PyMUST: an open-source Python library for the simulation and analysis of ultrasound

Background, Motivation and Objective

Computer simulators based on the physics of acoustic waves transmissions can be used to create controlled environments with known ground truth. This enables quantitative verification of acquisition settings and analysis algorithms in different scanners without need of physical phantoms. One of such simulators is the Matlab UltraSound Toolbox (MUST, www.biomecardio.com/MUST), a toolbox for simulating ultrasound acquisitions and analyzing radiofrequency signals. MUST implementation is close to the mathematical formulation, is modular and has an extensive documentation and in-code references. However, it requires a MATLAB license, limiting its accessibility. To cope with this limitation, we present PyMUST, a Python library covering the same functionalities in MUST.

Statement of Contribution/Methods

PyMUST source code is in github (anonymous.4open.science/r/PyMUST-5F73), and can be directly installed. The library is composed of different functions, which are consistent with MUST and and have the same signature, thus benefiting from the original Matlab documentation. It includes functions for generating synthetic radiofrequencies and obtain the acoustic pressure distribution for given transducer activations, the processing of the radiofrequency signals for the creation of images, including demodulation, time-gain-compensation and beamforming using the delay-and-sum-matrix, as well as velocity estimation using phase-shift Doppler or speckle tacking from sequential acquisitions. To obtain a high computational performance, our library uses the numerical python package (Numpy) for fast vector/matrix operations. Our library also permits acceleration using several CPUs using threads via the multiprocessing Python package.

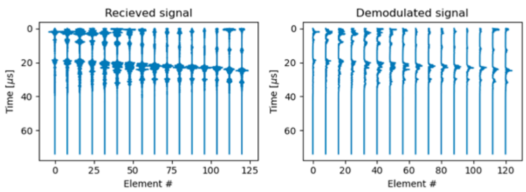
Results/Discussion

Figure a) shows the lines of code for simulating an acquisition of a phantom and generating a B-mode. In subfigure b) the simulated received and demodulated radiofrequency. Finally, subfigure c) shows the reconstructed B-mode image.

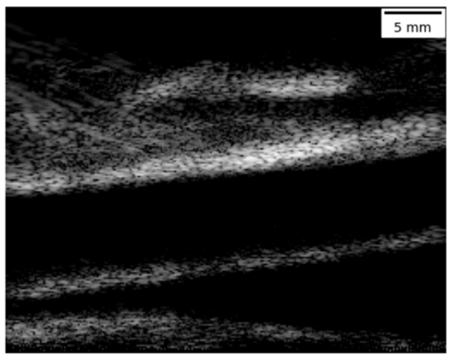
We expect that our Python library will democratize the use of ultrasound simulators, as Python is freely available the most popular introductory programming language at universities. The medical image analysis community will benefit from this library, as they primarily work with Python, and they need realistic synthetic data for training machine learning models.

```python
param = pymust.getparam('L11-5v')
param.attenuation = 0.5
xs,_,zs,RC = pymust.genscat([5e-2, np.nan],
            1540/param.fc,I,.4)

L = (param.Nelements-1)*param.pitch
dels = pymust.txdelayPlane(param, 0)
RF,_ = pymust.simus(xs,zs,RC,dels,param)

IQ = pymust.rf2iq(RF,param.fs,param.fc);
xi,zi = np.meshgrid(np.linspace(-L/2,L/2,256),
                    np.linspace(1e-9,3e-2,
                    int(256/L*3e-2)))

param.fnumber = None;
M = pymust.dasmtx(IQ,xi,zi,param)
IQb = pymust.utils.applyDasMTX(M, IQ, xi.shape)
B = pymust.bmode(IQb)
```

a)



b)



c)