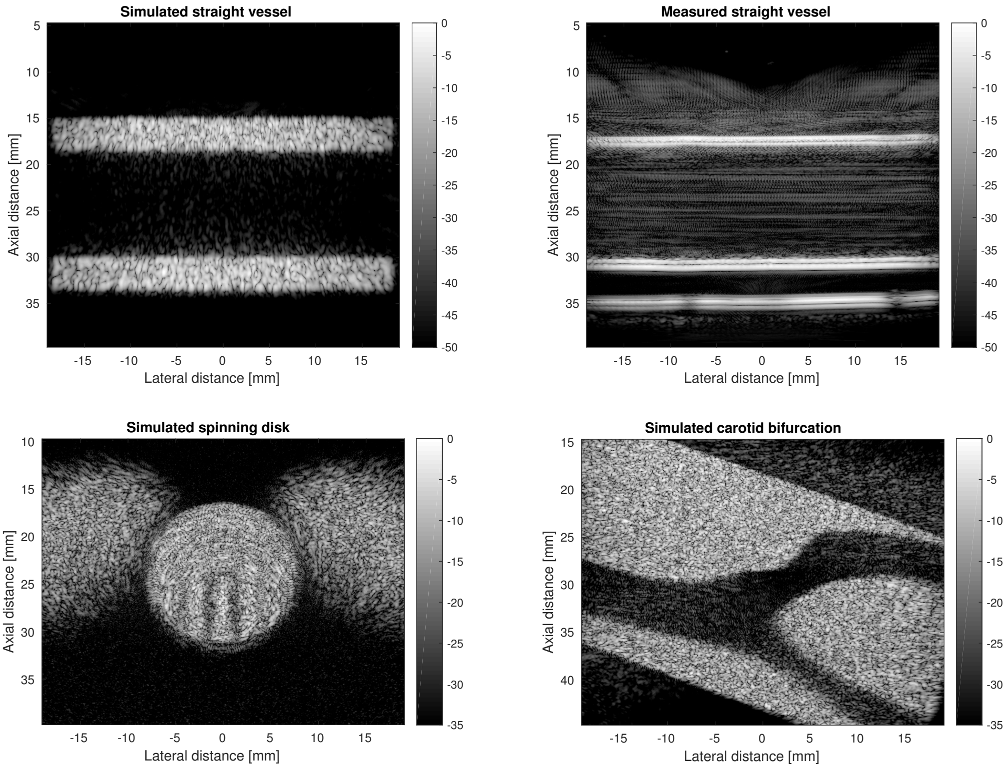

Fig. 5. Beamformed images of the data sets #6, #7, #8 and #3 for the B-mode emissions.

The performance of a beamformer is better shown as the point spread function of a wire phantom. Therefore, a wire phantom is provided as a reference. The example code beamforms a selected wire and produces contour plots, as the ones shown in Fig. 6 and in Fig. 7. The contour plots are within 6 dB intervals for both simulated and measured data sets for the first and last wire. The measured wires on the right consists of a pair of twisted copper wires. Higher side lobes are present in the flow sequence since since fewer emissions are used. The dataset and sequence to beamform are selectable in the code example.

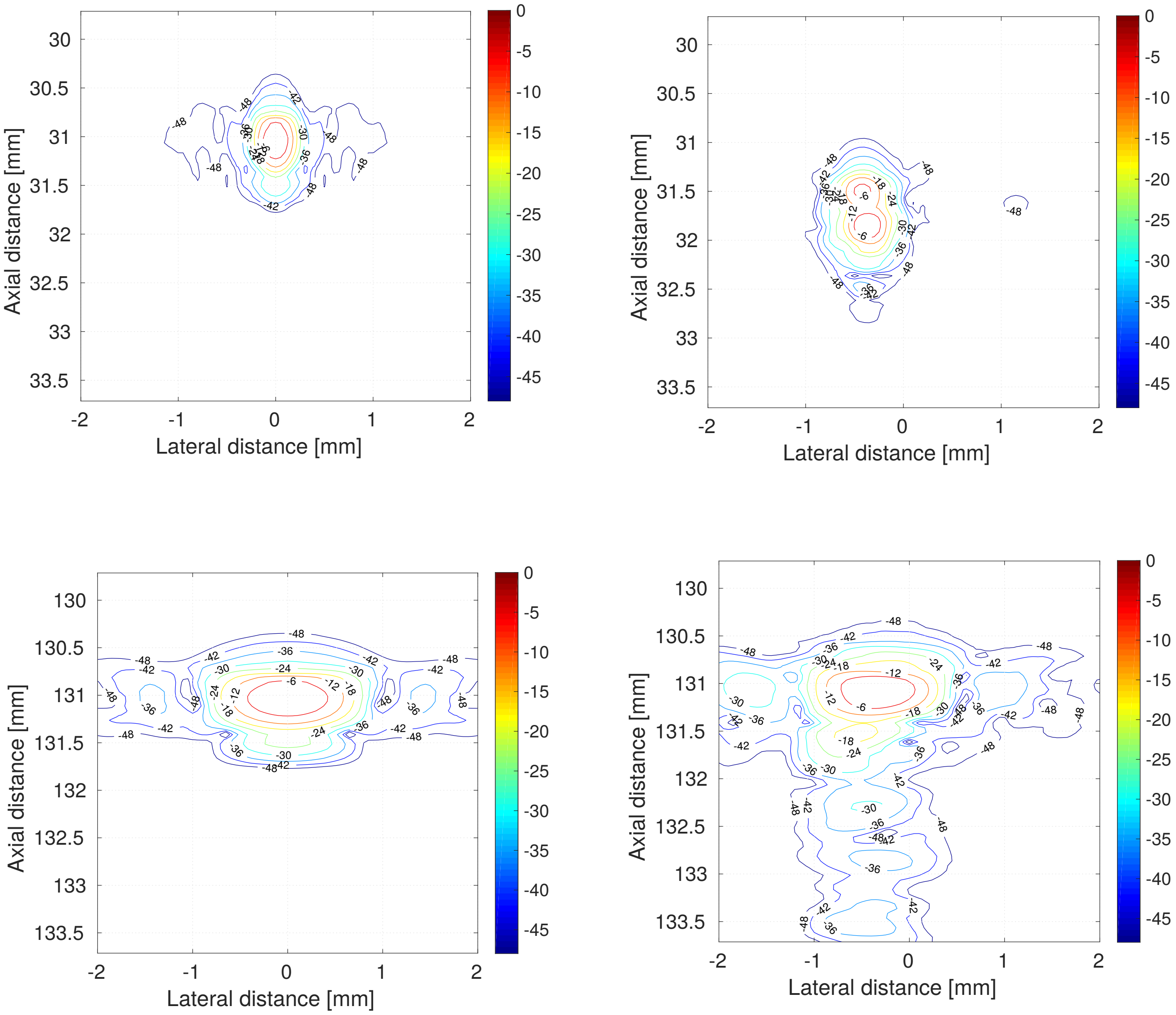

Fig. 6. Beamformed images of the wires in data #1 and #2 for the B-mode emissions. The contour plots for the simulated data set are shown on the left and for the measured data sets on the right. There are 6 dB between the contours.

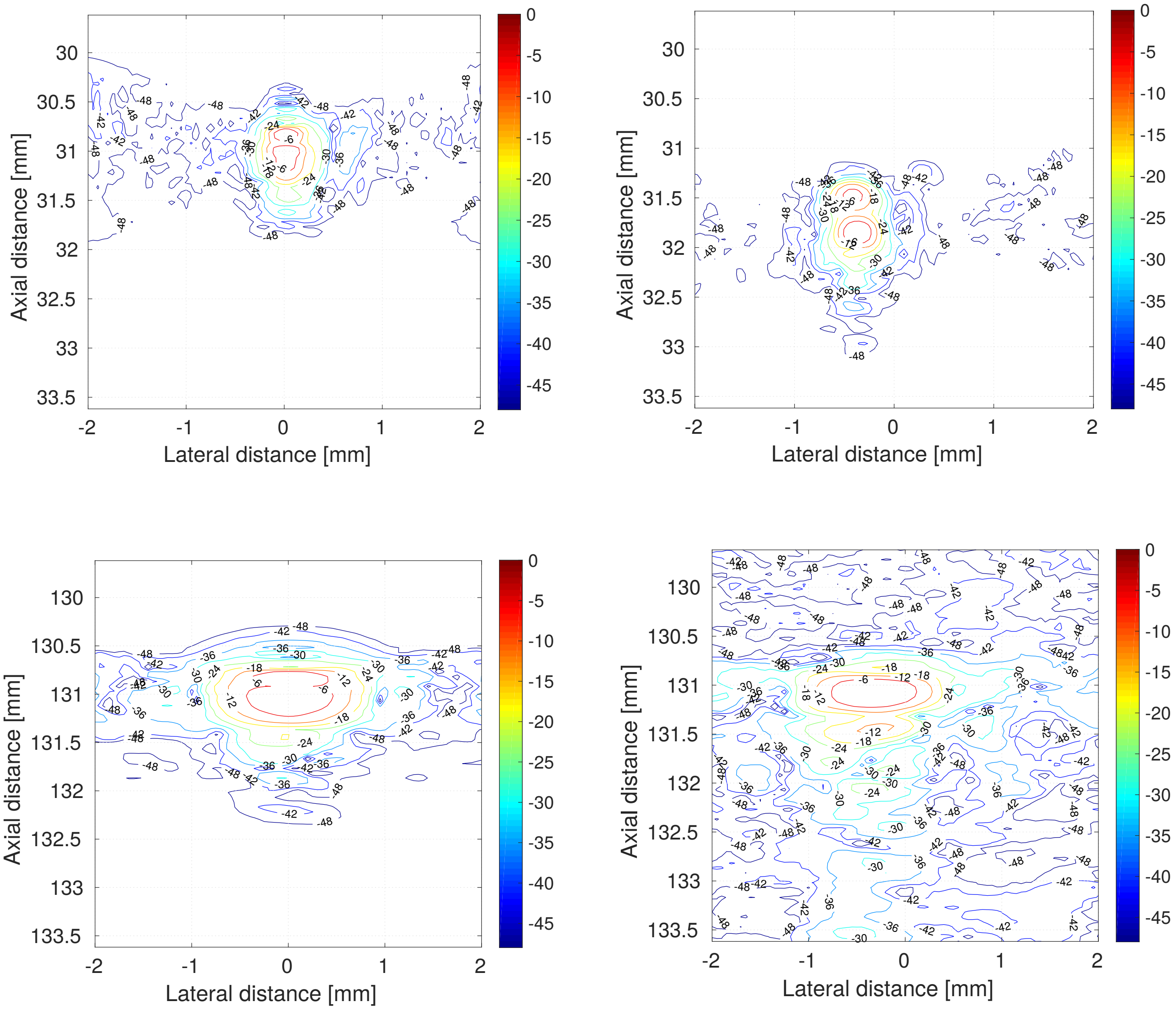

Fig. 7. Beamformed images of the wires in data #1 and #2 for the five flow emissions. The contour plots for the simulated data set are shown on the left and for the measured data sets on the right. There are 6 dB between the contours.

### *B. Velocity estimation example*

A code example for estimating velocity magnitude using directional beamforming and cross-correlation, as described in [6], is found at */code/velocity_estimation*. The code estimates the velocity magnitude provided the given angle at the selected points. A simple angle estimator is also included as part as the example code. The codes are meant as examples and the the participant shall proposed their own velocity estimators for both the angle and magnitude. An example of the generated flow profile with data set #4 is shown in Fig. 8

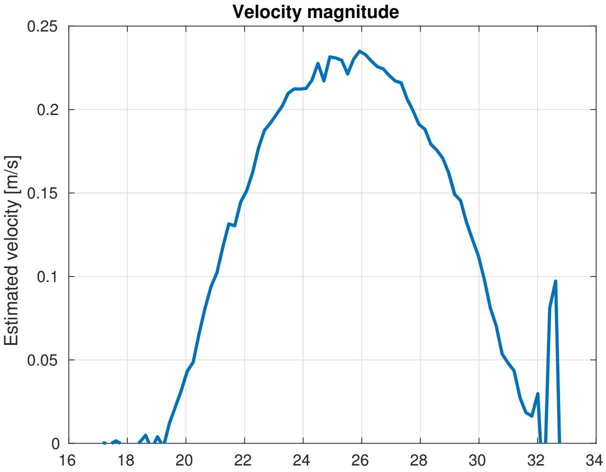

Fig. 8. Profile plots for straight vessel phantom #4, single magnitude profile is generated with example code.

## VI. Evaluation examples

The evaluation code are in the folder */SAVFI-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 */SAVFI-code/example_results*. By default, the evaluation scripts read these results and produce both a numerical and a graphical result that is stored at */evaluation_results*.

### A. Straight vessel

An example plot for the */example_results* of data set #4 is shown in Fig. 9. 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 are displayed in the screen.

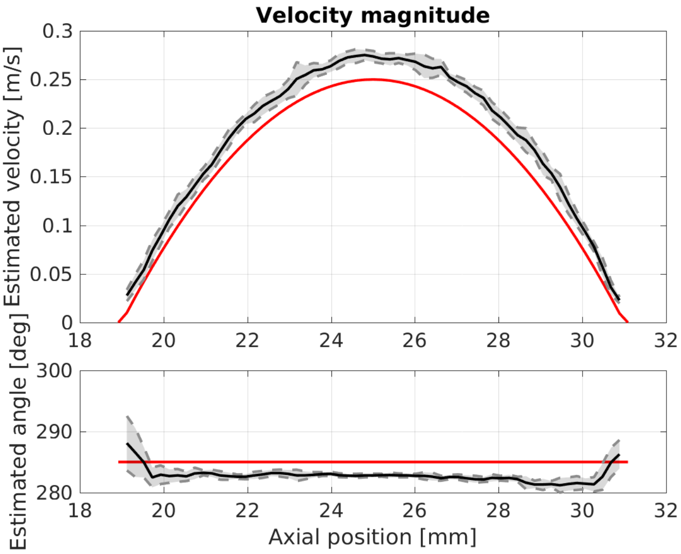

Fig. 9. Profile plots for the provided example result for the straight vessel phantom #4. The true profiles are shown as the red line, while the mean and standard deviation are shown in black and grey, respectively.

### B. Spinning disk

An example evaluation plot for data set #8 in */example_results* is shown in Fig. 10. 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.

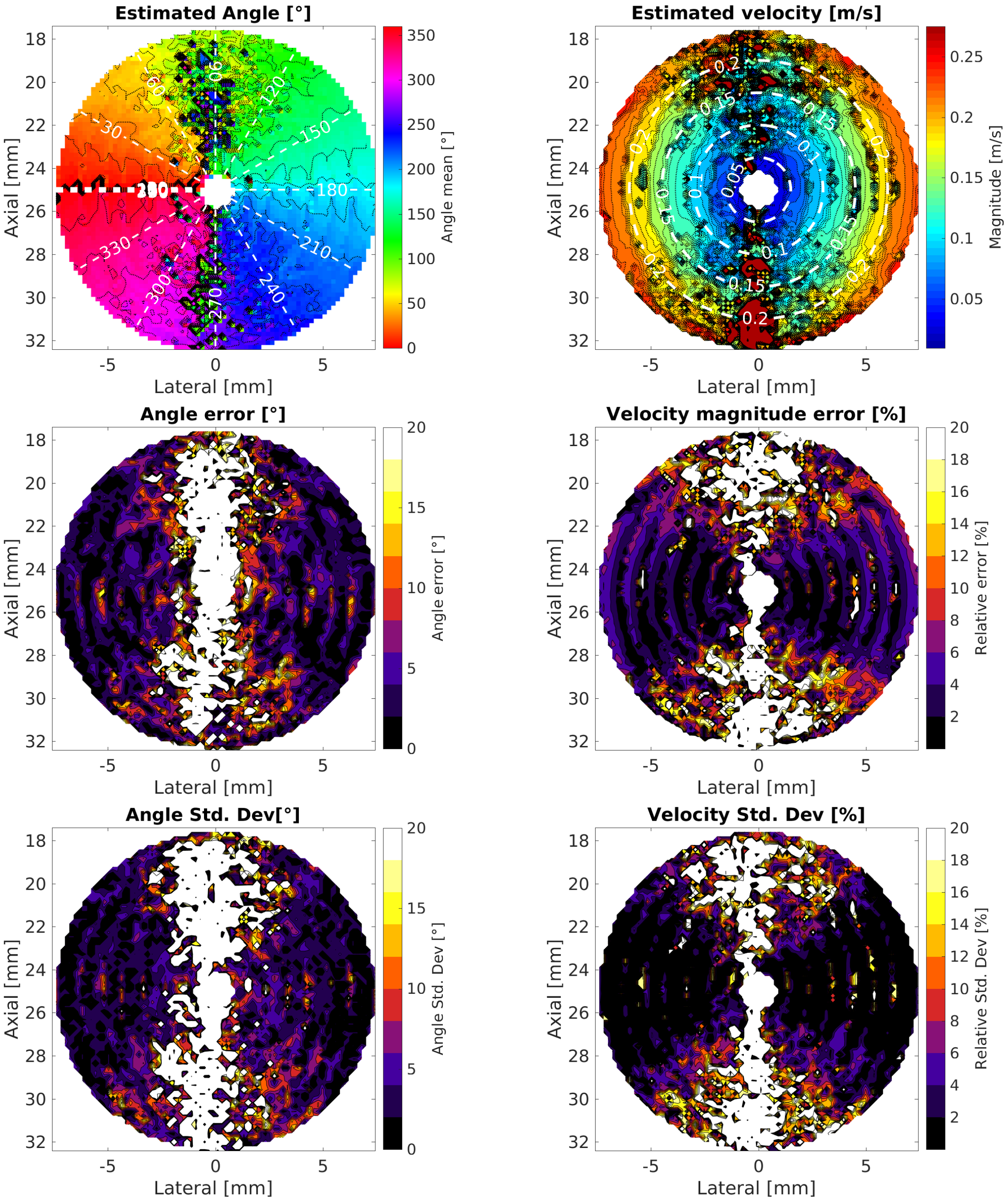

Fig. 10. Plots generated using the evaluation code on the spinning disk example results (data set #8). On the left column: The estimated angle, angle error, and angle standard deviation. On the right column: The velocity magnitude, relative velocity error, and relative velocity standard deviation.

### C. CFD

An example evaluation plot for reference data set #3 in */example_results* is shown in Fig. 11. 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.

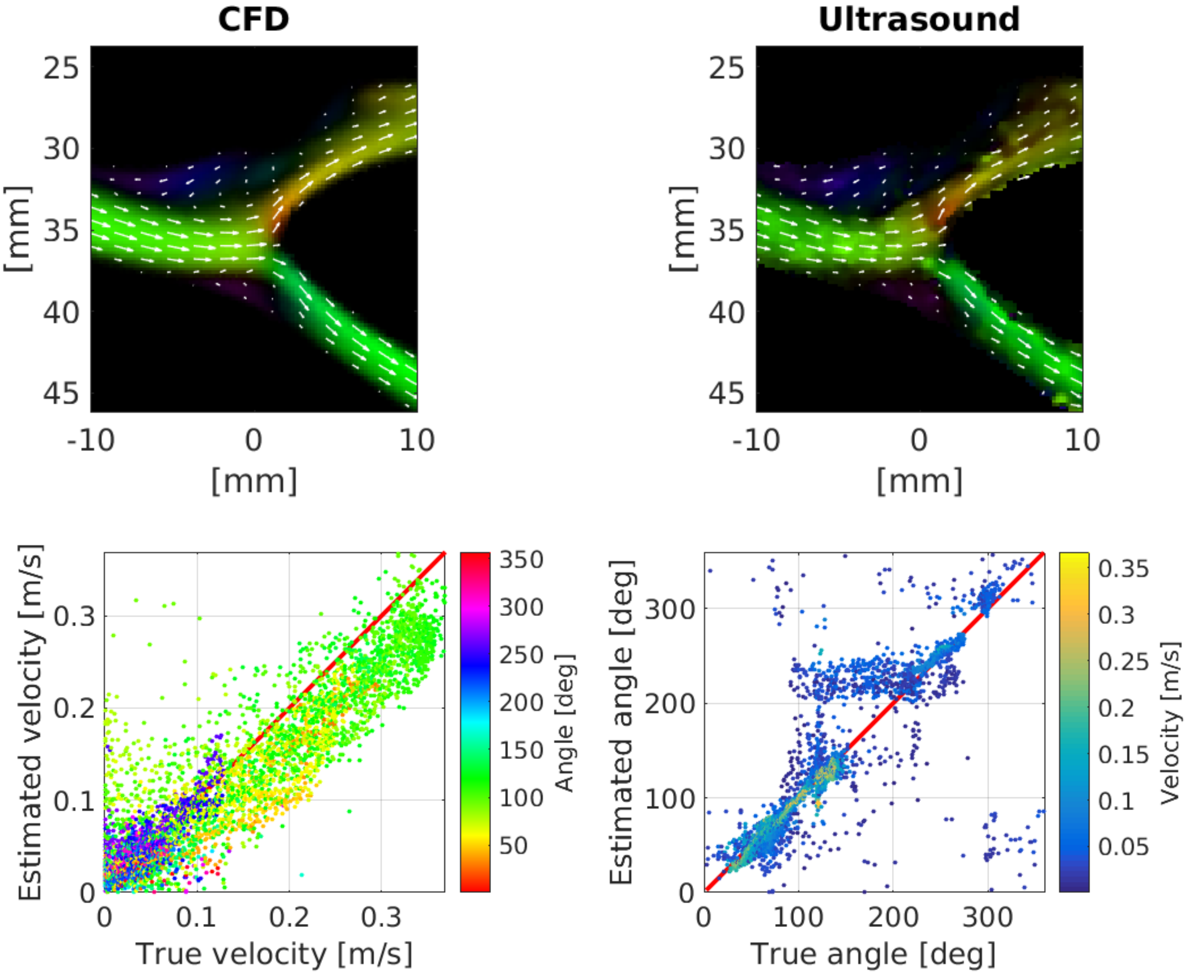

Fig. 11. Plots generated using the evaluation code on CFD example results. [Top] Vector flow imaging (VFI) frames from the reference CFD model (left), and the estimated velocity fields from ultrasound (right) during late systole. [Bottom] Scatter plots comparing the reference CFD values to the ultrasound 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, S. I. Nikolov, A. Yu, and D. Garcia. Ultrasound vector flow imaging II: Parallel systems. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 63(11):1722–1732, 2016.

[3] 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.

[4] 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.

[5] J. A. Jensen. Safety assessment of advanced imaging sequences, II: Simulations. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 63(1):120–127, 2016.

[6] 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.

[7] 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)*