Design of an In-Vitro Set-up for Sonothrombolysis of human blood clots using microbubbles

J O V A N A   J A N J I C

Supervisor at UNIPD: Alfredo Ruggeri
Supervisors at KTH: Anna Bjällmark, Malin Larsson

Acknowledgments

I would like to start by thanking my supervisors Anna Bjällmark and Malin Larsson, for the help, guidance and constructive comments in all the stages of the thesis work.

Special thanks also to all my friends that have shared with me the invaluable experience of Erasmus, supporting me in all the difficult moments.

Last but not the least, I wish to express my love and gratitude to my family and to my boyfriend Enrico for their understanding and endless love through the duration of my studies.

ii

Abstract

Several studies suggest that the use of ultrasound in conjunction with microbubbles (MBs) can induce the lysis of the blood clots through acoustic cavitation and through the production of microjets and microstreaming. However, there is no accordance about the optimal ultrasound parameters that have to be considered in order to achieve the maximum thrombolytic effect, neither a clear agreement about the type of MBs that have to be used.

This project had two main goals: the design and optimization of an in-vitro set-up for the study of clot lysis within coronary arteries and its testing with ultrasound in conjunction with two different types of MBs. The MBs considered were the 3MiCRON MBs and the SonoVue MBs.

The ultrasound sequence was developed using a programmable ultrasound architecture (Verasonics, Inc) and was tested using commercially available clinical transducers.

Using the designed set-up and varying the ultrasound parameters (frequency, pulse length and pulse amplitude) it was possible to study the clot lysis efficiency in conjunction with the two types of MBs. For the 3MiCRON MBs no increase in clot lysis was found with the implemented ultrasound parameters, while considering the SonoVue MBs, a 10% increase in clot lysis was found with 10ms long pulse delivered at 50V (peak-to peak value).

The obtained set-up had several aspects in common with the real situation of occluded coronary arteries, although some limitations were present and further optimizations are required.

Further work is required in order to assess if different combination of ultrasound parameters are able to lead to an increase in clot lysis when delivered with 3MiCRON or SonoVue MBs.

Sommario

Diversi studi suggeriscono che l'utilizzo di ultrasuoni e di microbolle (agenti di contrasto) può indurre la lisi di trombi sanguigni attraverso il fenomeno della cavitazione acustica e tramite la produzione di microgetti e microvortici. Tuttavia, non vi è un comune accordo su quali siano i parametri ultrasonori ottimali nè le microbolle da utilizzare al fine di ottenere il massimo effetto trombolitico.

Questo lavoro di tesi presenta due obiettivi fondamentali: la progettazione ed ottimizzazione di un sistema in-vitro per lo studio della lisi dei trombi all'interno delle arterie coronarie e l'utilizzo di tale sistema in concomitanza con l'applicazione di ultrasuoni e di microbolle. Due tipi di microbolle sono state prese in esame: 3MiCRON e SonoVue.

La sequenza di onde ultrasonore è stata sviluppata tramite un'architettura programmabile (Verasonics, Inc) ed è stata trasmessa usando trasduttori clinici presenti in commercio .

Utilizzando il sistema sviluppato e variando i parametri ultrasonori (frequenza, lunghezza e ampiezza dell'impulso) è stato possibile studiare l'efficienza trombolitica degli ultrasuoni con i due tipi di microbolle. Applicando le microbolle 3MiCRON non è stato riscontrato alcun miglioramento nella lisi dei trombi, mentre considerando le microbolle SonoVue è stato notato un aumento della trombolisi del 10%, trasmettendo impulsi ultrasonori della durata di 10ms a 50V (valore picco-picco).

Il sistema progettato ha diversi aspetti in comune con la situazione reale di occlusione delle arterie coronarie, sebbene siano presenti alcune limitazioni e ulteriori studi di ottimizzazione siano necessari.

Studi aggiuntivi sono inoltre necessari al fine di chiarire se diverse combinazioni di parametri ultrasonori siano capaci di condurre ad un aumento della trombolisi in combinazione con i due tipi di microbolle considerati.

Contents

List of Notations

SD Standard deviation

MB microbubble

t-PA   tissue-plasminogen activator

rt-PA   recombinant tissue plasminogen activator

HIFU   High Intensity Focused Ultrasound

λ wavelength

Introduction

Vascular thrombosis is the formation of clots of different sizes within the vessels. The treatment of this disease can be done pharmacologically or through an intravascular device (e.g. inflatable balloon), but both of these strategies produce serious shortcomings such as bleeding (e.g. intracerebral hemorrhage in the treatment of ischemic stroke) and blood vessels damage. Moreover, they do not achieve optimal reperfusion of the microvasculature.

Several studies suggest that the use of ultrasound can induce the lysis of the clots through acoustic cavitation and through the production of microjets and microstreamings. This application of US for the treatment of clots and for their reduction is defined as sonothrombolysis.

In addition, intravenously injected microbubbles (MBs) combined with ultrasound can further improve thrombolysis, lowering the cavitation threshold.

There is no accordance about the optimal ultrasound parameters that have to be considered in order to achieve the maximum thrombolytic effect, neither a clear agreement about the type of MBs that have to be used. This is due to the unclear process of cavitation and rather undefined dynamics of MBs within and around the clots.

Aim of the Project

This project had two main goals:

1. The first goal was to design an efficient in-vitro set-up that enabled the study of the effects of ultrasound waves and MBs in the treatment of occluded vessels. The set-up had to reproduce the physiological situation of thrombosis in the coronary arteries and to allow estimation of sonothrombolysis efficiency. To achieve these aspects, optimization and validation processes were performed in order to obtain a set-up as reliable as possible.

2. The second goal was to assess the ability of MBs to improve sonothrombolysis in conjunction with a specific sequence of ultrasound pulses delivered using linear transducers. In particular, a novel type of MBs, developed within the framework of the european project 3MiCRON, was studied in order to assess its capacity to enhance the thrombolysis. These new MBs were compared with commercially available SonoVue bubbles. The ultrasound sequence was developed using a programmable ultrasound architecture that enabled to study and to implement algorithms with variable ultrasound parameters such as frequency, pulse length and amplitude.

Structure of the work

Chapter 1: the definition of thrombus is given together with the different types of disease that can cause. For each type of disease the actual treatments are also presented. In addition, a brief description of ultrasound principles and physics is presented together with a brief introduction to contrast agents and their use in ultrasound. Moreover, sonothrombolysis is defined together with its mechanisms of action, safety aspects and the different techniques of ultrasound delivery for clot lysis.

Chapter 2: the methods and the results for the design, optimization and validation of the in-vitro set-up are described.

Chapter 3: the methods and the results of the tests with ultrasound in conjunction with MBs are presented. All the tests were performed using the in-vitro set-up obtained after optimization and validation processes. Moreover, in this chapter, a description of the Verasonics system and of the implemented algorithm is provided.

Chapter 4: the results obtained are analyzed and discussed. In addition, a conclusion with study limitations and future developments is presented.

The MatLab code for pulse sequences programmed with Verasonics is given as an appendix.

Chapter 1

Background

1.1 Thrombosis

Ischemic heart disease and stroke, induced by vascular thrombosis, are the major causes of death in the high-income countries [1].

Normally, cardiovascular system maintains equilibrium between anticoagulant and procoagulant state. The first state is very important for the cells on the endothelium, whereas the second state is important when vessel damage occurs.

Anticoagulation is necessary for maintaining normal blood fluidity and is regulated by agents released by endothelial cells. These agents, called anticoagulant factors, are proteins able to interfere with the clotting process.

In the procoagulant state it is possible to distinguish two pathways that converge in a common pathway and lead to the formation of fibrin, a protein involved in blood clotting. These pathways are a serious of reactions during which specific enzymes are activated (they normally circulate in-activated) in a cascade-like behavior. These activated enzymes, called coagulation factors, catalyze different reactions leading to the formation of fibrin. The first pathway is the extrinsic one, which occurs when an injury to a blood vessel is present and is started by a particular protein called tissue factor. This factor activates several reactions until a factor named X, necessary for the common pathway, is activated. On the other side, the intrinsic pathway starts when contact between specific proteins and a negatively charged surface, such as bacteria surface, occurs. Again, similar reactions as in the extrinsic pathway follows leading to the activation of factor X. Factor X is the first enzyme of the common pathway, which leads to the release of thrombin, allowing the conversion of fibrinogen (a glycoprotein) in fibrin. The latter is the main constituent of blood plug and is the final result of the coagulation process.

When the equilibrium between procoagulant and anticoagulant states is damaged, pathological states occur. On one side, inadequate procoagulant state lead to the leakage of blood form the vascular system. On the other hand overactive coagulant state may lead to thrombosis and blood flow interruption.

A thrombus is an intravascular coagulation of blood with a site-dependent composition of fibrin, platelets, erythrocytes, leukocytes and serum [2]. Thrombus in arterial circulation contains a higher concentration of platelets, whereas venous thrombus is rich in fibrin. When a thrombus does not contain fibrin, it is called blood clot. Clots are made only of cells and they are more instable

and fragile than thrombi. Thrombus is in general produced in the intravascular site and it is the result of the coagulation cascade and the aggregation of platelets on the side of a vessel, while clots are semisolid mass that are the result of only the coagulation cascade. For example, clots can be obtained when blood is let to coagulate in a vial. Both thrombi and clots can be further distinguished in white (arterial) or red (venous), depending whether the coagulation processes (and, in the case of thrombus formation, the platelets aggregation) occur within arterial or venous blood. Within the intravascular site, each of the two types of thrombi, arterial and venous, can lead to different diseases.

If the thrombus breaks down, the small fragments can travel within the circulation leading to the formation of emboli, which can obstruct vessels and produce tissue necrosis far away from the thrombus formation site. In the subsequent session a brief description of different vascular thrombosis and their current treatments will be presented.

1.1.1   Arterial Thrombosis

Arterial thrombosis is the formation of thrombi in arteries and is characterized by high concentrations of platelets aggregates. Usually, it occurs in the presence of high blood flow and many studies have shown that arterial thrombi are caused by damage in the atherosclerotic plaques, which are formed by release of procoagulant factors [3].

1.1.1.1   Coronary circulation and myocardial infarction

The coronary circulation, which is responsible for the supply of blood to the heart muscle, originates from the root of the aorta and it is structured in right and left coronary arteries. The right coronary artery is responsible for sending the blood to the right atrium and right ventricle. In analogy, the left coronary artery supplies the left atrium and left ventricle with blood. Furthermore, the left coronary artery is divided in left circumflex artery and left anterior descending artery [2] (Figure 1.1). The latter one is the most affected by emboli during coronary circulation disease [4]. The coronary arteries (approximately 4mm in diameter) are subdivided in smaller vessels until they reach the size of a capillary network (40μm), which penetrate in the tissue and allow the exchange of oxygenated blood to the heart cells.

The amount of cardiac output that goes into the coronary circulation is 5% and the provided O_2 is used for oxidative processes and production of energy. Reduced oxygen supply for long periods, due to obstructions, can lead to myocardial necrosis (myocardial infarction). However, collateral blood vessels (originating from existing vessels through remodeling processes) may moderate the reduced flow, diminishing tissue damage. Nevertheless, collateral flow may be not sufficient and the patient may feel pain in the chest (angina pectoris).

In order to treat patients with coronary artery disease, different fibrinolytic agents can be administered. Some examples of fibrinolytic agents are: tissue plasminogen activator (t-PA or recombinant rt-PA when manufactured using recombinant biotechnology), urokinase-type plasminogen activator (u-PA) and streptokinase.

When the drug delivery is not sufficient or cannot lead to a total reperfusion, mechanical procedure for revascularization may be necessary. Examples of this technique are coronary artery bypass grafting and transluminal angioplasty. The latter procedure requires the use of an inflatable balloon, inserted using a balloon-tipped catheter. These techniques are used mostly for the treatment of plaques, which are lesions within arteries. Plaques are characterized by accumulation of lipids

A VIEW FROM FRONT

Pulmonary artery
Aorta
Left coronary artery
Superior vena cava
SA nodal artery
Right coronary artery
Right atrium
Right marginal (acute) artery
Inferior vena cava
Right ventricle
Great cardiac vein

B VIEW FROM DIAPHRAGM

Superior vena cava
Aorta
Right pulmonary veins
Left atrium
Inferior vena cava
Coronary sinus
Right atrium
Left circumflex branch
AV node artery
Left anterior descending branch
Right main coronary artery
Left diagonal branch
Left ventricle
Right marginal (acute) branch
Right ventricle
Left marginal (obtuse) branch
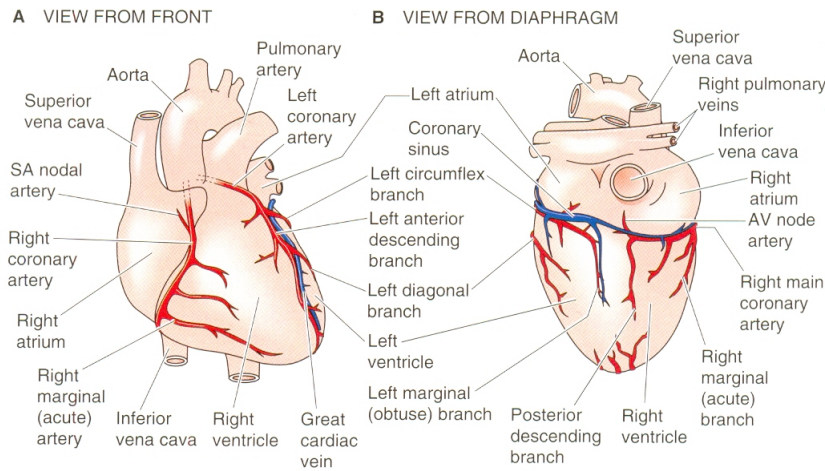Posterior descending branch

Figure 1.1: Coronary arteries [2]

and fibrous elements and the activation process is started by alteration of the endothelium and activation of the inflammatory process [5, 6]. Plaques can become complex, growing and obstructing the blood flow. In this situation, mechanical procedures are used in order to flatten the plaque and restore blood flow [2]. However, clinical complications are most likely to occur when a plaque is disrupted and leads to acute occlusion due to thrombus formation.

Both pharmacological and mechanical treatments have low efficiency and can lead to dangerous side effects such as hemmorrage and blood vessels damage. Moreover, in the treatment of myocardial infarction, these two methods often do not achieve reperfusion. This is because the principle aim of these techniques is to restore the patency of the vessel and not the reperfusion. With the term "patency" is meant the quantity of vessels that are unoccluded after the treatment (including also the vessels that were not completely obstructed before treatment), while the term "reperfusion" indicates the restored tissue perfusion within the myocardium [7]. A lack in reperfusion leads to disorganization of capillary structures, swelling and edema. All these processes converge to the phenomenon of no-reflow that leads to the impossibility of late recovery because pharmacological agents are not able to reach the area [8]. Moreover, small particles may be produced within the infarcted area and obstruct small arteries and arterioles downstream.

Different studies have demonstrated that sonothrombolysis is a therapeutic technique that can achieve clot lysis and myocardial reperfusion and that ultrasound combined with MBs can treat the infarcted zone even if the upstream artery remains occluded [9]. The mechanisms that are thought to be responsible for the reperfusion with this technique are the triggering of collateral perfusion and lysis of microemboli [9].

1.1.1.2 Ischemic Stroke

Another consequence of arterial thrombosis is ischemic stroke, during which the blood supply to the brain is interrupted and brain cells do not receive the substances for maintaining their function. The

interruption of blood flow arises from clots stopping in cerebral vessels after they have travelled long distances [10]. When the blood supply to the brain is compromised, the ischemic cascade occurs, which is a sequence of events at cellular level that governs the progressive cells death. In a patient affected by an ischemic stroke it is possible to distinguish a zone of death cells surrounded by hypoperfused tissue called penumbra zone [11](Figure 1.2).
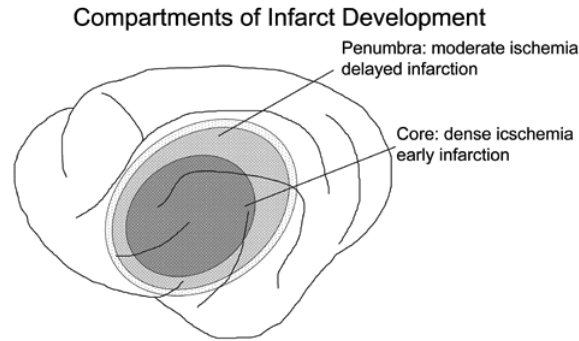
Figure 1.2: **Ischemic area**. Diagram of ischemic zones [12]

As for the myocardial infarction, thrombolytic agents can be used for the treatment of ischemic stroke and the most used is rt-PA. However, several studies have shown that this technique leads to intracerebral hemorrhage and bleeding [13]. Moreover, even without side effects, the thrombolytic treatment may be inefficient and clot may still obstruct the brain vessels [14].

In addition, it is important to underline that not all the patient with ischemic stroke can be treated with thrombolytic agents because of different factors such as comorbidity and protocol exclusions. Indeed, this type of drug must be administrated within three hours from symptom manifestation and patients that have taken heparin in the previous 48 hours should not be treated with thrombolytic therapy. Other reasons for exclusion are: another stroke in the previous three months, surgery within the preceding 14 days, systolic blood pressure greater than 185 mm Hg or diastolic blood pressure greater than 110 mm Hg [15].

1.1.1.3 Systemic embolism

Thrombus can be disrupted forming emboli, which are small particles that move within the blood stream and can reach limbs occluding an organ extremity. This can be seen through imaging, surgery or autopsy and it can occur in the absence of traumas or atherosclerosis.

1.1.2 Venous thrombosis

Venous thrombosis is the formation of thrombi in the veins and is characterized by high concentrations of fibrin. The most common type of venous thrombosis is deep venous thrombosis. Other types of venous thrombosis include portal vein, renal vein and cerebral venous sinus thrombosis.

1.1.2.1 Deep Venous thrombosis

Deep vein thrombosis occurs within the deep veins and is the most frequent form of venous thrombosis. Deep venous thrombosis usually originates from the veins of the calf and can be classified in proximal and distal. Proximal deep venous thrombosis involves thigh veins, while the distal one involves calf veins. Complication of this type of thrombosis can lead to venous valvular insufficiency, venous chronic obstruction and pulmonary embolism, which is the most common outcome [16]. Pulmonary embolism is the obstruction of main artery in the lung. More than 90% of acute pulmonary embolisms are caused by emboli produced in the proximal deep venous thrombosis [17].

The treatment of this type of thrombosis implies the use of heparin and warfarin derivatives, which are anticoagulant drugs. This therapy prevents the propagation of the thrombus without lysis of the existing one [18].

Another possible treatment is through catheter-based deliver of thrombolytic agents reducing the post-thrombotic syndrome that may occur after a deep venous thrombosis manifestation and may lead to chronic venous ulcerations, swelling and pain, which are caused mostly by damages to venous valves [17].

1.2   Ultrasound principles

In this section a brief introduction to the ultrasound principles will be presented. The aim is not to give an exhaustive description, rather an overview in order to better understand the subsequent chapters and the project aim.

1.2.1   Ultrasound Physiscs

Ultrasound waves are sinusoidal mechanical perturbations with a frequency (number of vibrations per unit of time) greater than 20kHz. In order to propagate, the ultrasound waves need a medium. The molecules of the medium should not be too widely spaced like in gases, where the propagation of the waves is strongly reduced. For this reason, in some bodies regions, such as lungs, the ultrasound has poor efficacy. The propagation of ultrasounds occurs through consecutive phases of compression and rarefaction along the propagation direction (Figure 1.3). Therefore, displacement, velocity, pressure and acceleration of ultrasound can be described through a sinusoidal function, characterized by a specific frequency and wavelength.
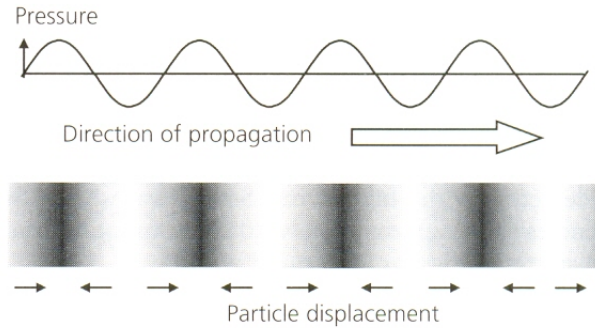
Figure 1.3: Propagation of ultrasounds [19]

The elastic properties of the ultrasound waves in a medium are described by the following Poisson's equation:

$$P = -B\frac{\partial U}{\partial x} \tag{1.1}$$

where P is the acoustic pressure, B is the elastic modulus and U is the displacement of the particle around the equilibrium position. Combining the equation 1.1 with the equation of motion for a particle, it is possible to find the relation between the elastic modulus, density and the velocity of the wave, which is:

$$c = \sqrt{\frac{B}{\rho}} \tag{1.2}$$

with c being the velocity and ρ the density of the tissue. Therefore, low density and high stiffness lead to high speed of sound.

The velocity of sound is assumed constant in all the soft tissues and equal to 1540m/s.

The energy transported in an ultrasound wave is usually characterized by an istantaneous acoustic intensity *i(t)* defined as

$$i(t) = \frac{p(t)^2}{\rho c} \tag{1.3}$$

with *p(t)* being the instantaneous pressure. From equation (1.3) it is possible to see that the intensity increases with the pressure amplitude of the wave.

The ultrasound waves for diagnostic purposes (2-15MHz) are delivered through transducers which may be linear or phased-array. The former are larger in size with rectangular field of view and, due to the high frequencies used, they are suitable for near surface applications. The phased-array transducers, instead, are smaller, with a sector field and they are more suitable for deeper targets.

The delivery of ultrasound waves can be continuous or pulsed, where the latter is characterized by sequences of pulses of specified duration spaced by an idle time. In this condition the important parameters to consider are the pulse length and duty cycle. The latter is the percentage of time that the ultrasound pulses are delivered as a fraction of the total time under consideration. Duty cycle is correlated with the pulse repetition frequency, where the latter is the number of pulses per unit of time. High values of pulse repetition frequency lead to high values of duty cycle.

1.2.2 Ultrasound interaction with tissue

When ultrasound wave propagates, it interacts with tissue through mainly three factors: reflection, refraction and attenuation.

1.2.2.1 Reflection

Reflection is caused by an ultrasound wave travelling from a medium with impedance Z_1 to another medium with different impedance, Z_2. Across the interface between the two media, part of the acoustic energy is reflected back in the first medium. This type of interaction is very important in image formation because the time point at which the returning signal occurs, gives information about the position of the target within the body.

1.2.2.2 Refraction

When the direction of an ultrasound wave is not perpendicular to the target surface and there is a change in the ultrasound speed between two different materials, part of the wave that is not reflected passes through the boundary and changes the direction of the propagation leading to the refraction effect. When there is an increase in the velocity of the ultrasound wave from one medium to the other the angle to the normal of the boundary surface also increases as described by the Snell´s law:

$$\frac{sin\vartheta_i}{sin\vartheta_t} = \frac{c_1}{c_2} \tag{1.4}$$

where ϑ_i is the angle between the incident wave and the normal to the boundary, ϑ_t is the angle between the transmitted wave and normal to the boundary, c_1 is the velocity in the first medium and c_2 is the velocity in the second medium.

1.2.2.3   Attenuation

Attenuation is responsible for reducing the intensity of the ultrasound signal traveling through a medium and is mainly caused by scattering and absortion.

For small targets the reflection laws are not valid due to the fact that, when the wave impacts on such targets, it is scattered over a large range of angles. Therefore scattering phenomena lead to changes in the sound direction of propagation.

Absorption is the conversion of the sound energy to other forms of energy. It is mainly caused by molecular relaxation, which consists in the transformation of ultrasound energy in cellular energy (e.g. heat). Each molecule has a specific frequency of absorption and in biological tissue the absorption increases with frequency. The total attenuation coefficient α, which includes the previously described factors, is given by

$$\alpha := \frac{\frac{\triangle I}{I}}{\triangle x} \tag{1.5}$$

where x is the displacement and I is the intensity and it is given by:

$$I = I_0 e^{-\alpha x} \tag{1.6}$$

with I_0 being the initial intensity.

The attenuation coefficient of most tissues when expressed in $dBcm^{-1}$ increases approximately linearly with frequency. Therefore, when imaging deep organs a low frequency must be used, whereas high frequency can only be used to image superficial targets.

1.2.3   Non linear propagation

When ultrasound waves are propagating in tissue, they generate echoes at foundamental frequency f_0 and at harmonic frequency (mostly at $2f_0$). The echoes at harmonic frequency vary with the depth (Figure 1.4) and they are caused by non-linear propagation of ultrasound in the tissue, multiple reflections and tissue compressibility. To understand the harmonic generation it is important to consider the ultrasound pressure wave. Indeed, the velocity of the ultrasound wave is higher in the phase of compression, when the pressure is positive. This leads to a distortion (Figure 1.5) of the transmitted wave that generates echoes with frequencies at multiple values of the transmitted one.
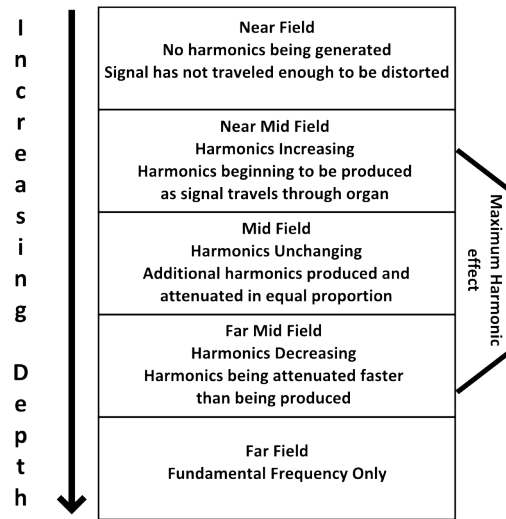
Figure 1.4: Harmonic production with increasing depth
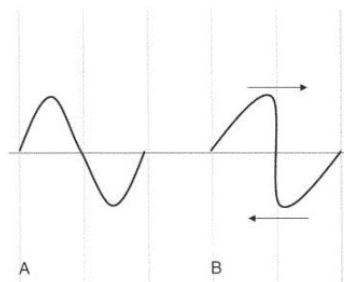
Figure 1.5: Distortion of ultrasound wave. A: wave not distorted B: distorted wave

In imaging mode, harmonics can allow deep penetration and higher resolution because the transmission is at the frequency f_0 and reception at the harmonic $2f_0$. In addition, the harmonics generation occurs only if the value of the acoustic pressure is higher than a threshold, leading to the reduction of some artifacts due to, for example, lateral lobes. Lateral lobes are caused by ultrasound waves that propagate in other directions than the frontal one. However, the intensity of such waves is too low to trigger the generation of the harmonics and therefore the signal is not detected by the transducer.

1.2.4   Contrast agents

Non-linear propagation and harmonic generation may be enhanced with specific contrast agents composed by small (1-6μm) encapsulated microbubbles (MBs), which are injected intravenously and combined with specific contrast sequences in order to enhance the ultrasound effect. Several types of MBs have been developed and in Table 1.1 some examples are listed.

Table 1.1: Characteristic of some contrast agents [19, 20].

AGENT	*DISTRIBUTION AREA*	*TYPE OF AGENT*	*SHELL*	*GAS*	*BUBBLE SIZE*
LEVOVIST	*Europe, Japan, Canada*	*Lipid stabilised bubble*	*Palmitic Acid*	*Air*	*3-5μm*
SONOVIST	*development stage*	*Solid microspheres*	*Cyano-acrylate*	*Air*	*Mean 2μm*
DEFINITY	*USA, Europe, Canada*	*Encapsulated bubble*	*Lipid*	*Perfluoropropane*	*Mean 2μm*
OPTISON	*USA, Europe, Canada*	*Encapsulated microsphere*	*Albumin*	*Octafluoropropane*	*Mean 3.7μm*
SONOVUE	*Europe, China*	*Stabilised bubble*	*Phospholipids*	SF_6	*2-3μm (90%<8μm)*
SONAZOID	*not commercialized*	*Information not available*	*Surfactant membrane*	*Fluorocarbon*	*Median 3.2μm*
ALBUNEX	*not commercialized*	*Encapsulated bubbled*	*Albumin*	*Air*	*Mean 4μm (Range 2-10μm)*

Early agents contained air and the coating was made of albumin (Albunex). Later, agents were developed with a fluorinated gas core (Optison) or perfluoropropane gas and protein shell (Definity).

The first air-filled MBs tend to collapse in saline and their weak shell decrease their ability to cross the circulation of lungs. The latest MBs, which are lipid coated, are more stable in saline and their size is consistent with the dimensions of the microcirculation.

The manufacturing process of MBs is done mostly by mechanical agitation and, after their injection they circulate similarly to red blood cells for an interval of time that is of the order of minutes. When the MBs are reached by ultrasound waves with specific frequencies, their vibration is enhanced, thus increasing the echo signal strength. MBs are strong scatterers and the maximal enhancement of the echo signal occurs when the applied frequency is close to the resonance frequency of the MBs. This resonance frequency has values close to the ones used in diagnostic ultrasound. For example, the SonoVue MBs have a mean resonance frequency that ranges between 1-4MHz [21], which is a typical range at which most of the clinical transducers work. Therefore, the use of these MBs does not require changes of the ultrasound machines or transducers.

The pressure value of the incident waves is another important factor to consider when analyzing the MBs interaction with ultrasound. For low peak rarefactional pressure (around 100kPa) the oscillations of MBs corresponds to the rarefaction and compression of the ultrasound wave with a frequency close to the incident ultrasound frequency. For higher peak rarefactional pressure the oscillation of the MBs has a non-linear relationship to the driving pressure. The MBs expansion in

this case is no more sinusoidal and it is followed by a faster collapse. Rising even more, the peak rarefactional pressure leads to the destruction of the MBs, with gas diffusing in the surrounding space. When MBs collapse fluid jets are produced that can result in 16μm wide pit in the surface of cells [22].

The first imaging techniques that included dedicated contrast sequences for enhancing the signal from the MBs and suppressing tissue signal, were based on the detection of the harmonics returned from each transmitted pulse. Each echo was received and filtered in order to keep only the harmonics. Later techniques involved the use of low-amplitude pulse trains which consists of pulses with alternated phase, amplitude or both phase and amplitude. Summation of the returned echoes cancels the linear component coming from the tissues while keeping MBs echoes. Harmonic imaging with MBs has many applications in cardiology for the assessment of wall motion and perfusion defects and great interest is raising in MBs applications for targeted ultrasound imaging. This technique involves the use of contrast agents having a ligand on the surface that binds to a function-specific molecule. This leads to persistent enhancement during imaging [23], enabling to distinguish better the target area (where the contrast agent binds) from normal tissue.

Beside transmission frequency and peak rarefactional pressure, other important parameters are pulse duration, time between destructive pulses, MBs concentration and size distribution. Pulse has to be short during imaging for better axial resolution, which is the smallest distance between two targets along the beam axis that enables to obtain two separable images. Therefore, short pulses will lead to short echoes from the MBs improving the image resolution.

In case of destructive pulse delivery, the time between two pulses should be enough for bubbles replenishment, allowing new bubbles to reach the area of interest and to be disrupted emitting non-linear echoes.

In perfusion imaging it was seen that there is a strong dependence between signal intensity and the size distribution and concentration of the contrast agent administered [24]. Increasing the MBs concentration led to better imaging of the interested area until a saturation point is reached. For higher concentration than the saturation value no improvement in image quality was seen. Also an increase in MBs size led to better image quality through contrast enhancement.

In this project, two types of MBs were compared:

- **SonoVue MBs.** The SonoVue MBs (Bracco, Milan, Italy) contain sulphur hexafluoride, which is a very stable molecule, without any interaction with other body molecules. The shell consists of a phospholipidic membrane that is highly flexible, allowing the MBs to change the size and shape. The mean size of these MBs is 2.5μm and the resonance frequency ranges between 1 and 4MHz [21].

- **3MiCRON MBs.** The 3MiCRON MBs are air-filled MBs characterized by a polymeric shell, which is much thicker and stiffer than the SonoVue one. This leads to the need of high pressure values for the bubble oscillation and shell rupture.The mean size of these MBs is 3μm and the resonance frequency is around 12MHz [25].

1.3   Sonothrombolysis

In the last years sonothrombolysis has become a very attractive alternative in the treatment of thrombus and several studies were performed and are still ongoing in order to gain more knowledge about the process of interaction between ultrasound waves and thrombi.

Investigations of the use of ultrasound in the lysis of thrombi located in different areas were done in particular within brain, heart and peripheral arteries. The combination of ultrasund and fibrinolytic drugs was also studied and the results showed that sonothrombolysis in these conditions may be enhanced (40% increase in clot mass loss), leading to the reduction of the concentration of rt-PA needed. In particular, the use of ultrasound and rt-PA enhance thrombolysis in vitro and in vivo [26, 27, 28], accelerating the recanalization through the increase of the transport and uptake of rt-PA into the thrombus. It is important to underline that the enhancement of the lysis is not due to the specific fibrinolytic agent used. Indeed, in several studies, other anticoagulant drugs, such as urokinase [29], were used and the results are comparable with the ones obtained using rt-PA. All the fibrinolytic agents used were able to reduce the clot size considerably when delivered together with ultrasound.

Moreover, injection of MBs in combination with ultrasound and rt-PA may further improve the clot lysis and increase the thrombolytic therapy [26]. However, research about this combination is still on going and the MBs potentiality in improving clot lysis is still under study.

Furthermore, ultrasound, either alone or in conjunction with fibrinolytic agents and MBs, is a very attractive approach in treatment of thrombosis because of its ease of use, low costs and bedside applicability [30].

1.3.1   Mechanisms of action and MBs influence

The mechanisms of action of ultrasound in the lysis of thrombi are still poorly understood and more studies are needed in order to understand the interaction between the ultrasound waves and thrombi. Some possible mechanisms of action of ultrasound waves are cavitation, vibration of solid structure, acoustic streaming and heating. However, the latter is too mild to be considered as a possible cause of the thrombolytic effect [30].

The most studied mechanism is cavitation that is the formation of bubbles within ultrasound field that undergo oscillation and disruption. More in detail, when a fluid is exposed to an acoustic pressure field, gas- and vapor-filled bubbles are produced. This mechanism, when occurs within a thrombus, generates fragments of fibrin [31] characterized by irregularity and porosity [32].

There are two types of cavitation: stable cavitation and inertial cavitation. Stable cavitation originates from bubbles oscillating with a non-linear behavior and leading to the emission at frequencies that are multiple or submultiple of the central frequency. Stable cavitation is the main cause of microstreaming production, which is the formation of small streams within the fluid. The microstreaming patterns depend on the particular oscillation mode of the bubble. Microstreaming generates shear stress in the vicinity of a bubble and this may cause the reduction of the thrombi when the bubbles are in contact with their surface [33].

On the other side, inertial cavitation is generated by bubbles that collapse emitting broadband noise producing microjetting [26]. Microjetting causes the pitting on solid-surfaces, and, in the presence of thrombi, the erosion of their outmost part. Cavitation is strictly correlated with clot mass loss and this mass loss was seen mainly in the presence of stable cavitation than in the presence of inertial cavitation or both stable and inertial, as stated in [32]. In this article a distinction

between stable cavitation inside and at the surface of the clot is presented. Inside the clot, the bubbles undergoing oscillation close to the fibrin mesh expose new binding sites for plasmin that were previously not exposed. At the surface instead, bubbles are comparable to micropumps which make the removal of fibrin debris easier. Further studies are necessary in order to better understand which of the two types of cavitation mechanisms plays the major role in thrombolysis.

The ability of ultrasound to induce cavitation is described through the Mechanical Index (MI), which is defined as follows:

$$MI = \frac{P_n}{\sqrt{f}} \tag{1.7}$$

Where P_n is the peak negative pressure (MPa) and f is the ultrasound transmission frequency (MHz). The MI is displayed and can be changed in all ultrasound devices. The upper limit imposed by the FDA is 1.9 and higher values are considered dangerous [19].

Cavitation is triggered above a certain threshold and different studies have found different cavitation threshold values. For example, inertial cavitation in human blood was found to be 2.95MPa at 2.5Mhz and 6.2MPa at 4.3Mhz [34]. Therefore, cavitation thresholds depend on frequency of the pulse. Moreover, it was shown that continuous wave ultrasound led to lower cavitation thresholds in the presence of rt-PA within the clot and the same trends were present for pulsed ultrasound exposure [32]. The presence of contrast agents also lowered the cavitation threshold, thus less ultrasound energy was needed (one-third) in order to produce the same cavitation effect as when MBs were not used.

The cavitation properties of MBs depend on their size, concentration and shell elasticity. It is not easy to adjust MB's properties in order to enhance thrombolysis because most of the studies are performed using the commercially available MBs. Therefore, it is very common to adjust the ultrasound parameters according to the MBs used. Although the properties of the MBs are already defined, it is important to understand how these characteristics may affect the sonothrombolysis. In [35], the influence of shell elasticity and MB's size on the efficacy of sonothrombolysis was studied. The results showed that increasing the Young's modulus increases the sonothrombolysis. However, too high values of the Young's modulus led to more stiff MBs shell, increasing the resonance frequency. Concerning the size, it was shown that smaller MBs lead to the need of higher concentrations in order to achieve optimal clot lysis. For example, it was shown that the optimal concentration with 1µm MBs was $5.4\text{x}10^8$ MB/mL, fivefold greater than the optimal concentration of $1.1\text{x}10^8$ MB/mL for 3µm MBs [35].

1.3.2 Ultrasound thrombolysis techniques

Since the first application of ultrasound as thrombolytic technique several studies have been performed and different ultrasonic approaches have been considered. It is possible to distinguish three main techniques:

- Catheter based ultrasound delivery

- Pulsed High Intensity Focused Ultrasound (HIFU)

- Surface/transcutaneous ultrasound delivery

All these approaches were studied in vitro and in vivo, both in animal studies and in human clinical trials, with or without the use of pharmacological agents and/or MBs. In the following section a brief description of each of these techniques and the relative studies results will be presented.

1.3.2.1   Catheter-delivered ultrasound thrombolysis

There are two subtypes of catheter based ultrasound delivery for thrombolysis: the catheter-delivered transducer-tipped ultrasound and the catheter-delivered external transducer ultrasound.

The first one involves the transducer positioned directly on the tip of the catheter in order to treat the thrombus with direct contact, without involving the surrounding tissue. Catheter-delivered transducer-tipped ultrasound system is in clinical use for the treatment of deep venous thrombosis and acute stroke [16] with frequencies ranging from 100kHz to 1.5MHz and intensities up to $2W/cm^2$ [36]. The advantage of this method is in the reduced risk of tissue damage around the target. This technique was also combined with plasminogen activators (rt-PA or urokinase) in order to accelerate drug delivery within the clot and enhance thrombolysis, as shown by Tachibana in several studies [37, 38]. Moreover, this technique was analyzed successfully with the combination of both fibrinolytic and contrast agents. The presence of contrast agent increased the rate of thrombolysis by 33-51% [29].

In the second type of catheter-delivered ultrasound the transducers is positioned outside of the body and a metal wire transmits the ultrasound signal to the tip of the catheter inserted in the body. The length of the wire affects the time necessary for clot disruption. This technique was shown to be effective in the clot dissolution of acute myocardial infarction. With this method, the frequency used is low, in the range from 19kHz to 45kHz, whereas the intensities are much higher if compared to the transducer-tipped method, having values up to $24W/cm^2$ [36].The catheter-delivered external transducer method produce local cavitation, microstreaming and other mechanical effects, but it does not enhance drug delivery and for this reason this technique was not applied in conjunction with fibrinolytic agents.

Both the types of catheter-delivered methods have the disadvantage of the invasive delivery of ultrasound, with the possibility of the wire and catheter flexion, torsion and the risk of perforation of the tissue.

1.3.2.2   Transcutaneous-delivered HIFU external ultrasound

High Intensity Focused Ultrasound is a technique that, compared to the previous one, is not invasive and applies the ultrasound externally through a focused transducer, generally characterized by a spherically and curved shape.

The potential of HIFU as a therapeutic method was first studied by Lynn et al in [39] and its applicability in thrombus destruction was analyzed in [40], where a 500kHz pulsed ultrasound was applied through a transducer with 3 piezo-electric elements and a spherical ring shape. Moreover, the transducer had an acoustic lens with a focus point set at 45mm. The pulsed wave parameters was found to be very important and critical for the success of the treatment. In this study no pharmacological agents were used and no damage to the surrounding tissue was detected. Only applying higher intensities ($\geq 45W/cm^2$) for longer times (>5 min) led to the damage of the tissue. More recently, in another study [41] the potential of HIFU was combined with t-PA, resulting in the enhancement of thrombolysis. The therapeutic transducer used in this study was a custom-built concave and spherical element with a diameter of 5cm.

In both [40] and [41] pulsed-wave has been found more effective than continuous wave and it was shown that the parameters of ultrasound signals affect the results. In particular, pulse length and duty cycle play the major role for the achievement of thrombolysis efficiency. Shorter duty-cycle can reduce tissue damage, whereas longer duty-cycle leads to increased thrombolysis.

Despite the different studies that have been performed, more work is needed in order to understand the feasibility and safety of this technique.

1.3.2.3   Surface ultrasound delivery

Another non-invasive approach for the delivery of ultrasounds in order to achieve lysis of the clots consists in the use of diagnostic transducer (phased or curvilinear array), which, compared to the HIFU technique, leads to less possibilities of tissue damage through heating.

In [42] it was shown that diagnostic ultrasound, combined with rt-PA, can enhance enzymatic thrombolysis, reducing the recanalization time of vessels up to a depth of 50mm. The mean frequency applied in this study was 1.8MHz. In several other studies, surface ultrasound delivery was combined with MBs in order to enhance the thrombolytic efficiency. In [43] a diagnostic ultrasound system with 4C1 transducer (ACUSON Sequoia 512, Siemens Medical Solutions, California) was used in combination with MRX-801 MBs in order to assess if it was possible to achieve thrombus dissolution with high MI (1.9) impulses. The results, during a continuous infusion of MBs, lead to a 71% of success at 30 min of treatment in a canine model. In another study [44], high mechanical index impulses from a diagnostic transducer combined with platelet-targeted intravenous MBs have shown improvement in recanalization of epicardial and microvascular areas in pigs. The transducer used was the 4V1c (Siemens Ultrasound Solution), operating at 1.5MHz with high MI (1.9). Regarding recent studies, it is important to mention the in-vivo investigation described in [45]. In this study, the safety, feasibility and therapeutic effectiveness of the MB enhanced thrombolysis was analyzed using high MI (1.1-1.7) US (iE33, S5-1 transducer, Philips) applied intermittently on pigs. In this study it was shown that inertial cavitation was the predominant mechanism in the clot lysis, enhanced with the use of a contrast agent similar to the commercially available Definity. Moreover, high MI ($>$1.0) was found to further increase the reduction of chronic venous thrombi in vivo.

1.3.3   Safety and efficiency aspects of Sonothrombolysis

Efficiency and safety of sonothrombolysis were deeply analyzed and studies are still ongoing. It is not easy to assess when to consider the use of ultrasound in combination with tPA or contrast agents completely riskless. Moreover, there is no accordance over the optimal ultrasound parameters in order to perform efficient and safe treatments. A great limitation for achieving this is the small number of patients in the different safety studies and the heterogeneities in the study protocols. However, some important results were obtained when analyzing the ultrasound exposure in patients with different types of thrombosis as it can be seen in the following paragraphs.

Cardiovascular thrombosis

Cardiovascular applications of sonothrombolysis resulted in higher success rate of complete lysis and reduced time needed for coronary artery flow restoration than administration of thrombolytic agents alone. However, intensity exposure higher than $45W/cm^2$, for long periods ($>$ 5 min), led to arterial wall damage [46]. Moreover, ultrasound approaches such as catheter-based have invasive nature and still the design of these catheters is the major challenge due to difficulties in transmitting the ultrasound wave unchanged along the wire. Other approaches, such as HIFU, with the use of low frequencies, demonstrated the ability of ultrasound deep penetration, minimal tissue attenuation and mild temperature elevations. These elements are particularly important when treating

areas in the body which are difficult to reach, such as the hearth, where it is necessary to focus the beam transthoracically considering the high attenuation of the ribs [46]. Concerning contrast agent application, the combination of transcutaneous ultrasound with MBs in different studies demonstrated no evidence of skin damage, lung parenchyma or coronary vasculature. Moreover, after sonothrombolysis, the clot debris was less than 10μm in size.

Ischemic Stroke

Considering ischemic stroke treated with ultrasound, the safety was analyzed controlling the presence of hemorrhage. In [47], it was found that sonothrombolysis in combination with intravenous thrombolytic agent led to higher likelihood of intracerebral hemorrhage than sonothrombolysis alone. In addition, low frequency and high intensity ultrasound has to be preferred in the treatment of ischemic strokes, as stated in [48]. Another parameter considered for the safety and efficacy of sonothrombolysis was the possibility of re-occlusion in the first hours after the treatment. Using diagnostic 2MHz ultrasound probes ensures that this event will not occur.

On the opposite, other studies found that high frequency ultrasound waves were safer than low frequency waves. Indeed, in [49] high frequency appeared to be safe with small occurrence of intracerebral hemorrhage and with greater likelihood of complete recanalization at 3 months. In that study, the author hypothesized that low frequency ultrasound was able to damage small vessels, generating vasodilatation and opening of the blood-brain barrier, which is the separation (made of conjunctions around capillaries) between blood circulation and extracellular fluids in the central nervous system.

Another important result about safety of sonothrombolysis was achieved in [50], where the clinical tests were performed using perflutren-lipid MBs. This study demonstrated no increased risk of symptomatic intracranial hemorrhage (presence of intracranial blood diagnosed by repeated computed tomography or magnetic resonance imaging scanning) in patients treated with intravenous tPA with MBs and 2MHz ultrasound.

Deep venous thrombosis

Thrombolysis using ultrasound was also found to be effective and safe in the treatment of deep venous thrombosis, reducing the total infusion time of fibrinolytic agents and lowering the likelihood of bad outcomes such as bleeding. The reduction of the infusion time of fibrinolytic agents is important because long infusions lead to major bleeding complication. Catheter-based ultrasound can achieve venous recanalization, preservation of valvular function, prevention of pulmonary embolism and improvement of quality of life [51].

Pulmonary embolism

In the treatment of pulmonary embolism, the safety of sonothrombolysis was studied employing high-frequency, low-power ultrasound delivered through catheter-based technique in conjunction with fibrinolytic agent (rt-PA) [52]. The results showed complete resolution of the clot burden after 20 hours. As in the treatment of deep venous thrombosis, ultrasound can reduce the dose of lytic agents necessary in the treatment of pulmonary embolism. This aspect is very important when considering patients with high risks of bleeding risk complications.

Chapter 2

Design, validation and optimization of the in-vitro set-up

As stated in the introduction, the first goal of the present work was to develop an in-vitro set-up for sonothrombosis and to evaluate its perfomance.

In this chapter the methods and results for the design and validation of the set-up are presented. Optimization was required because some problems arose when testing the set-up. The results obtained led to the definition of a final set-up which, afterwards, was used for the evaluation of the thrombolytic effect using ultrasound in conjunction with MBs.

The design of the set-up included decision about the materials for reproducing the vessel mimicking phantom, mechanism of flow delivery, methods for detecting sonothrombolysis efficiency and methods for clot production.

2.1 Methods

The starting point for the set-up realization was the information found in [9] where the authors modeled the microembolization in order to assess the role of sonothrombolysis in treating the no-reflow phenomena, which was described in section 1.1.1.1. They used a 40μm pore mesh (cell strainer; Falcon Franklin Lakes, NJ, USA) mounted across the lumen of an artificial blood vessel made of rubber and with a diameter of 4mm (Figure 2.1). In this experiment the mesh was occluded with multiple microemboli and the ultrasound treatment was delivered using a single-element transducer at 1MHz in conjunction with perfluorobutane gas lipid-encapsulated MBs.

Using this set-up as a reference, a new one was developed. The obtained set-up presented two main problems: air bubble formation within the system and leakages. In order to solve these problems, possible solutions were evaluated and the final set-up was defined form the results of these tests and optimizations.

The overall initial set-up is presented in Figure 2.2 and the main components were a vessel mimicking phantom, a peristaltic pump, a pressure measuring device and a programmable ultrasound system (the Verasonics) connected to a computer.

The vessel-mimicking material of the phantom was obtained mixing 400ml of deionized water with 15% of poly(vinyl alcohol) (PVA) and 3% of graphite as described in [53]. The PVA ensured
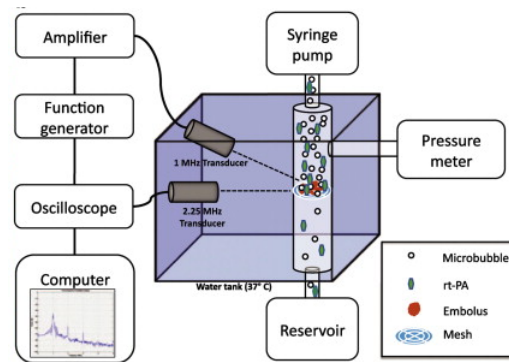
Figure 2.1: The set-up proposed in [7] consisting in a rubber flow phantom containing a 40μm pore mesh
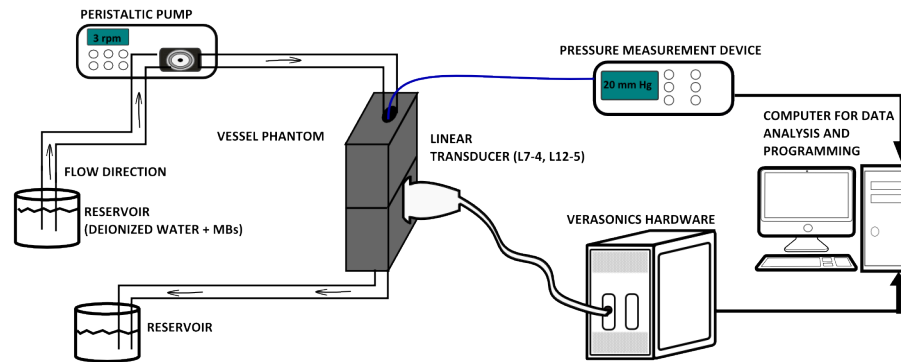
Figure 2.2: The initial in-vitro set-up

mechanical characteristic similar to the soft tissues and the graphite was added in order to reproduce scattering properties of tissues and tissue-like speckle texture. The mixture of PVA, graphite and deionized water was stirred and heated up to 90°C and then poured in two moulds (made of delrin polyacetal and polymethyl methacrylate), designed in order to obtain the shape of a rectangular prism (Figure 2.3) with a vessel lumen of 4mm in size (the same diameter as found in [9]). This lumen size reproduce the real size of the coronary arteries, in particular the lumen of the left anterior descending coronary artery. When the moulds were completely filled with the mixture they were stored for 12 hours at -20°C. After that, the phantom was kept at room temperature for another 12 hours. This freeze-thaw cycle was repeated three times in order to obtain optimal vessel-mimicking material, with speed of sound, attenuation and backscattered characteristics similar to that of soft tissues. It was noticed that the phantom shrank in size after the three freeze-thaw cycles, keeping, however, the predetermined lumen diameter.

The thickness of the phantom was chosen considering the ultrasound frequencies applied (4.09MHz and 11.25MHz) and the attenuation coefficient (assumed equal to the soft tissues one, approximately $0.5\text{dBcm}^{-1}\text{MHz}^{-1}$, as found in previous tests done in the laboratories of KTH). When transmitting
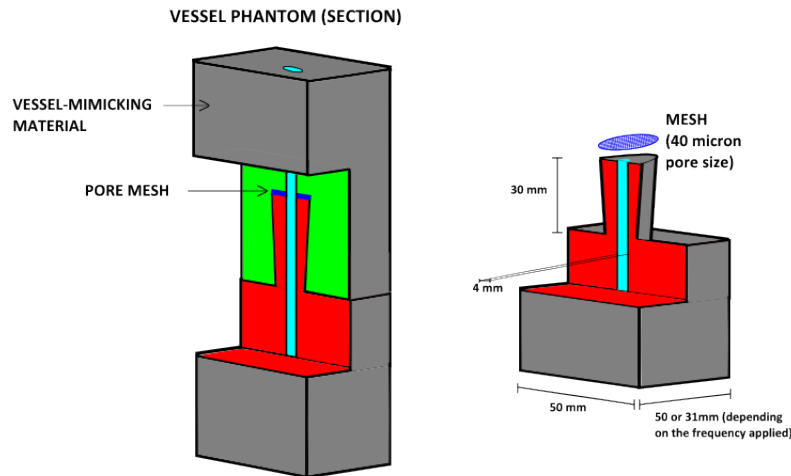
Figure 2.3: Vessel phantom

pulses at 4.09MHz the transducer was placed 25mm from the vessel lumen while the highest frequency (11.25MHz) led to the need of a smaller distance, which was equal to 10mm approximately. In this way, the ultrasound voltage in the clot area was reduced to the same value (43% of the transmitted voltage) for both the thicknesses. In order to reproduce the microcirculation and to keep the clots in the same position during the delivery of the flow, a 40μm pore mesh (cell strainer; Falcon Franklin Lakes, NJ, USA, the same used in [9]) was placed across the vessel lumen. For this reason the vessel phantom was composed of two halves, which were connected after positioning the mesh between them (Figure 2.3). This design enabled to change the mesh and to extract the remained clot, which is important when repeated measurements are being performed.

In order to reproduce the physiological flow, a saline solution was delivered together with the MBs, using a peristaltic pump (Watson Marlow, Falmouth, United Kingdom). The peristaltic pump was set to a flow velocity of approximately 8.75mm/s, which was the lowest velocity achievable with the available pump. The flow was delivered using plastic tubes 5mm in diameter and plastic connectors were used as linkage between the tubes and the vessel phantom. After passing through the vessel, the flow solution was collected in a second reservoir.

In order to keep the two halves of the phantom together, a system with a metallic rod was used with two movable and flat elements mounted on it. The two elements were regulated in order to keep the phantom in a stable vertical position. The set-up included also an ultra-miniature optical pressure transducer connected to the Samba 201/202 control unit (Life Science, Gothenburg, Sweden) in order to continuously measure the upstream pressure. The pressure sensor was kept 4.5cm above the mesh for all the experiments and the data were collected using the Samba software installed on a portable computer. The sample frequency of the Samba unit was 10Hz and the measured pressure was the relative pressure, which was computed after calibration of the sensor against ambient pressure. This calibration procedure was performed before each test.

The clots were produced using venous blood drawn from a healthy volunteer and stored in 4.5ml tubes (BD Vacutainer, US) containing 0.5ml of citrate solution (0.105M), which prevents the blood

from coagulating. In order to produce clots, 750µl of blood were mixed with 60µl of $CaCl_2$ solution (0,756M) in plastic Eppendorf tubes [54]. The $CaCl_2$ solution was prepared mixing deionized water with granular anhydrous $CaCl_2$ (Sigma-Aldrich, St. Louis, MO, US) using a magnetic stirrer. The tubes containing the mixture of blood and $CaCl_2$ were than incubated at room temperature for 3 hours. The obtained clots were than cut in order to obtain a size comparable to the lumen diameter of the vessel (3-4mm approximately). As a complement to the pressure measurements, the clot mass loss (%) was computed after each experiment. The obtained clots before each test were bottled on an absorbent paper and weighed on a 0.001g precision scale (Sartorius, Goettingen, Germany). After each test the clots were extracted from the vessel (taking apart the two halves of the phantom), bottled and weighed again [55]. The clot mass loss (%) was expressed as the difference between the initial and final weight divided by the mass before the test.

The obtained in-vitro set-up was tested before the application of ultrasound, in order to assess its functionality. Two main problems arose after the realization of the entire set-up:

1. Air bubble formation within the flow immediately after clot insertion

2. Leakages of the saline solution from the vessel phantom

The air bubbles problem

The first problem was related to the fact that the insertion of the clots required to stop the flow, disconnect the uppermost tube delivering flow to the phantom and extract the connector from the artificial vessel. When the connector was extracted and tube disconnected, air entered in the system and air bubbles were formed. In order to assess the effect of the air bubbles, three tests were performed following the normal procedure during clot insertion:

- Disconnect the uppermost tube and extract the connector from the phantom

- Mimic the insertion of a clot

- Reposition the connector

- Perform the calibration against ambient pressure of the Samba sensor

- Connect the tube with the pressure sensor to the connector

- Start the 3rpm (8,75mm/s) flow and the pressure measurements

In the three tests, pressure measurements were performed only for 5min because after that time no variation in the pressure pattern was seen. This procedure enabled to see the consequences of air bubble formation from the pressure plots within the flow. These pressure measurements served as baseline data, to which the pressure data obtained during tests of possible solutions, were compared.

In order to eliminate air bubbles from the flow, two solutions were evaluated:

1. A flush was performed before starting the delivery of flow with the peristaltic pump

2. An infusion syringe pump delivering continuous flow was used instead of the peristaltic pump

The first solution was to use a flush to remove air bubbles. The flush was delivered by setting the peristaltic pump to run at a velocity equal to 30rpm for a time sufficient to remove the air bubbles within the flow. As before, three pressure measurements were performed for 5min after flush delivery.

The second solution for reducing air bubbles formation was to substitute the peristaltic pump with a syringe pump (Alaris, CareFusion, San Diego, California). Therefore, the set-up was changed in order to have continuous flow. The pump used was an infusion syringe pump and the flow velocity was set to 1.5ml/min. The syringe (50ml capacity) was connected to the plastic tube that delivered the flow to the phantom. The connector between the syringe and the pump had a stopcock (BD Medical, NJ, US) with an aperture allowing the possibility to connect a second syringe. Using the stopcock, it was possible to choose to deliver the flow from one of the two syringes. Therefore, one syringe (50ml capacity) was used for the delivery of saline solution and MBs, while the second syringe (20ml) allowed the injection of the clots. Also for the set-up with the syringe pump three pressure measurements were performed for 5 min after mimicking the injection of a clot with the 20ml syringe.

For each of the situations (normal condition, flush delivery and continuous infusion) the three pressure curves were filtered showing the pattern more clearly. The filtering was performed averaging 200 data samples around the output point using the MatLab function *filter*. The filtered pressure measurements were normalized to the first value and the mean normalized pressure over time for each situation was plotted together with the standard deviations bars.

The leakages problem

When the vessel of the PVA phantom was obstructed with a 3-4mm clot and the flow was delivered, the saline solution with and without MBs started to leak in the area where the two halves of the phantom were connected. This leakage was present also when compressing the two halves with a higher force. The leakage was probably due to the low stiffness of the phantom and the high pressure generated within the flow when the clot was present. In particular, when the pressure rose due to the clot obstruction, a higher force was applied on the vessel walls, which bent leading to a change in the overall shape of the phantom and allowing the leakage of the solution.

In order to have a stiffer phantom a different material was tested. The material was Elastosil M4601 silicone rubber (Wacker, Germany), obtained mixing two liquid components (90% component A and 10% component B). Moreover, an additional silicone fluid (AK 35, Wacker, Germany) was added (20% of the total) in order to obtain a lower stiffness. The mixture was poured in the same moulds used for the PVA phantom and left overnight to allow solidification of the mixture. The silicone phantom and the PVA phantom were than tested in the previously described set-up. Three tests were performed per each of the two types of phantoms. Firstly, clots 4mm in size were bottled and weighed. Secondly, the clots were inserted in the vessel disconnecting the uppermost connector. Before reconnecting the tube the calibration of the pressure sensor against ambient pressure was performed. Afterwards, the tube with the pressure sensor was connected to the phantom and an initial flush was performed. After that, saline solution was delivered at 8.75mm/s for 20min and the pressure measurements were recorded.

Beside the leakage testing, an additional test was performed in order to estimate the silicone

rubber attenuation coefficient. For this estimation, a flat sample (thickness 10mm) of the silicone rubber was positioned in deionized water on a metal reflector. A single element transducer (delivering ultrasound at frequency of 2.25MHz) was positioned opposite to the reflector and perpendicularly to the sample. The ultrasound was delivered using an ultrasonic pulser-receiver (Olympus, NJ, US).

The frequency spectrum was recorded using an oscilloscope (Tektronix, Beaverton, US). The frequency spectrum was computed firstly with only the metal reflector without the sample and, secondly, with the metal reflector and the sample. The two spectra were than used for computing the attenuation coefficient knowing the distance of the transducer form the sample (10cm) and the ultrasound velocity in water (1500m/s).

2.2 Results

The air bubbles problem

Figure 2.4 presents the mean normalized pressure pattern for the normal condition as for the two presented solutions (flush delivery and infusion pump application). The high standard deviation in pressure for the normal condition verifies the instability caused by the presence of air bubbles and their travelling within the flow. When using the flush and the continuous infusion the mean normalized pressure had approximately a constant value during the time considered. Small standard deviation bars verifies the more stable situation.

The application of the continuous infusion pump led to a set-up more similar to the physiological situation, where no flushes are present and the velocity of the flow is much lower than the velocity considered using the peristaltic pump. Moreover, qualitative observations enabled to notice reduced leakages using the continuous infusion pump.
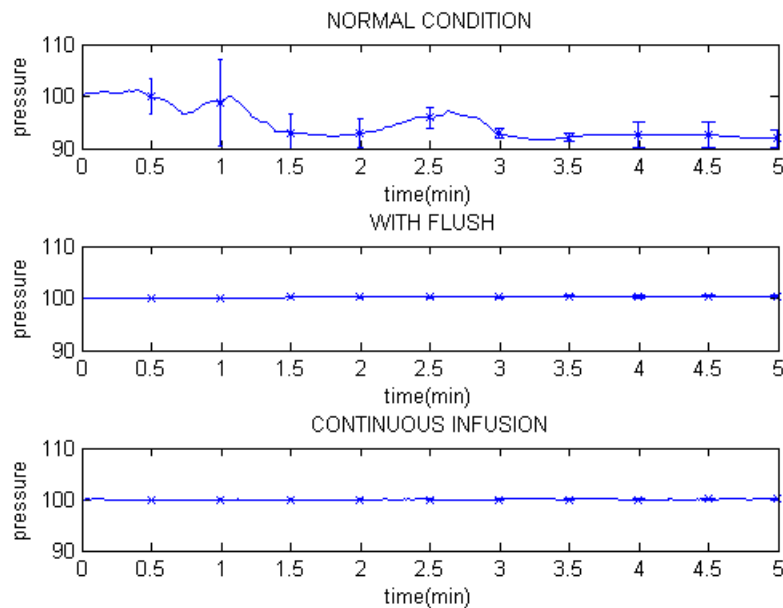
Figure 2.4: Mean normalized pressure pattern for the three situations: normal condition (upper) using the peristaltic pump, after delivery of the flush (mid) with the peristaltic pump and continuous infusion (lower) with the syringe pump.

The leakages problem

The phantom made of silicone rubber, when connected to the peristaltic pump after clot insertion, led to reduced leakages, however not eliminating completely the problem. The silicone rubber material was stiffer compared to the phantom made with PVA and graphite, thus the vessel walls

bent less. In Table 2.1 the initial clot mass are reported for both the types of phantom and for each test together with the mean pressure computed during flow delivery.

Table 2.1: Initial clot mass and mean pressure value for each test during the flow delivery

SILICON PHANTOM	INITIAL MASS (g)	MEAN PRESSURE VALUE ± SD (mm Hg)	PVA PHANTOM	INITIAL MASS (g)	MEAN PRESSURE VALUE ± SD (mm Hg)
TEST 1	0.022	44.56 ± 3.91	TEST 1	0.029	-7.85 ± 0.48
TEST 2	0.020	70.91 ± 1.15	TEST 2	0.020	-11.22 ± 0.96
TEST 3	0.020	24.28 ± 2.33	TEST 3	0.025	-9.29 ± 0.52

The higher values of the mean pressure considering the silicon rubber phantom verify the reduced leakage. However, even if the general mean value of the pressure in the tests with silicone rubber phantom was higher than in the test with PVA phantom, different pressure mean values for the same phantom and similar clots weight were found. For example, in TEST 2 and TEST 3, using the rubber phantom, the initial clot weight was equal (0.02g). However, the mean pressure value obtained for the TEST 3 and computed after the flush was lower than the mean value obtained from the TEST 2. This was due to the leakage still present with the silicone rubber phantom during flow delivery.

Beside considerations of the ability of silicone rubber in reducing leakages, the acoustic attenuation coefficient was also computed. The graph of the attenuation coefficient values expressed in dB/cm to the respect of the frequency (MHz) is shown in Figure 2.5. Since the single element transducer used was delivering ultrasound at 2.5MHz, the obtained attenuation coefficient was computed considering this frequency. Therefore, at 2.5MHz the attenuation coefficient value was 8.05dB/cm. Assuming a linear dependence between the attenuation coefficient and the frequency, the final attenuation coefficient obtained was $3.2\text{dBcm}^{-1}\text{MHz}^{-1}$, which was higher compared to the attenuation coefficient of soft tissue (approximately $0.5\text{dBcm}^{-1}\text{MHz}^{-1}$). Switching to a stiffer phantom could reduce the leakage, but the acoustic properties of the silicon material highly reduced the ultrasound intensity in the clot region. With a $3\text{dBcm}^{-1}\text{MHz}^{-1}$ attenuation coefficient, considering the silicon rubber material, the surrounding tissue thickness required for having low attenuation was too small. The good compromise between stiffness and acoustic properties was not found. Therefore, the silicon rubber material was not considered in the final set-up.
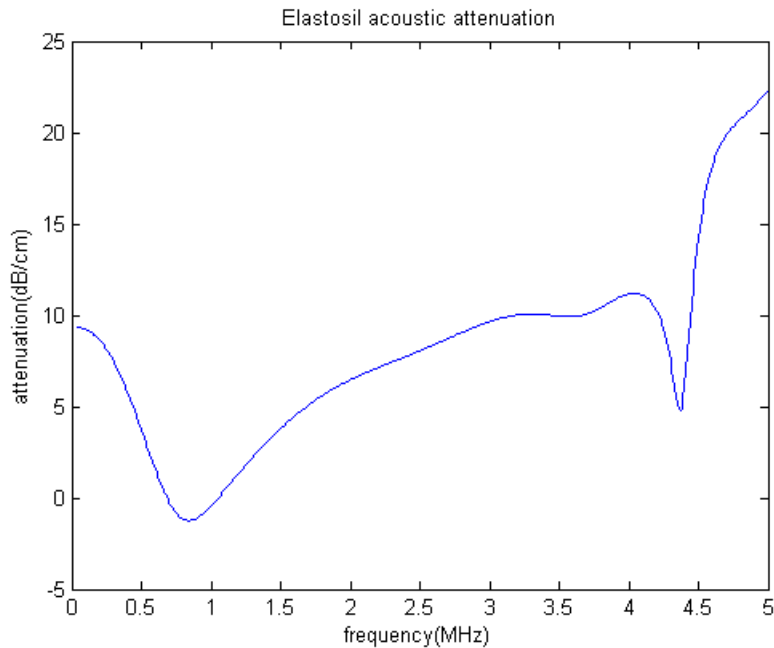
Figure 2.5: Acoustic attenuation of the silicon rubber material excited with ultrasound at 2.5MHz.

2.3 The final in-vitro set-up

Considering the results obtained during the optimization and validation of the initial in-vitro set-up a new final set-up was defined as shown in Figure 2.6. The silicone rubber material, even though was able to reduce the leakage problem, had a too high attenuation coefficient in order to be used during the ultrasound application. In order to have the same decrease in the transmission voltage as for PVA phantom (43% in the mesh region), the thickness required for the silicon vessel phantom was too small, leading to an unstable phantom. Therefore, the phantom made of PVA, graphite and deionized water was used in the final set-up, with the wall thickness equal to 25mm or 10mm, depending on the frequency applied. The final set-up was similar to the initial one with the exception of the pump used for the delivery of flow. Instead of the peristaltic pump, the infusion pump mentioned earlier was used. The new pump, delivering flow with lower velocity (2mm/s), reduced the instability of the set-up due to the leakages phenomena, solving at the same time the air bubble problem. Indeed, injection of clots using a syringe and a connector with a stopcock stopped the air form entering in the system. Moreover, since the plastic of the tube was slightly translucent, it was possible to see the clots when injected and to stop compressing the syringe when they entered in the phantom vessel. Therefore, delivering the clots in this way did not require disconnecting the uppermost connector of the phantom, thus avoiding the formation of air bubbles.
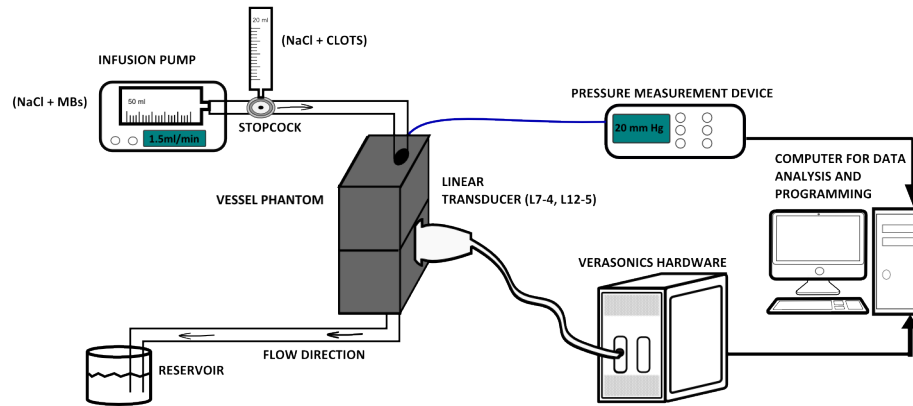
Figure 2.6: The final in-vitro set-up

Chapter 3

Clot lysis tests with ultrasound and MBs

The set-up obtained after the optimization process (Figure 2.6) was used for the tests of clot lysis with ultrasound application in conjunction with MBs. Two types of MBs, the 3MiCRON MBs and SonoVue MBs, were considered.

The ultrasound sequences were programmed using the Verasonics architecture (Verasonics, Inc, Redmond, WA, USA) and delivered through the L12-5 50mm and L7-4 transducers (ATL/Philips, Bothell, WA, USA).

3.1   Methods

The two types of MBs were diluted in saline (NaCl) in order to have a concentration of $2x10^6$MBs/ml as stated in [9]. Afterwards, MBs and NaCl solution were delivered to the vessel phantom using the 50ml syringe and the infusion pump. When the MBs were seen in the clot area (applying B-mode imaging), the sequence of long pulses was started and after 20min the clot mass loss was computed. The upstream pressure was continuously recorded for 20 minutes. The pressure measurements and clot mass loss were compared in order to give an estimation of the thrombolytic efficiency. The time for the tests was chosen in accordance to [9, 27] where it was seen that this time was sufficient for the detection of the thrombolytic efficiency.

During the tests, the ultrasound transducer was fixed by a metallic holder and kept in contact with the phantom surface, which was previously covered with ultrasound gel. The transducer was positioned longitudinally to the vessel.

In order to develop the long pulse sequences that were transmitted to the clot area, a code was programmed using the Verasonics software. The code was written in *MatLab* (MathWorks, Massachusetts, US) and completely reported in the Appendix.

In the following sections the MBs production, the implemented protocols and the Verasonics ultrasound system architecture are described.

3.1.1 Production of 3MiCRON MBs

The production of the novel agent was described in [25] and was performed preparing two batches that were later added together. For each batch, 200 ml of MilliQ water were mixed with 4g PVA (2%) and heated up to 80 °C. When the mixture reached the desired temperature, 380mg of $NalO_4$ were added and the solution was kept at 80°C for 1 hour continuously agitated with a magnetic stirrer. After 1 hour the mixture underwent high shear stirring (8000rpm for 3 hours) in order to selectively split the head-to-head sequence contained in the PVA chains. This stirring was achieved using an Ultra Turrax (IKA, Germany) at room temperature. The resulting batch was washed seven times every 24 hours. The obtained MBs have a mean diameter of 3µm and they can be stored in water for months due to their high stability. Before their application, MBs concentration was assessed using a light microscope and a counting chamber as described in the protocol presented by Pretzl and Cerroni in [56]. Briefly, MBs solution was diluted (1:5) and transferred into the Neubauer counting chamber that has squares with an area of 0.25x0.25mm and a depth of 0.1mm. The images (Figure 3.1) were taken with a transmission microscope (20x objective) and transformed in binary pictures using ImageJ Analysis Software. The binary pictures were than used by the software in order to count the particles. Four squares were imaged and analyzed and the mean number of particles was used for the computation of the concentration. The obtained concentration was 10^9MBs/ml.
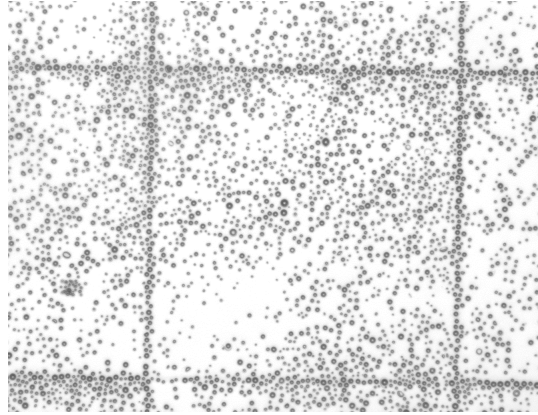
Figure 3.1: Image of 3MiCRON MBs with transmission microscope (objective 20x)

3.1.2 Production of SonoVue MBs

The protocol for the SonoVue MBs preparation was fixed by the manufacturer that delivers the SonoVue contrast agent as a kit with a vial containing a lyophilized powder (25mg) and a syringe prefilled with sterile saline solution (5ml). To prepare the MBs the saline solution and the powder were mixed for 20s. The obtained suspension can be stored for 6 hours and if the MBs accumulate at the upper surface the suspension should be shacked again before the application. The obtained MBs have a concentration of 1-5x10^8MBs/ml.

3.1.3 Implemented protocols for ultrasound transmission

In order to apply ultrasound with a frequency close to the resonance frequency of the two types of MBs (1-4MHz for SonoVue and 12MHz for 3MiCRON), the L12-5 50mm transducer transmitting at a frequency equal to 11.25MHz (highest value achievable) and the L7-4 transducer transmitting at 4.09MHz (lowest value achievable) were used. Four protocols for ultrasound transmission were developed and tested. For each of the protocols seven tests were performed. MBs were delivered after clot injection and, when they were seen in the clot region, ultrasound exposure was performed for 20 minutes. Moreover, seven control tests with no ultrasound and no MBs were executed and used as reference. In Table 3.1 the different protocols are summarized.

Table 3.1: The protocols developed and tested

MBs TYPES	TRANSDUCER/ FREQUENCY (MHz)	VOLTAGE (V)	PULSE DURATION (ms)	DISTANCE TRANSDUC- ER/CLOT (mm)	NUMBER TRANSMIT- TING ELEMENTS
3MiCRON	L12-5 50mm/11.25	100	5	10	56
3MiCRON	L7-4/4.09	100	5	25	40
SonoVue	L7-4/4.09	100	5	25	40
SonoVue	L7-4/4.09	50	10	25	40

The ultrasound focus was set within the clot region and the long pulses were sent from a subset of transducers elements in order to cover an area of 10-11mm that included the clot and a small surrounding region.

The distance between the transducer and the clot area was approximately 10mm in the first protocol, where the L12-5 50mm transducer was used. In the remaining protocols, since the frequency applied was lower, the distance was set equal to 25mm. The highest voltage applied during the long pulse delivery was equal to 100V, the highest value achievable, due to safety aspects of the system.

Between the long pulses, idle time was set in order to enable the MBs replenishment. Indeed, with a flow velocity equal to 2mm/s, the time necessary for the MBs replenishment was approximately 6s.

During the exposure of the clots to ultrasound and MBs, no imaging was performed except at one time point to check the effective MBs replenishment. This imaging was performed during the idle time between two trains of pulses using the already built-in MatLab script provided with the Verasonic software and opportunely modified for having imaging only during the idle time and not during the transmission of the long pulses. In this way, in the case of MBs destruction within the clot region during the long pulse delivery, imaging enabled to see if the idle time was sufficient for letting new MBs to reach the area.

For each group of tests the clot mass loss was computed and the mean initial mass and mean mass reduction with the respective standard deviation was calculated. Statistical analysis was performed applying the student's t-test in order to compare the mass loss for each protocol with the control group. Statistical significance was defined as $p < 0.05$.

Beside clot mass loss, three pressure measurements for three different tests within each protocol were considered. The pressure plots were only three due to the difficult reproducibility of the measurements caused by the leakages problem.

The pressure measurements of the three considered tests were filtered using an average filter. The filtering was performed averaging 200 data around the output value using the MatLab function *filter*. Afterwards, pressure plots were normalized to the first value and the mean normalized pressure was plotted together with standard deviations bars. Normalization of the plots was necessary in order to allow comparisons due to the fact that slightly different values in the initial clot mass led to different initial pressure values.

Each long pulse delivered using the Verasonics system consisted in trains of pulses and in Table 3.2 their characteristics are summarized for all the protocols. In the next sections the explanation of each pulse sequence will be provided.

Table 3.2: Train of pulses characteristics.

PROTOCOL	SINGLE PULSE DURATION (μs)	IDLE TIME (μs)	NUMBER OF PULSES
3MiCRON and high frequency	88.18	35.82	40
3MiCRON and low frequency	240	7	20
SonoVue and low frequency	240	7	20
SonoVue with longer pulse and lower voltage	240	7	40

3.1.3.1   Control tests: no MBs and no ultrasound application

Control tests were performed delivering saline solution without MBs after clots injection. Ultrasound was used just for imaging purposes and the long pulses with high voltage were not applied. In order to image the clot area, the L7-4 transducer was used and B-mode images were obtained using the already built-in MatLab script provided with the Verasonic software.

3.1.3.2   3MiCRON and high frequency exposure

In this protocol, the 3MiCRON MBs and long ultrasound pulses (5ms), with a frequency equal to 11.25MHz and voltage set to 100V, were applied (Table 3.1, first row).

The long pulses were sent from the 56 central elements of the L12_5 50mm transducer (Figure 3.2). Each element transmitted trains of pulses obtained combining 40 pulses, each with duration of 88.18μs (Table 3.2, first row), which is the maximal single pulse duration achievable with the Verasonics architecture and with a frequency equal to 11.25MHz . The idle time between these pulses had duration of 35.82μs. Therefore, 40 pulses with 88.18μs duration and 35.82μs of idle time led to a train of pulses with total duration of 5ms. A schematic representation of the ultrasound pulse sequence is shown in Figure 3.3.
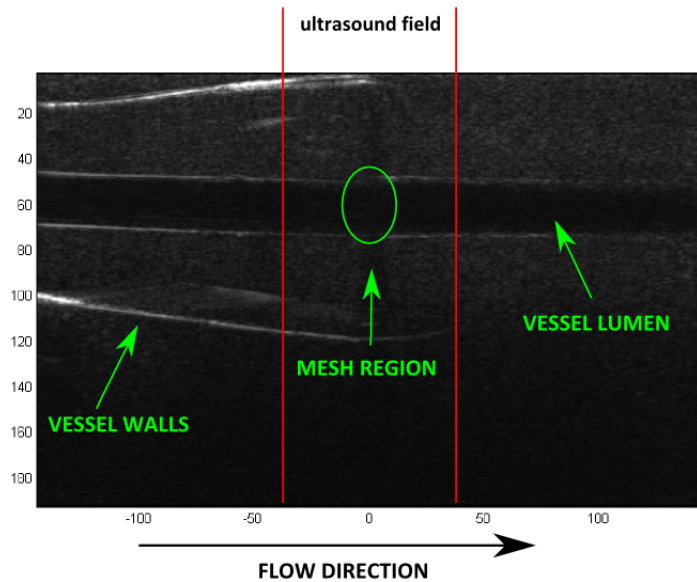
Figure 3.2: The ultrasound field using the L12-5 50mm transducer.
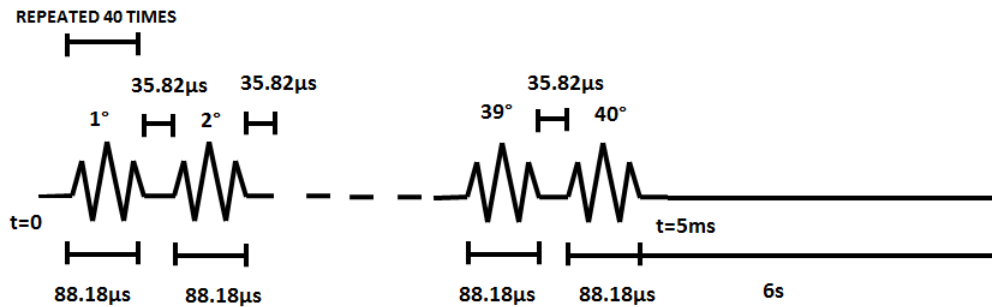
Figure 3.3: The ultrasound pulse sequence

3.1.3.3   3MiCRON/SonoVue and low frequency exposure

Two protocols were identical with the exception of MBs used. For both the types of MBs the frequency applied was 4.09MHz and the long pulse had duration of 5ms (Table 3.2, second and third row). The long pulses were sent from the 40 central elements (Figure 3.4) and they were obtained combining 20 pulses, each with duration of 240µs, which is the maximal length achievable with the Verasonics system and with a frequency equal to 4.09MHz (Table 3.2, second and third

rows). The idle time between these pulses had duration of 7µs. Therefore, combination of 20 of these pulses led to a train of pulses with duration of 5ms (Figure 3.5).
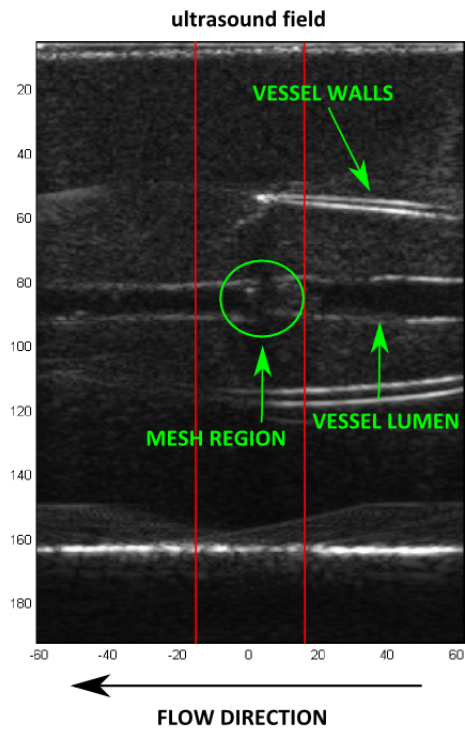
Figure 3.4: Ultrasound field with transducer L7-4.
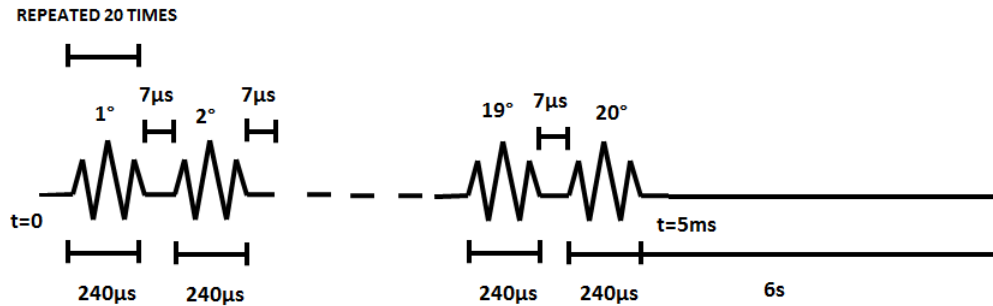
Figure 3.5: The ultrasound pulse sequence

3.1.3.4 SonoVue with longer pulse and lower voltage exposure

In this protocol SonoVue MBs were considered with ultrasound exposure at a frequency of 4.09MHz. The voltage was reduced by 50% and long pulse duration was doubled in comparison to the previous protocols (Table 3.2, last row).

The long pulses were sent from the 40 central elements and in order to achieve 10ms duration, 40 pulses of the same duration and the same idle time as in the previous two protocols were combined (Figure 3.6 and Table 3.2, last row).
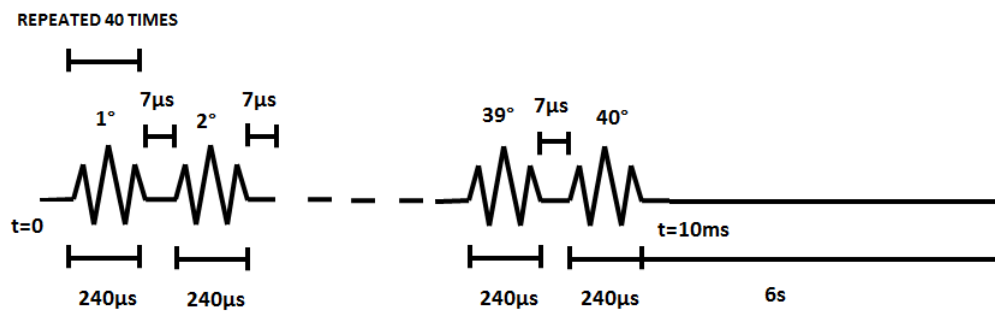
Figure 3.6: The ultrasound pulse sequence.

3.1.4   The Verasonics system

The aim of this section is not to provide a complete description of the Verasonics architecture, but only an overview in order to better understand the developed code. For a detailed explanation of the Verasonic system the reader should refer to the Verasonics Sequence Programming Manual.

The Verasonics system is an ultrasound system architecture (Figure 3.7) that enables implementation of traditional and new ultrasound sequences. Moreover, it is very flexible in transmitting, receiving and processing ultrasound. The Verasonics architecture allows also the simulation of the system using a software simulator running on the host computer. This enables the possibility to check the programmed sequence before running it with the probe and the hardware.
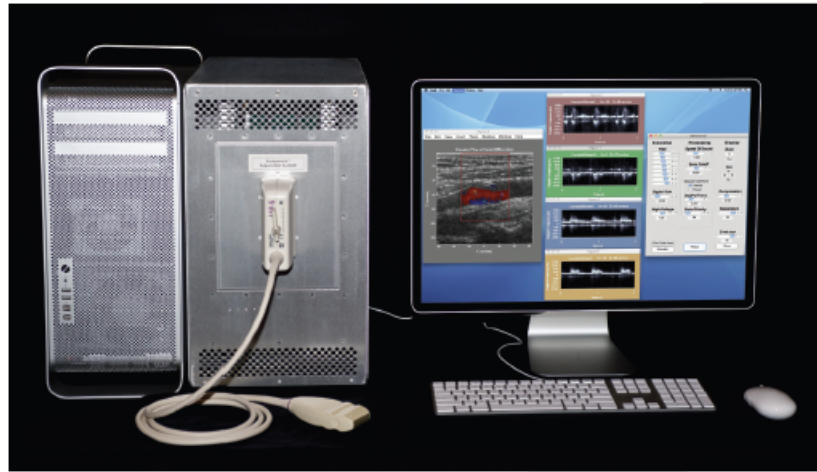
Figure 3.7: The Verasonics system (picture taken from the Verasonics brochure)

The Software-based System

The system is programmed generating a collection of objects which are defined using a MatLab script and characterized by specific attributes. After executing the script, a *MatLab.mat* file is generated, which can be loaded into the system by the VSX program (Verasonic Script eXecution). This program performs checks on the script and adds other attributes necessary for the Verasonics Data Acquisition System (VDAS) hardware. At the same time a GUI and a display window are displayed. The GUI panel allows the user to change some parameters such as voltage, gain and the dynamic range, while the display window reproduces the image of the region where the ultrasound waves are delivered.

In the final step the sequence is loaded into the hardware sequencer and executed.

The VDAS Hardware

The VDAS hardware is schematically shown in Figure 3.8 . The front of the available 4 board system is composed by a scanhead interface with two connectors for plugging the desired transducers. The 128 channels of the first connector are connected to the 128 A transmitters and the 128 channels of the second connector are connected to the 128 B transmitters. The connectors include a system for

detecting the presence of the probe and for allowing the execution of the sequence event without it. The acquisition modules are stored inside the chassis and they contain the circuitry for transmitting and receiving the signals for multiple channels. These modules contain also a local memory for the storage of the received signals before sending them to the host computer. The VDAS hardware contains also a Transmit Power Controller module which has the high voltage supplier for the transmitters. This module allows for high power bursts and extended pulse length.
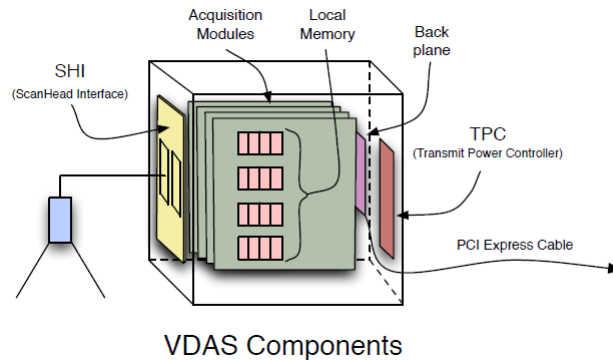
Figure 3.8: The VDAS hardware (picture taken from the Verasonics Tutorial)

The Objects definition

The objects that have to be defined in the MatLab script in order to run the sequence can be subdivided in two groups:

1. **The System objects**: this group of objects is needed in order to define the transducer used, the scanning format and the pixel data region. It also includes the resource object, which contains information about the storage buffers.

2. **The Sequence objects**: this set of objects include instructions for the transmission, reception, reconstruction and processing of the ultrasound. Moreover, through the *Event* object, it is possible to define the events for acquiring one or more frames of data.

Each object is defined and described by a set of attributes. For example, some of the *Transducer* attributes are:

- *name*: a string containing the scanhead name
- *frequency*: the center frequency expressed in MHz
- *type*: linear, curvilinear or 2D
- *elementWidth*: the element width expressed in wavelength
- *spacing*: the element spacing in wavelength
- *radius*: the radius of curvature

For a complete list of the attributes for the transducer and for all the other objects, the reader should refer to the Verasonics Tutorial.

3.1.5 The implemented codes

In order to program the Verasonics system and the transducer for the delivery of long train of pulses, the following objects have to be defined in the MatLab code: some initial resource parameters, the transducer, the storage buffer, the waveform, the transmission, the time gain controller, the receiving process, the control sequence and the list of the events.

For each of the protocols a code was implemented. The objects defined in these codes are presented in the following section. The complete code for each protocol is reported in the Appendix. The distance units in the code are expressed in wavelengths (λ) computed considering the central frequency of each transducer (5MHz for the L7-4 and 9MHz for the L12-5 50mm).

3.1.5.1 The *Resource.Parameters* attribute

The *resource.Parameters* stores some attributes that are global and required for the configuration of the system. These parameters were set equal for all the codes. In the developed sequences the parameters that had to be defined were the number of transmit and receive channels (128 for both) and the speed of sound (1540m/s). The *simulationMode* and *fakeScanhead* attributes, when set to 1, allowed running the program respectively in simulation mode or without the probe. These attributes enabled to check the pulse sequence before running the program with the hardware and the probe.

3.1.5.2 The *Transducer* object

For each ultrasound probe used, a *Trans* object was defined. Setting the *Trans.name* attribute equal to 'L12-5 50mm' and to 'L7-4' and running the function *computeTrans* led to the definition of transducers with the characteristics presented in Table 3.3 .

Table 3.3: The transducer attributes.

TRANS ATTRIBUTE	DESCRIPTION	L12-5 50mm	L7-4
numelements	Number of elements in the transducer	256	128
frequency	Central frequency of the transducer	9MHz (λ_1=0.17µm)	5MHz (λ_2=0.3µm)
spacing	The spacing between the elements, including the element width	$1.14\lambda_1$	$0.97\lambda_2$
maxHighVoltage	Maximal voltage (half of the peak to peak value)	50	50
impedance	Complex impedance at the central frequency	45-60i	45-60i

3.1.5.3 The *Resource* object

If imaging is not needed, no storage buffers are needed. However the VSX program requires that at least one storage buffer is defined. For this purpose, for all the protocols, one odd *RcvBuffer* was specified for the storage of the data from the local memories on the VDAS modules. However, these data were not further processed. The *RcvBuffer* contained a number of frames that was set to 1 in order to let the program to run.

3.1.5.4 The *TW* object

The *TW* object defines the transmit waveform for each transmitter. The type was set to 'parametric' because it had to be generated by the VDAS hardware. The parametric waveform was described by the four descriptors in the *TW.Parameters* (A, B, C and D), different for the two transducers used, and by the *extendBL* attribute, always set equal to 1. In Table 3.4 the values set for the two transducers are presented.

Table 3.4: The *TW* attributes. **A** value chosen according to Table 3.2.1.1 in the Verasonics Sequence Programming Manual.

TW ATTRIBUTE		DESCRIPTION	L12-5 50mm	L7-4
Parameters	**A**	waveform frequency expressed by the number of transmit clock periods in a half cycle	8 (11.25MHz)	22 (4.09MHz)
	B	the duty cycle period expressed by the number of transmit clock periods the positive or negative drive is active during the half cycle period	3 (37%)	8 (37%)
	C	the pulse length expressed by the numbers of half cycle periods (0-31)	31 (1.38µs)	31 (3.7µs)
	D	the polarity: 1 (first transmission positive) or -1 (first transmission negative)	1	1
extendedBL		If set to 1 lead to the multiplication of the C value by 64	1	1

3.1.5.5   The *TX* object

This object defines all the aspects about the transmit waveform and the timing for each of the active element in the transducer aperture. The attribute that were set equally in all the codes were the waveform, the origin and the transmitter delay. For each of the active transmitter the used waveform was the one defined in the *TW* object. For this reason the waveform attribute of the *TX* object was set to 1. In this way, all the transmit elements were characterized by the same waveform. The origin point, which is the point on the transducer that the beam appears to originate from, was defined setting the x, y, z coordinates equal to 0. The delay for each active transmitter was computed calling the function *computeTXDelays(TX(1))* that is already available within the Verasonic software.

The attributes that were set differently for the two transducers were the focus value, the *Apod* array for the selection of the active transmitters and the transducer aperture. These values are reported in Table 3.5.

TX ATTRIBUTE	DESCRIPTION	L12-5 50mm	L7-4
focus	Focal distance in wavelengths	$60\lambda_1$	$85\lambda_2$
aperture	Selects a sub-aperture of the full array aperture	65 (because transmit channels=128, while Trans.numelements = 256)	Not needed
Apod	Array with 0 (transmitter on) and 1 (transmitter off) values	56 central values equal to 1. The other 72 equal to 0.	40 central values equal to 1. The other 88 equal to 0.

Table 3.5: The *TX* attribute

3.1.5.6   The *TPC* object

The Verasonics *Transmit Power Controller* is an object that provides for rapid changes in high voltage output, allowing the use of different power levels for different modes of acquisition. Each power level can have its own set of unique attributes, including specification of transmit drive voltage level, and maximum power output. The collection of attributes for a given power level is referred to as a profile, and up to four profiles are supported in the current *TPC* firmware, with a fifth available as a "high power" option. The TPC profile 5 allows use of the system at very high transmit power levels. In particular, the "*Extended Burst Option*" utilizes a dedicated internal power supply with 48 Watt capacity to power the HIFU transmit bursts. A large energy storage capacitor provided within the system allows transmit for short bursts at much higher power levels, provided that enough time is allowed between bursts to recharge the capacitor from the internal supply. The amount of transmit power available is highly dependent on the transmit burst duration, the number of active transmit channels, and the load impedance presented by the transducer to the system. The profile 5, when defined in the code, enabled the use of the *extendBL* attribute, which was necessary for having longer pulses.

For the protocols where 100V were applied, the maximal voltage was set to 50V, meaning a peak to peak value of 100V, defining the *maxHighVoltage* and *highVoltageLimit* of profile 5 equal to 50.

3.1.5.7 The *TGC* object

When imaging is not needed, the *Time Gain Compensation* is not necessary. However, the Verasonic VDAS requires the definition of at least one *TGC* object that has to be referenced in the *Receive* object (see next paragraph). In the developed script the *TGC* object used was the same found in all the scripts already provided within the Verasonic software.

3.1.5.8 The *Receive* object

This object defines all the characteristics of the receive phase of an acquisition event. Therefore, when imaging is not required, this object is not needed. However, as the *RcvBuffer* had to be defined also the associate *Receive* structure had to be set in order to run the VDAS program. Therefore, the MatLab code was taken from an already built in script just to define a simple *Receive* object and let the VDAS to run.

3.1.5.9 The *Sequence Control* object and the *Event* object

The *SeqControl* object provides the flow control of the *Events* which are defined immediately after.

For all the codes the first *Event* that had to be defined was the one that enabled to switch to profile 5. The associated SeqControl commands used were '*setTPCProfile*' with the argument set to 5 and the '*noop*' command set to 8ms. The first command enables the transmission using the high power profile 5, while the second set 8ms for ensuring enough time to change to profile 5, due to the not immediate switching.

The Events for the delivery of the long train of pulses were slightly different for the different codes depending on the number of single pulses that were sent. For having 5ms train of pulses at 11.25MHz and 10ms at 4.09MHz, 40 events were required. on the other hand, for having 5ms train of pulses at 4.09MHz, 20 events were required.

After the train of pulses was sent, 6s of idle time were set for MBs replenishment. After this period of time the sequence was repeated from the second event (the one immediately after the setting of the profile 5) using the SeqControl command '*jump*' . When the command '*jump*' was called, an unconditional branch was performed to the event indicated in the argument which in the specific case was the number 2. In the end, the last events after the '*jump*' command were required in order to run the VDAS program. Indeed, this program required that at least one '*transferToHost*' command was set, generating a command needed by the VSX. The '*transferToHost*' command required that at least one receiving event was present. In order to satisfy this request a sham receiving event using the fake *Receive* object and *RcvBuffer* was set.

3.2   Results

Table 3.6 summarizes the initial mean weight and the mean mass loss, with the corresponding p value, for the control tests and the four protocols. Moreover, in Figure 3.9 the change in upstream pressure over exposure time is presented.

Table 3.6: Initial mean weight and mean mass loss for each protocol (control tests included). NS= not significant, n=number of clots tested.

PROTOCOL	INITIAL MEAN WEIGHT ±SD (g)	MEAN MASS LOSS ± SD (%)	p-value
Control tests (n=7)	0.064 ± 0.007	38 ± 9.6	-
3MiCRON and high frequency (n=7)	0.054 ± 0.01	37 ± 9.7	NS
3MiCRON and low frequency (n=7)	0.063 ± 0.01	37.2 ± 5.4	NS
SonoVue and low frequency (n=7)	0.051 ± 0.01	35.8 ± 6.3	NS
SonoVue with longer pulse and lower voltage (n=7)	0.063 ± 0.01	49 ± 7.9	0.025
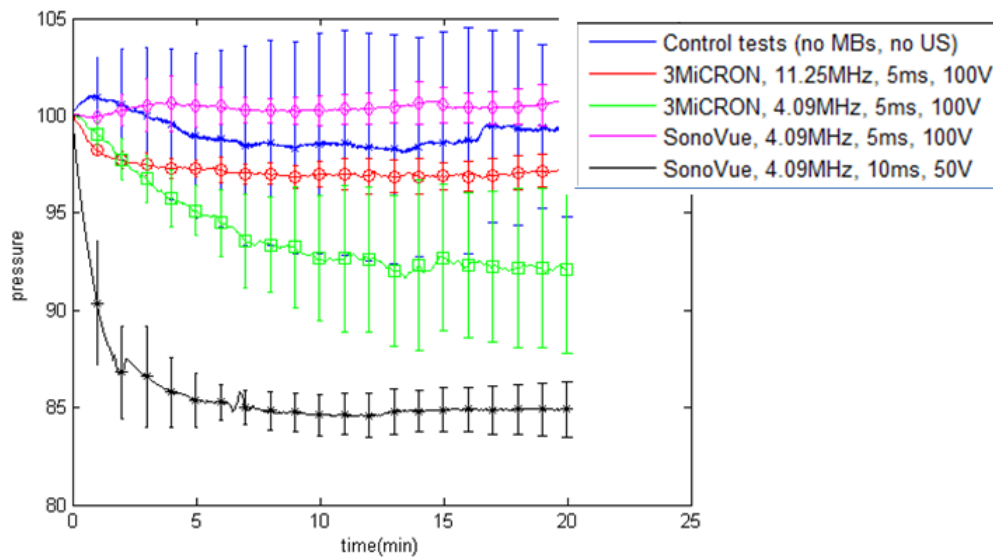
Figure 3.9: Mean normalized pressure plots for all the protocols (control tests included).

Clot mass loss was in general seen for all the protocols (control tests included). However this was not the case for the pressure measurement where decreased values were mainly seen for the SonoVue MBs exposed to low voltage. One possible explanation for this is that the saline flow produced clots debris which remained entrapped within the mesh, hence not allowing the flow to be restored and keeping the pressure to high values.

As it can be seen in Table 3.6, the first three protocols led to a mean mass loss comparable to the one obtained during the control tests and no statistical difference was seen between these protocols and the control tests. In the protocols where the 3MiCRON MBs were used, it was possible to see that there were no frequency-dependent variations in the clot mass loss. For both these protocols, it was possible to see form the images between long pulses, that the 3MiCRON MBs were not destroyed.

On the other side, in the protocol where SonoVue MBs were used with low frequency exposure, MBs destruction was noticed after pulse delivery.

In the last protocol lower SonoVue MBs destruction was seen. Moreover, increased clot lysis was noticed, as confirmed through the clot mass loss computation. Indeed, a 10% increase in the clot mass loss was seen compared to the control tests. The p value confirmed the statistical difference between this protocol and the control tests.

The mean pressure over time for this protocol was characterized by an early decrease in the first 4 minutes, followed by a plateau region. The early decrease is in accordance with [9], where the maximal thrombolyitic effect was seen within the first 3min. Furthermore, the mean normalized pressure presented a 15% decrease over the 20min of testing with standard deviations within acceptable ranges.

Chapter 4

Discussion and Conclusions

4.1 Design and optimization of the in-vitro set-up

As it can be seen from the results obtained during the validation process, the optimal set-up was not easy to achieve.

Several problems arose during the realization and the main ones were related to leakage problems and air bubble formation.

The air bubbles formation was effectively solved switching from the peristaltic pump to the infusion pump. Air bubbles led to the presence of some random peaks in the pressure patterns. After the flush and during the continuous infusion the pressure curves were more regular and no random peaks were present. Therefore, air bubbles formation was avoided. The solution chosen for the final set-up was to use the continuous infusion pump because, beside eliminating the problem of the air bubbles formation, it led to a more realistic set-up.

The leakage problems affected the pressure measurements, which were not always reliable as the clot mass computation before and after each test.

The final in-vitro set-up presented several aspects similar to the physiological situation of occluded coronary arteries. The vessel lumen was 4mm in size, comparable to the dimension of the left anterior descending coronary artery. Moreover, the 40µm pore mesh reproduced the microcirculation and the flow velocity was 2mm/s, similar to the real situation value. Lastly, the material of the phantom had acoustic properties comparable with the soft tissues ones.

On the other side, the distance between the clot region and the ultrasound transducer was not comparable to the real distance between the probe and the coronary arteries. Moreover, in the real situation the presence of the ribs and other body structures that can attenuate and scatter the ultrasound waves makes the application of linear transducers not possible. Another limitation of the in-vitro set-up was the continuous flow delivery, which is not present in the real situation where the flow is pulsatile. Moreover, the implemented set-up did not allow estimation of the clot debris stuck in the mesh.

4.2 Test of the in-vitro set-up with ultrasound and MBs

In general, using the Verasonics system, several limitations in the coding and implementation of different combinations of ultrasound parameters were found. This was due to the fact that the entire system had power limitations and hardware constraints that did not allow realization of continuous long pulses. Another general limitation for all the protocols (control tests included) was the low number of tests performed. Higher number of tests will enable more reliable statistics.

4.2.1 Control tests

The control tests, with no application of ultrasound and MBs led to a mean mass loss equal to 38 ± 9.6 %.

In another study [30], were ultrasound efficiency was studied in a set-up were no flow was present, the control test (20min of clot immersion in saline solution) led to a clot reduction of 29 ± 11%. The weighing procedure was similar to the one presented in this work. The higher clot mass loss found in the present project was probably due to the flow delivery, which may be able to further reduce the clot size. Another aspect to consider is that the clots produced in [30] are obtained letting human blood to coagulate spontaneously after being drawn. The different methods for clots production may be another reason for different values in clot mass loss. Moreover, the mean initial weight of the clots produced within this project was much lower than the mean weight of the clots found in [30], because of the 4mm diameter vessel. Indeed, the mean initial mass of the 3-4mm clots computed considering all the tests of all the protocols (control tests included) was 0.057 ± 0.01 g. In [30] that value was equal to 0.498 ± 0.073 with a size of approximately 1cm. The small dimension of the clots used in the present project led to a more difficult mass loss computation.

Comparing the results obtained computing the clots mass loss and the pressure measurements during the control tests it seemed that the pressure sensor was not able to detect the 38% mass loss. Therefore, potential clot debris was probably bigger in size than the mesh pores (40µm), hence remaining stuck in the mesh and keeping the pressure to high values. In a real physiological situation this will lead to the occlusion of the microcirculation (approximately 40µm in size). A further study with a microscope, in order to analyze the mesh, may confirm this possibility.

4.2.2 3MiCRON with high and low frequency exposure

The frequency value of 4.09MHz, applied delivering 3MiCRON MBs was lower than the resonance frequency. On the other side, the highest frequency applied was closer to the resonance frequency, but still not equal due to the transducer used and the Verasonics hardware limitation. The enhancement of the non-linear oscillation of 3MiCRON MBs is maximal when the resonance frequency is applied. In the tests performed the MBs might have passed the clot and the mesh without any effect when delivering ultrasound at 4.09MHz and also the highest value may has not been sufficient. If non-linear oscillation is considered as one possible reason for clot lysis, in the considered tests the oscillation may has not been enhanced sufficiently in order to increase the clot reduction.

In [30], application of ultrasound without MBs led to decreasing thrombolysis efficiency with increasing frequency. Considering this result, a limitation of the 3MiCRON MBs might be the need of very high frequencies in order to trigger an oscillation that can improve the clot lysis.

For both the protocols where 3MiCRON MBs were applied, the intensity delivered to the clot area was reduced by 43% of the transmitted value. Therefore, of the 100V sent from the transducers, 57V reached the clot. The voltage is the only indication of the ultrasound wave amplitude. No

information about the intensity or pressure was available due to the fact that the Verasonics system enabled only the setting of the voltage. Thus, the mechanical index could not be computed and the comparison with other studies mentioned in the background section was difficult. The images after the long trains of pulses did not show regions were no bubbles were present and one possible explanation of this could be that the intensity was not enough high to disrupt the MBs. 3MiCRON MBs have much stiffer shell than other lipid-encapsulated MBs, hence they may require very high values of intensity, not achievable with the L7-4 and L12-5 50mm transducers and the Verasonics system.

In [9], a peak-to-peak acoustic pressure of 1.5MPa led to high thrombolytic efficiency in the first 5min of testing with 5ms pulse duration. In detail, the pressure decreased by 75% over 20min of testing. In the present work no pressure decrease was found, thus the value of 57V in the clot region might not be sufficient for triggering the cavitation process or MBs oscillation. Moreover, in [34] it was seen that cavitation threshold increased with increasing frequencies values. Therefore, if the cavitation is supposed to be one of the main mechanisms of clot lysis, in the present study the threshold could have been greatly increased applying a frequency equal to 11.25MHz.

The ultrasound train of pulse with duration of 5ms was chosen in accordance to the value used in [9]. However, in [9] the pulse length was delivered form a single element transducer and the 5ms pulse was continuous. In the present work the 5ms pulse was composed by several shorter pulses with small idle times in between. More in detail, using the L12-5 50mm transducer, the longest pulse achievable with 11.25MHz had a duration of 88.18µs, while the L7-4 transmitting at 4.09MHz enabled pulses with maximal duration of 240µs . Thus, in order to obtain a longer pulse, several of these short pulses were combined. The idle time in between was the lowest achievable and smaller values were not possible due to power limitations. Moreover, each pulse had a duty cycle equal to 37%. Low values of duty cycle led to irregular waveforms, thus the final ultrasound wave did not have a regular sinusoidal shape with alternating phases of compression and rarefaction.

Due to several power limitation of the system, good compromise between voltage, pulse duration, duty cycle and idle time between pulses had to be found. In order to transmit at 100V, the train of 88.18µs and of 240µs pulses could not be longer than 5ms. All these limitation in pulse length may affect the ultrasound efficiency in clot lysis and ultrasound interaction with MBs.

4.2.3 SonoVue and low frequency exposure

For the SonoVue MBs it was seen that the pulse burst the bubbles, thus creating a dark area within the flow where almost no bubbles were present. This result is in accordance to previous knowledge, namely that SonoVue bubbles are easier to break compared to the 3MiCORN MBs, which remained intact when the same ultrasound pulse was delivered.

As seen in the results, this protocol led the conclusion that no improvement in clot lysis was seen compared to the control tests. One possible explanation of this result is that the 100V pulses were able to destroy the SonoVue bubbles without triggering any process of clot lysis. As stated in the Background, several studies found that inertial cavitation (generated when MBs are disrupted) was less effective in clot lysis compared with stable cavitation. Stable cavitation was probably not possible for SonoVue MBs due to their disruption. This could have led to ineffective clot reduction.

The frequency applied, 4.09MHz was almost in the range of the resonance frequencies for SonoVue (1-4MHz), but the disruption of the bubbles did not allow evaluation of the effects of MBs oscillation.

4.2.4   SonoVue with longer pulse and lower voltage exposure

According to the results obtained in the previous protocol, a new pulse sequence was developed for the ultrasound application with SonoVue MBs. The lower voltage was set in order to reduce the bubble destruction which was still present but with lower severity. Furthermore, reducing the voltage, it was possible to program a train of pulses with longer duration (10ms). Longer pulses were implemented because they were seen effective in clot lysis in several studies. In [9], five pulses with increased duration (from 0.1ms to 5ms) were compared, resulting in highest thrombolyitic efficiency with 5ms duration. Moreover, also in [30] it was seen that thrombolysis efficiency (%) increased with pulse duration.

As seen in the results, the implemented pulse sequence, in conjunction with SonoVue MBs was able to improve the clot lysis. This may be due to the fact that lower voltage reduced the MBs destruction, while the longer pulse duration enabled the enhancement of MBs interaction with the clot. The MBs that were not destroyed were triggered with a pulse which, with longer duration and with a frequency within the range of the resonance frequency for the SonoVue, could have improved the MBs oscillation. Furthermore, the MBs oscillation could have enhanced the cavitation within the clot area, increasing the clot lysis.

4.3   Conclusion

In the present work, a set-up for sonothrombosis was designed and tested.

The set-up presented several aspects that enabled its comparison with the physiological situation of coronary arteries occlusion, allowing testing of clot lysis with ultrasound exposure in conjunction with MBs. However, the set-up had also several limitations and two main problems were found: air bubbles formation within the flow and leakages phenomena. The first one was completely solved, while the leakage was only reduced.

The obtained set-up and the implemented ultrasound sequences were able to allow a first study of the sonothrombolysis with 3MiCRON and SonoVue MBs. From the developed protocols it was seen that 10ms long pulses, delivered at 50V in conjunction with SonoVue MBs, were able to improve the clot lysis by 10% compared to the control tests and to the tests where the 3MiCRON MBs were used.

4.4   Future work

The set-up has to be further optimized in order to eliminate the leakage problem and to obtain more reliable pressure measurements. Different phantom materials and geometries should be considered and tested in order to address these problems. Furthermore, the efficiency of the clot lysis should be computed considering also the clot debris stuck in the 40µm pore mesh. Developing a method for analyzing the material entrapped in the mesh, it will be possible to see if the reason of not varying pressure over 20min was due to clot debris bigger in size than the mesh pores.

The programmed ultrasound sequences have to be further modified and tested in order to evaluate possible effects in clot lysis with different combination of ultrasound parameters. Different pulse lengths, voltages and frequencies should be tested in order to find the right combination for higher thrombolytic effects when delivering 3MiCRON or SonoVue MBs. In addition, the application of different transducers should be considered. For example, using a single-element

transducer it will be possible to have fewer limitations compared with the programmable system, although the setting will be less realistic when compared to the clinical one.

Bibliography

[1] Media centre, WHO, The 10 leading causes of death by broad income group (2008), Updated June 2011, http://who.int/mediacentre/factsheets/fs310/en/ (accessed 07/06/2013).

[2] Walter F. Boron, Emile L. Boulpaep, "Special circulation" in Medical Physiology, Updated Edition, Philadelphia: Elsevier Saunders 2005, ch 23, 562-565

[3] Lippi G et al., "Arterial thrombus formation in cardiovascular disease", Nat. Rev Cardiol., 2011; 8(9); 502-512

[4] Robert S. Shwartz, "Microemboli and microvascular obstruction in acute coronary thrombosis and sudden coronary death", Journal of the American College of Cardiology, 2009; Vol 53; No 23.

[5] Aldons J. Lusis, "Atherosclerosis", Nature,2000, vol 407; 233-241.

[6] Legein et al., "Inflammation and immune system interactions in atherosclerosis", Cell. Mol. Life Sci., 2013, DOI 10.1007/S00018-013-1289-1.

[7] Lincoff AM, Topol EJ, "Illusion of reperfusion.Does anyone achieve optimal reperfusion during acute myocardial infarction?", Circulation, 1993; 88; 1361-1374

[8] Ito H., "No reflow phenomenon and prognosis in patients with acute myocardial infarction", Nat. Clin. Pract. Cardiovasc. Med, 2006; 3; 499-506.

[9] Leeman E. et al., "Effective of acoustic conditions on microbubble-mediated microvascular sonothrombolysis", Ultrasound in Med and Biol, 2012; Vol 38; 9; 1589-1598

[10] Janice L. Hinkle, "Acute Ischemic Stroke review", J Neurosci. Nurs., 2007; 39(5); 285-293, 310.

[11] Muir et al, "Imaging of acute stroke", Lancet Neurology, 2006; 5; 744–768.

[12] Wolf-Dieter Heiss, "Systemic rtPA in Patients with Acute Ischemic Stroke", Max-Planck-Institut für neurologische Forschung, Köln, Germany [Online]. Available: http://www.fac.org.ar/scvc/llave/stroke/heiss/heissi.htm (accessed 15-02-2013)

[13] Marler JR, "Tissue plasminogen activator for acute ischemic stroke", N Engl J Med; 1995; 333; 1581-1587

[14] Blakeley JO, Llinas RH. "Thrombolytic therapy for acute ischemic stroke". J.Neurol.Sci. 2007;261:55-62.

[15] H.P. Adams et al., "Guidelines for Thrombolytic Therapy for Acute Stroke: A Supplement to the Guidelines for the Management of Patients With Acute Ischemic Stroke A Statement for Healthcare Professionals From a Special Writing Group of the Stroke Council, American Heart Association" Stroke; a journal of cerebral circulation, 1996; 27; 1711-8.

[16] Robert J. Siegel, Huai Luo, "Ultrasound thrombolysis", Ultrasonics,2008; 48; 312-320

[17] Denise E et al., "Advancements in Catheter-Directed Ultrasound-Accelerated thrombolysis", J Endvasc Ther, 2011; 18; 418-434

[18] Hirsh J et al., Executive summary: American College of Chest Physicians evidence-based clinical practice guidelines (8th edition), Chest. ,2008; 133(6 SUPPL. 6); 110S-112S

[19] Hoskins P.R. et al., "Physiscs", in Diagnostic Ultrasound-Physics and equipment, London, UK, GMM, 2003, ch 1, 7-22.

[20] Ultrasound - Technology INformation Portal, http://www.us-tip.com/serv1.php?type=db, last update 22/04/2013 (accessed 07/06/2013)

[21] Greis, "Technology overview: SonoVue (Bracco, Milan)", Eur. Rad. Suppl, 2004;14[Suppl 8];11-15.

[22] Shengping Qin et al.,"Ultrasound contrast microbubbles in imaging and therapy: physiscal principles and engineering" Phys Med Biol, 2009; 54(6): R27. DOI:10.1088/0031-9155/54/6/R01.

[23] Villanueva FS et al, "Targeted ultrasound imaging using microbubbles", Cardiol. Clin., 2004; 22(2); 283-98.

[24] Streeter Jason E. et al., "Improving sensitivity in ultrasound molecular imaging by tailoring contrast agent size distribution: in vivo studies", Mol Imaging, 2010; 9(2); 87-95.

[25] Cavalieri et al., "Stable Polymeric Microballoons as Multifunctional Device for Biomedical Uses: Synthesis and Characterization", Langmuir, 2005;21; 8758-8764.

[26] Kathryn E. Hitchcock and Christy K. Holland, "Ultrasound-Assisted Thrombolysis for Stroke Therapy", Stroke, 2010;41; S50-S53.

[27] Francis CW et al., "Ultrasound accelerates transport of recombinant tissue plasminogen activator into clots" Ultrasound Med Biol., 1995; 21; 419-424.

[28] Saguchi T et al., "Effective and safe conditions of low frequency transcranial ultrasonic thrombolysis for acute ischemic stroke. Neurologica and histologic evaluation in a rat middle cerebral artery stroke model", Stroke, 2008; 39;1007-1011.

[29] Tachibana K, Tachibana S., "Albumin microbubble echo-contrast material as an enhancer for ultrasound accelerated thrombolysis", Circulation, 1995;92;1148-50.

[30] Schafer et al., "Influence of Ultrasound operating parameters on ultrasound-induced thrombolysis in vitro", Ultrasound in Med. And Biol, 2005; vol 31, No 6; 841-847.

[31] Rosenschein U et al., "Study of the mechanism of ultrasound angioplasty from human thrombi and bovine aorta" Am. J. Cardiol.,1994; 74; 1263-1266.

[32] Datta et al., "Correlation of cavitation with ultrasound enhancement of thrombolysis", Ultrasound Med Biol, 2006;32(8); 1257-1267.

[33] Collis et al., " Cavitation microstreaming and stress fields created by microbubbles", Ultrasonics, 2010;50; 273-279.

[34] Deng et al., "In vitro measurements of inertial cavitation thresholds in human blood", Ultrasound Med BIol, 1997; 23; 619-624.

[35] Borelli et al, "Influences of Microbubble Diameter and Ultrasonic Parameters on in Vitro Sonothrombolysis Efficacy", J Vasc Interv Radiol, 2012; 23; 1677-1684.

[36] Stefan Pfaffenberger et al.,"Ultrasound thrombolysis", Thromb Haemost, 2005;94; 26-36.

[37] Tachibana K., Tachibana S., "Enhancement of fibrinolysis with ultrasound energy", J Vasc Interv Radiol, 1992; 3; 299-303.

[38] Tachibana K., Tachibana S., "Prototype therapeutic ultrasound emitting catheter for accelerating thrombolysis", J Ultrasound Med, 1997; 16; 529-35.

[39] Lynn JG et al., "A new method for the generation and use of focused ultrasound in experimental biology", J Gen Physiol.,1942; 28; 179-193.

[40] Rosenschein et al., "Ultrasound Imaging-Guided Noninvasive Ultrasound Thrombolysis: Preclinical Results", Circulation, 2000; 102; 238-245.

[41] Frenkel et al., "Pulsed High-Intensity Focused ultrasound Enhances Thrombolysis in an in-vitro model", Radiology, 2006;239(1);86-93.

[42] Pfaffenberger et al., "Can a Commercial Diagnostic Ultrasound Device Accelerate Thrombolysis?: An in Vitro Skull Model", Stroke, 2005; 36; 124-128.

[43] Xie et al., "Treatment of Acute Intravascular Thrombi With Diagnostic Ultrasound and Intravenous Microbubbles", JACC Cardiovasc Imaging, 2009; 2(4); 511-8.

[44] Xie et al., "Diagnostic Ultrasound Combined with Glycoprotein 2b/3a Targeted Microbubbles Improve Microvascular Recovery Following Acute Coronary Thrombotic Occlusions", Circulation, 2009; 119(10); 1378-1385.

[45] Kutty et al., "Microbubble Mediated Thrombus Dissolution with Diagnostic Ultrasound for the Threatment of Chronic Venous Thrombi", PloS One, 2012; 7(12); e51453.

[46] Nesser et al., "Therapeutic Ultrasound in Cardiology", Herz, 2002; 27; 269-278.

[47] Skoloudik et al., "Safety and efficacy of the sonographic acceleration of the middle cerebral artery recanalization: results of the pilot thrombotripsy study", Ultrasound Med Biol, 2008;20; 58-63

[48] Bardon et al., "Safety and efficacy of sonothrombolysis using bilateral TCD monitoring by diagnostic 2MHz probes- a pilot study", Biomed Pap Med Fac, 2012; 156; DOI: 10.5507/BP.2012.064.

[49] Rubiera et al., "Safety and Efficacy of Ultrasound-Enhanced Thrombolysis: A Comprehensive Review and Meta-Analysis of Randomized and Nonrandomized Studies", Stroke, 2010;41; 280-287.

[50] Alexandrov et al., "A Pilot Randomized Clinical Safety Study of Sonothrombolysis Augumentation With Ultrasound-Activated Perflutre-Lipid Microspheres for Acute Ischemic Stroke", Stroke, 2008;39(5); 1464-1469.

[51] Parikh et al., "Ultrasound-accelerated Thrombolysis for the treatment of Deep Vein Thrombosis: Initial Clinical Experience", J Vasc Interv Radiol, 2008; 19; 521-528.

[52] Moskovitz, "Ultrasound Accelerated Thrombolysis: A Safe and Effective, Low-dose Thrombolytic Therapy for Acute Pulmonary Embolism", Vein, 2012; issue 3 volume 5.

[53] Malin Larsson et al., "Evaluation of the acoustic response of novel polymer-shelled contrast agents using clinical ultrasound systems" , Submitted to Ultrasonics, 2013.

[54] Devcic-Kuhar, "In vitro thrombolysis enhanced by standing and travelling ultrasound wave fields" Ultrasound in Med. & Biol., 2002;Vol. 28, No. 9, 1181–1187.

[55] Xie et al., "Effect of Attenuationc and Thrombus Age on the Success of Ultrasound and Microbubble Mediated Thrombus Dissolution", Ultrasound Med. Biol.,2011, 37(2), 280-288;

[56] Melanie Pretzl and Barbara Cerroni, "Standard protocol: concentration of MB/ml", 3MiCRON project, 2010.

Appendix

```matlab
1  % File name SetUpL12_5_50mmNoImaging1.m:
2  % code for Protocol 1
3  % frequency: 11.25MHz
4  % pulse length: 5ms
5  % max voltage: 100V
6  % time between trains of long pulses= 6s
7
8  clear all
9
10 % Specify system parameters.
11 Resource.Parameters.numTransmit = 128;      % number of transmit channels.
12 Resource.Parameters.numRcvChannels = 128; % number of receive channels.
13 Resource.Parameters.speedOfSound = 1540;    % speed of sound in m/sec
14 Resource.Parameters.simulateMode = 0; % 1=simulation; 0=run with hardware
15 Resource.Parameters.fakeScanhead=1;  % 1= run without probe
16
17 % Specify Trans structure array.
18 Trans.name = 'L12-5 50mm';        % L12-5 transducer is 'known'
19 Trans.frequency = 9.0;            % center frequency
20 Trans = computeTrans(Trans);     % L12-5 transducer is 'known'
21 Trans.maxHighVoltage = 50;       % set a high voltage limit.
22 Trans.impedance  = 45-60i;        % set transducer impedance.
23
24 % Specify Media object. 'pt1.m' script defines array of point targets.
25 pt1;
26 Media.function = 'movePoints';
27
28 % Specify Resource.
29 Resource.RcvBuffer(1).datatype = 'int16'; %  bit size of the sample
30 Resource.RcvBuffer(1).rowsPerFrame = 4096; %  size for acquisitions range
31 Resource.RcvBuffer(1).colsPerFrame = Resource.Parameters.numRcvChannels;
32 Resource.RcvBuffer(1).numFrames = 1;       % number of frames (not used)
33
34 % Specify TW structure array.
35 TW(1).type = 'parametric';   % waveform type
36 TW(1).Parameters = [8,3,31,1];   % A=11.25 MHz, B =37%, C =1.4us, D= 1
37 TW(1).extendBL = 1;     % if equal to 1, C value multiplied by 64 thus C=88.18us
38
39 % Specify TX structure array.
40 TX(1).waveform=1;               % same waveform for all the elements
41 TX(1).Origin =[0.0,0.0,0.0]; % beam origin
42 TX(1).focus =60 ;             %focus at 10 mm
43 TX(1).aperture = 65;         % tx aperture starts at element 65.
44 TX(1).Apod  =[zeros(1,36)  ones(1,56) zeros(1,36) ];% tx from 56 elem
```

```
45   TX(1).Delay= computeTXDelays(TX(1));       % delay for each active transmitter
46
47   % Specify Profile 5.
48   TPC(5).name = 'long pulse';   % name of the profile
49   TPC(5).maxHighVoltage = 50; % max voltage for the profile
50   TPC(5).highVoltageLimit =50;  % high voltage limit based on used model
51
52   % Specify TGC Waveform structure.
53   TGC.CntrlPts = [500,590,650,710,770,830,890,950]; % values already given
54   TGC.rangeMax = 185;                                        % max depth of the range
55   TGC.Waveform = computeTGCWaveform(TGC);       % use function already built in
56
57   % Specify Receive structure (not used)
58   Receive.Apod= zeros(1,Resource.Parameters.numRcvChannels);
59   Receive.aperture=1;
60   Receive.startDepth=2;
61   Receive.endDepth=185;
62   Receive.TGC=1;
63   Receive.mode=0;
64   Receive.bufnum=1;
65   Receive.framenum=1;
66   Receive.acqNum=1;
67   Receive.samplesPerWave=4;
68   Receive.InputFilter=[0.0036,0.0127,0.0066,-0.0881,-0.2595,0.6494];
69
70   % Specify SeqControl structure arrays.
71
72   % jump back to start.
73   SeqControl(1).command = 'jump';
74   SeqControl(1).argument = 2;
75   % Change to Profile 5 (long pulse)
76   SeqControl(2).command = 'setTPCProfile';
77   SeqControl(2).condition = 'immediate';
78   SeqControl(2).argument = 5;
79   % Set to allow time for profile change.
80   SeqControl(3).command = 'noop';
81   SeqControl(3).argument = 8000;% time in us
82   % Set time between long pulses.
83   SeqControl(4).command = 'timeToNextAcq';
84   SeqControl(4).argument = 124;  % time in us
85   % Set 4s idle time (2s more with SeqControl(7))
86   SeqControl(5).command = 'timeToNextAcq';
87   SeqControl(5).argument = 4000000; % time in us
88   % Set return to MatLab
89   SeqControl(6).command = 'returnToMatlab';
90   % Set 0.5s idle time (used for having 2s more after SeqControl(5))
91   SeqControl(7).command = 'noop';
92   SeqControl(7).argument = 500000 ; % time in us
93
94   % nsc is count of SeqControl objects
95   nsc = 8;
96
97   % n is count of Events
98   n = 1;
99
100  % Event for setting the profile 5
101  Event(n).info = 'set TPCProfile';
102  Event(n).tx = 0;          % no TX
```

```matlab
103  Event(n).rcv = 0;          % no Rcv
104  Event(n).recon = 0;        % no Recon
105  Event(n).process = 0;    % no Processing
106  Event(n).seqControl = [2,3]; % SeqControl(2) and SeqControl(3)
107  n=n+1;
108  % 40 Events for 5ms train of pulses
109  for i = 1: 40
110      Event(n).info = 'long pulse';
111      Event(n).tx = 1;           % use 1st TX structure.
112      Event(n).rcv = 0;          % no Rcv
113      Event(n).recon = 0;        % no reconstruction.
114      Event(n).process = 0;      % no processing
115      Event(n).seqControl = 4; % time between pulses
116      n = n+1;
117  end
118  % Idle time 4+2=6s and Return to Matlab
119   Event(n-1).seqControl = [5,6];
120  for t=1:4
121          Event(n).info='wait';
122          Event(n).tx = 0;            % no TX
123          Event(n).rcv = 0;          % no Rcv
124          Event(n).recon = 0;        % no Recon
125          Event(n).process = 0;    % no processing
126          Event(n).seqControl = 7; %  idle time
127          n=n+1;
128  end
129  % repeat sequence of events from second event
130  Event(n).info = 'Jump back to second event';
131  Event(n).tx = 0;          % no TX
132  Event(n).rcv = 0;         % no Rcv
133  Event(n).recon = 0;       % no Recon
134  Event(n).process = 0;     % no processing
135  Event(n).seqControl = 1;% jump command
136  n=n+1;
137  % sham events for VDAS running
138  Event(n).info = 'sham event';
139  Event(n).tx = 1;          % no TX
140  Event(n).rcv = 1;         % no Rcv
141  Event(n).recon = 0;       % no Recon
142  Event(n).process = 0;     % no processing
143  Event(n).seqControl = 0;
144  n=n+1;
145  Event(n).info = 'sham event';
146  Event(n).tx = 0;          % no TX
147  Event(n).rcv = 0;         % no Rcv
148  Event(n).recon = 0;       % no Recon
149  Event(n).process = 0;     % no processing
150  Event(n).seqControl = nsc;
151  SeqControl(nsc).command = 'transferToHost';
152      nsc=nsc+1;
153
154  % Save all the structures to a .mat file.
155  save('L12-5_50mmNoImaging1');
```

```
1  % File name SetUpL7_4NoImaging2_3.m:
2  % code for Protocol 2 and for Protocol 3
3  % frequency: 4.09MHz
4  % pulse length: 5ms
5  % max voltage: 100V
6  % time between trains of long pulses= 6s
7
8  clear all
9
10 % Specify system parameters.
11 Resource.Parameters.numTransmit = 128;     % number of transmit channels.
12 Resource.Parameters.numRcvChannels = 128; % number of receive channels.
13 Resource.Parameters.speedOfSound = 1540;   % speed of sound in m/sec
14 Resource.Parameters.simulateMode = 0; % 1=simulation; 0=run with hardware
15 Resource.Parameters.fakeScanhead=1;  % 1= run without probe
16
17 % Specify Trans structure array.
18 Trans.name = 'L7-4';                % L7-4 transducer is 'known'
19 Trans = computeTrans(Trans);    % L7-4 transducer is 'known'
20 Trans.maxHighVoltage = 50;      % set a high voltage limit.
21
22
23 % only for simulation
24 % Specify Media object. 'pt1.m' script defines array of point targets.
25 pt1;
26 Media.function = 'movePoints';
27
28 % Specify Resource.
29 Resource.RcvBuffer(1).datatype = 'int16'; %  bit size of the sample
30 Resource.RcvBuffer(1).rowsPerFrame = 4096; %  size for acquisitions range
31 Resource.RcvBuffer(1).colsPerFrame = Resource.Parameters.numRcvChannels;
32 Resource.RcvBuffer(1).numFrames = 1;       % number of frames (not used)
33
34 % Specify TW structure array.
35 TW(1).type = 'parametric';   % waveform type
36 TW(1).Parameters = [22,8,31,1];    % A=4.09 MHz, B =37%, C =3.7us, D= 1
37 TW(1).extendBL = 1;      % if equal to 1, C value multiplied by 64 thus C=240us
38
39 % Specify TX structure array.
40 TX(1).waveform=1;              % same waveform for all the elements
41 TX(1).Origin=[0.0,0.0,0.0]; % beam origin
42 TX(1).focus =85 ;            %focus at 25 mm
43 TX(1).Apod  =[zeros(1,44)   ones(1,40)  zeros(1,44)  ];% tx from 40 elem
44 TX(1).Delay= computeTXDelays(TX(1));    % delay for each active transmitter
45
46 % Specify Profile 5.
47 TPC(5).name = 'long pulse';  % name of the profile
48 TPC(5).maxHighVoltage = 50; % max voltage for the profile
49 TPC(5).highVoltageLimit =50;  % high voltage limit based on used model
50
51 % Specify TGC Waveform structure.
52 TGC.CntrlPts = [500,590,650,710,770,830,890,950]; % values already given
53 TGC.rangeMax = 185;                                % max depth of the range
54 TGC.Waveform = computeTGCWaveform(TGC);    % use function already built in
55
56 % Specify Receive structure (not used)
57 Receive = repmat(struct(...
```

```
58            'Apod', zeros(1,Resource.Parameters.numRcvChannels), ...
59            'aperture', 1, ...
60            'startDepth', 2, ...
61            'endDepth', 185, ...
62            'TGC', 1, ...
63            'mode', 0, ...
64            'bufnum', 1, ...
65            'framenum', 1, ...
66            'acqNum', 1, ...
67            'samplesPerWave', 4, ...                    'InputFilter'
                 ,[0.0036,0.0127,0.0066,-0.0881,-0.2595,0.6494]) ,...
68                     1,Resource.RcvBuffer(1).numFrames);
69
70 % Specify SeqControl structure arrays.
71
72 % jump back to start.
73 SeqControl(1).command = 'jump';
74 SeqControl(1).argument = 2;
75 % Change to Profile 5 (long pulse)
76 SeqControl(2).command = 'setTPCProfile';
77 SeqControl(2).condition = 'immediate';
78 SeqControl(2).argument = 5;
79 % Set to allow time for profile change.
80 SeqControl(3).command = 'noop';
81 SeqControl(3).argument = 8000;% time in us
82 % Set time between long pulses.
83 SeqControl(4).command = 'timeToNextAcq';
84 SeqControl(4).argument = 124;  % time in us
85 % Set 4s idle time (2s more with SeqControl(7))
86 SeqControl(5).command = 'timeToNextAcq';
87 SeqControl(5).argument = 4000000; % time in us
88 % Set return to MatLab
89 SeqControl(6).command = 'returnToMatlab';
90 % Set 0.5s idle time (used for having 2s more after SeqControl(5))
91 SeqControl(7).command = 'noop';
92 SeqControl(7).argument = 500000 ; % time in us
93
94 % nsc is count of SeqControl objects
95 nsc = 8;
96
97 % n is count of Events
98 n = 1;
99
100 % Event for setting the profile 5
101 Event(n).info = 'set TPCProfile';
102 Event(n).tx = 0;         % no TX
103 Event(n).rcv = 0;        % no Rcv
104 Event(n).recon = 0;      % no Recon
105 Event(n).process = 0;  % no Processing
106 Event(n).seqControl = [2 ,3]; % SeqControl(2) and SeqControl(3)
107 n=n+1;
108 % 20 Events for 5ms train of pulses
109 for i = 1: 20
110     Event(n).info = 'long pulse';
111     Event(n).tx = 1;          % use 1st TX structure.
112     Event(n).rcv = 0;         % no Rcv
113     Event(n).recon = 0;       % no reconstruction.
114     Event(n).process = 0;     % no processing
```

```matlab
115          Event(n).seqControl = 4; % time between pulses
116          n = n+1;
117  end
118  % Idle time 4+2=6s and Return to Matlab
119   Event(n-1).seqControl = [5,6];
120  for t=1:4
121            Event(n).info='wait';
122            Event(n).tx = 0;           % no TX
123            Event(n).rcv = 0;          % no Rcv
124            Event(n).recon = 0;        % no Recon
125            Event(n).process = 0;      % no processing
126            Event(n).seqControl = 7; %  idle time
127            n=n+1;
128  end
129  % repeat sequence of events from second event
130  Event(n).info = 'Jump back to second event';
131  Event(n).tx = 0;           % no TX
132  Event(n).rcv = 0;          % no Rcv
133  Event(n).recon = 0;        % no Recon
134  Event(n).process = 0;      % no processing
135  Event(n).seqControl = 1;% jump command
136  n=n+1;
137  % sham events for VDAS running
138  Event(n).info = 'sham event';
139  Event(n).tx = 1;           % no TX
140  Event(n).rcv = 1;          % no Rcv
141  Event(n).recon = 0;        % no Recon
142  Event(n).process = 0;      % no processing
143  Event(n).seqControl = 0;
144  n=n+1;
145  Event(n).info = 'sham event';
146  Event(n).tx = 0;           % no TX
147  Event(n).rcv = 0;          % no Rcv
148  Event(n).recon = 0;        % no Recon
149  Event(n).process = 0;      % no processing
150  Event(n).seqControl = nsc;
151  SeqControl(nsc).command = 'transferToHost';
152       nsc=nsc+1;
153
154  % Save all the structures to a .mat file.
155  save('L7_4NoImaging2_3');
```

```
1  % File name SetUpL7_4NoImaging4.m:
2  % code for Protocol 4
3  % frequency: 4.09MHz
4  % pulse length: 10ms
5  % max voltage: 50V
6  % time between trains of long pulses= 6s
7
8  clear all
9
10 % Specify system parameters.
11 Resource.Parameters.numTransmit = 128;      % number of transmit channels.
12 Resource.Parameters.numRcvChannels = 128; % number of receive channels.
13 Resource.Parameters.speedOfSound = 1540;    % speed of sound in m/sec
14 Resource.Parameters.simulateMode = 0; % 1=simulation; 0=run with hardware
15 Resource.Parameters.fakeScanhead=1;  % 1= run without probe
16
17 % Specify Trans structure array.
18 Trans.name = 'L7-4';               % L7-4 transducer is 'known'
19 Trans = computeTrans(Trans);    % L7-4 transducer is 'known'
20 Trans.maxHighVoltage = 25;      % set a high voltage limit.
21
22
23 % only for simulation
24 % Specify Media object. 'pt1.m' script defines array of point targets.
25 pt1;
26 Media.function = 'movePoints';
27
28 % Specify Resource.
29 Resource.RcvBuffer(1).datatype = 'int16'; %  bit size of the sample
30 Resource.RcvBuffer(1).rowsPerFrame = 4096; %  size for acquisitions range
31 Resource.RcvBuffer(1).colsPerFrame = Resource.Parameters.numRcvChannels;
32 Resource.RcvBuffer(1).numFrames = 1;        % number of frames (not used)
33
34 % Specify TW structure array.
35 TW(1).type = 'parametric';   % waveform type
36 TW(1).Parameters = [22,8,31,1];    % A=4.09 MHz, B =37%, C =3.7us, D= 1
37 TW(1).extendBL = 1;      % if equal to 1, C value multiplied by 64 thus C=240us
38
39 % Specify TX structure array.
40 TX(1).waveform=1;              % same waveform for all the elements
41 TX(1).Origin=[0.0,0.0,0.0]; % beam origin
42 TX(1).focus =85 ;              %focus at 25 mm
43 TX(1).Apod  =[zeros(1,44)  ones(1,40) zeros(1,44) ];% tx from 40 elem
44 TX(1).Delay= computeTXDelays(TX(1));     % delay for each active transmitter
45
46 % Specify Profile 5.
47 TPC(5).name = 'long pulse';  % name of the profile
48 TPC(5).maxHighVoltage = 25; % max voltage for the profile
49 TPC(5).highVoltageLimit =25;  % high voltage limit based on used model
50
51 % Specify TGC Waveform structure.
52 TGC.CntrlPts = [500,590,650,710,770,830,890,950]; % values already given
53 TGC.rangeMax = 185;                                       % max depth of the range
54 TGC.Waveform = computeTGCWaveform(TGC);     % use function already built in
55
56 % Specify Receive structure (not used)
57 Receive = repmat(struct(...
```

```
58          'Apod', zeros(1,Resource.Parameters.numRcvChannels), ...
59          'aperture', 1, ...
60          'startDepth', 2, ...
61          'endDepth', 185, ...
62          'TGC', 1, ...
63          'mode', 0, ...
64          'bufnum', 1, ...
65          'framenum', 1, ...
66          'acqNum', 1, ...
67          'samplesPerWave', 4, ...                          'InputFilter'
             ,[0.0036,0.0127,0.0066,-0.0881,-0.2595,0.6494]) ,...
68                        1,Resource.RcvBuffer(1).numFrames);
69
70  % Specify SeqControl structure arrays.
71
72  % jump back to start.
73  SeqControl(1).command = 'jump';
74  SeqControl(1).argument = 2;
75  % Change to Profile 5 (long pulse)
76  SeqControl(2).command = 'setTPCProfile';
77  SeqControl(2).condition = 'immediate';
78  SeqControl(2).argument = 5;
79  % Set to allow time for profile change.
80  SeqControl(3).command = 'noop';
81  SeqControl(3).argument = 8000;% time in us
82  % Set time between long pulses.
83  SeqControl(4).command = 'timeToNextAcq';
84  SeqControl(4).argument = 124;  % time in us
85  % Set 4s idle time (2s more with SeqControl(7))
86  SeqControl(5).command = 'timeToNextAcq';
87  SeqControl(5).argument = 4000000; % time in us
88  % Set return to MatLab
89  SeqControl(6).command = 'returnToMatlab';
90  % Set 0.5s idle time (used for having 2s more after SeqControl(5))
91  SeqControl(7).command = 'noop';
92  SeqControl(7).argument = 500000 ; % time in us
93
94  % nsc is count of SeqControl objects
95  nsc = 8;
96
97  % n is count of Events
98  n = 1;
99
100 % Event for setting the profile 5
101 Event(n).info = 'set TPCProfile';
102 Event(n).tx = 0;          % no TX
103 Event(n).rcv = 0;         % no Rcv
104 Event(n).recon = 0;       % no Recon
105 Event(n).process = 0;  % no Processing
106 Event(n).seqControl = [2,3]; % SeqControl(2) and SeqControl(3)
107 n=n+1;
108 % 40 Events for 10ms train of pulses
109 for i = 1: 40
110     Event(n).info = 'long pulse';
111     Event(n).tx = 1;          % use 1st TX structure.
112     Event(n).rcv = 0;         % no Rcv
113     Event(n).recon = 0;       % no reconstruction.
114     Event(n).process = 0;      % no processing
```

```matlab
115        Event(n).seqControl = 4; % time between pulses
116        n = n+1;
117 end
118 % Idle time 4+2=6s and Return to Matlab
119  Event(n-1).seqControl = [5,6];
120 for t=1:4
121         Event(n).info='wait';
122         Event(n).tx = 0;          % no TX
123         Event(n).rcv = 0;         % no Rcv
124         Event(n).recon = 0;       % no Recon
125         Event(n).process = 0;    % no processing
126         Event(n).seqControl = 7; %  idle time
127         n=n+1;
128 end
129 % repeat sequence of events from second event
130 Event(n).info = 'Jump back to second event';
131 Event(n).tx = 0;          % no TX
132 Event(n).rcv = 0;         % no Rcv
133 Event(n).recon = 0;       % no Recon
134 Event(n).process = 0;    % no processing
135 Event(n).seqControl = 1;% jump command
136 n=n+1;
137 % sham events for VDAS running
138 Event(n).info = 'sham event';
139 Event(n).tx = 1;          % no TX
140 Event(n).rcv = 1;         % no Rcv
141 Event(n).recon = 0;       % no Recon
142 Event(n).process = 0;    % no processing
143 Event(n).seqControl = 0;
144 n=n+1;
145 Event(n).info = 'sham event';
146 Event(n).tx = 0;          % no TX
147 Event(n).rcv = 0;         % no Rcv
148 Event(n).recon = 0;       % no Recon
149 Event(n).process = 0;    % no processing
150 Event(n).seqControl = nsc;
151 SeqControl(nsc).command = 'transferToHost';
152     nsc=nsc+1;
153
154 % Save all the structures to a .mat file.
155 save('L7_4NoImaging4');
```

```matlab
% File name SetUpL12_5_50mmImaging.m:
% code for sending long pulses with imaging in between
% adapted from the Verasonic script SetUpL12_5_50mmFlash_4B.m
% LONG PULSES:
% - frequency: 11.25MHz
% - voltage: 100V
% - pulse length: 88.18us / idle time: 35.82us
% - train of pulses length: 5ms / idle time: 6s
% IMAGING:
% - frequency: 11.25MHz
% - voltage: 3V
% - pulse length 0.08us
% - three apertures
% 128 transmit channels are used for each of 3 transmit apertures. On receive, the
% 128 channels are positioned as follows (each char represents 4 elements):
% IMAGING
%
% Element Nos.                                     1       1   1                   2
%                                       6       8  2       7   9                   5
%                          1            5       6  9       2   3                   6
% Aperture 1: |                    |       |       |           |   |               |
%                 tttttttttttttttttttttttttttttttt ————————————————————————————————
%                 rrrrrrrrrrrrrrrrrrrrr ———————————————————————————————————————————
%                                     |       |       |           |   |               |
% Aperture 2: |                    |       |       |           |   |               |
%                 ————————————————————————tttttttttttttttttttttttttttttttt —————————————
%                 ————————————————————————rrrrrrrrrrrrrrrrrrrrr ————————————
%                                     |       |       |           |   |               |
% Aperture 3: |                    |       |       |           |   |               |
%                 ————————————————————————————————tttttttttttttttttttttttttttttttt
%                 ————————————————————————————————rrrrrrrrrrrrrrrrrrrrr
%                                     |       |       |           |   |               |
%
% The receive data from each of these apertures are stored under different acqNums in the
% Receive buffer. The reconstruction sums the IQ data from the 3 aquisitions and computes
% intensity values to produce the full frame.
%
% This version does asynchronous acquisition and processing.
%PULSE TRANSMISSION
% Aperture 1: |                    |       |       |           |   |               |
%                 ————————————————————tttttttttttttttttttttttttttttttt ————————————————


clear all

% Specify system parameters.
Resource.Parameters.numTransmit = 128;       % number of transmit channels.
Resource.Parameters.numRcvChannels = 128;    % number of receive channels.
Resource.Parameters.speedOfSound = 1540;     % set speed of sound in m/sec
Resource.Parameters.simulateMode = 0;
Resource.Parameters.fakeScanhead=1;

% Specify Trans structure array.
Trans.name = 'L12-5 50mm';
Trans.frequency = 9.0;
```

```
56  Trans = computeTrans(Trans);  % L12-5 transducer is 'known'
57  Trans.maxHighVoltage = 50;
58  Trans.impedance  = 45-60i;
59
60  % Specify SFormat structure array.
61  SFormat.transducer = 'L12-5 50mm';
62  SFormat.scanFormat = 'RLIN';     % rectangular linear array scan
63  SFormat.radius = 0;
64  SFormat.theta = 0;
65  SFormat.numRays = 1;        %  Flat Focus
66  SFormat.FirstRayLoc = [0,0,0];    % x,y,z
67  SFormat.rayDelta = 256*Trans.spacing;  % spacing in dist. between rays
68  SFormat.startDepth = 2;   % Acquisition start depth in wavelength
69  SFormat.endDepth = 185 ;  % Acquisition end depth in wavelength
70
71  % Specify PData structure array.
72  PData.sFormat = 1;         % use first SFormat structure.
73  PData.pdeltaX = 1.0;
74  PData.pdeltaZ = 0.5;
75  PData.Size(1) = ceil((SFormat.endDepth-SFormat.startDepth)/PData.pdeltaZ);
76  PData.Size(2) = ceil((Trans.numelements*Trans.spacing)/PData.pdeltaX);
77  PData.Size(3) = 1;         % single image page
78  PData.Origin = [-Trans.spacing*(Trans.numelements-1)/2,0,SFormat.startDepth];
79
80  % only for simulation
81  % Specify Media object. 'pt1.m' script defines array of point targets.
82  pt1;
83  Media.function = 'movePoints';
84
85  % Specify Resources.
86  Resource.RcvBuffer(1).datatype = 'int16';
87  Resource.RcvBuffer(1).rowsPerFrame = 4096*3; % this size allows for 3 acqs,
        maximum range
88  Resource.RcvBuffer(1).colsPerFrame = Resource.Parameters.numRcvChannels;
89  Resource.RcvBuffer(1).numFrames = 245;        % allows 5s of imaging
90  Resource.InterBuffer(1).datatype = 'complex';
91  Resource.InterBuffer(1).rowsPerFrame = 1024; % this is for greatest depth
92  Resource.InterBuffer(1).colsPerFrame = PData.Size(2);
93  Resource.InterBuffer(1).numFrames = 1;  % one intermediate buffer needed.
94  Resource.ImageBuffer(1).datatype = 'double';
95  Resource.ImageBuffer(1).rowsPerFrame = 1024;
96  Resource.ImageBuffer(1).colsPerFrame = PData.Size(2);
97  Resource.ImageBuffer(1).numFrames = 10;
98  Resource.DisplayWindow(1).Title = 'Image Display';
99  Resource.DisplayWindow(1).pdelta = 0.45;
100 Resource.DisplayWindow(1).Position = [250,250, ...    % upper left corner position
101        ceil(PData.Size(2)*PData.pdeltaX/Resource.DisplayWindow(1).pdelta), ... %
            width
102     ceil(PData.Size(1)*PData.pdeltaZ/Resource.DisplayWindow(1).pdelta)];     %
            height
103 Resource.DisplayWindow(1).ReferencePt = [PData.Origin(1),PData.Origin(3)];   % 2D
      imaging  (X,Z plane)
104 Resource.DisplayWindow(1).Colormap = gray(256);
105
106 % Specify Transmit waveform structure.
107 % for imaging
108 TW(1).type = 'parametric';
109 TW(1).Parameters = [8,4,2,1];   % A=11.25MHz, B =50%, C =0.08us, D= 1
```

```matlab
110  % for long pulses
111  TW(2).type = 'parametric';
112  TW(2).Parameters = [8,3,31,1];     % A=11.25MHz, B =37%, C =88.18us, D= 1
113  TW(2).extendBL = 1;
114
115  % Specify TX structure array.
116  % imaging
117  TX = repmat(struct('waveform', 1, ...
118                  'Origin', [0.0,0.0,0.0], ...
119                  'aperture', 1, ...
120                          'Apod', ones(1,Resource.Parameters.numTransmit), ...
121                          'focus', 0.0, ...
122                          'Steer', [0.0,0.0], ...
123                          'Delay', zeros(1,Resource.Parameters.numTransmit)), 1, 4);
124  TX(2).aperture = 65;  % Use the tx aperture that starts at element 65.
125  TX(3).aperture = 129; % Use the tx aperture that starts at element 129.
126  % for long pulses
127  TX(4).waveform = 2;
128  TX(4).aperture = 65;
129  TX(4).Apod  =[zeros(1,36)  ones(1,56) zeros(1,36) ];
130  TX(4).focus =60 ; %focus at 10mm
131  TX(4).Delay= computeTXDelays(TX(4));
132
133  % Specify TGC Waveform structure.
134  TGC.CntrlPts = [500,590,650,710,770,830,890,950];
135  TGC.rangeMax = SFormat.endDepth;
136  TGC.Waveform = computeTGCWaveform(TGC);
137
138  % Specify TPC structures.
139  % profile 1 for imaging
140  TPC(1).name = '2D';
141  TPC(1).maxHighVoltage = 50;
142  TPC(1).highVoltageLimit = 50;
143  % profile 5 for long pulses
144  TPC(5).name = 'long pulse';
145  TPC(5).maxHighVoltage = 50;
146  TPC(5).highVoltageLimit =50;
147
148  % Specify Receive structure arrays −
149  %   endDepth − add additional acquisition depth to account for some channels
150  %              having longer path lengths.
151  %   InputFilter − The same coefficients are used for all channels. The
152  %                 coefficients below give a broad bandwidth bandpass filter.
153  maxAcqLength = sqrt(SFormat.endDepth^2 + (Trans.numelements*Trans.spacing)^2) −
         SFormat.startDepth;
154  wlsPer128 = 128/(4*2); % wavelengths in 128 samples for 4 samplesPerWave
155  Receive = repmat(struct('Apod', zeros(1,Resource.Parameters.numRcvChannels), ...
156                          'aperture', 1, ...
157                          'startDepth', SFormat.startDepth, ...
158                          'endDepth', SFormat.startDepth + wlsPer128*ceil(
                               maxAcqLength/wlsPer128), ...
159                          'TGC', 1, ...
160                          'bufnum', 1, ...
161                          'framenum', 1, ...
162                          'acqNum', 1, ...
163                          'samplesPerWave', 4, ...
164                          'mode', 0, ...
165                          'InputFilter'
```

```
                                      ,[0.0036,0.0127,0.0066,−0.0881,−0.2595,0.6494], ...
166                        'callMediaFunc', 0),1,3*(Resource.RcvBuffer(1).numFrames));
167
168 % − Set event specific Receive attributes.
169 for i = 1:Resource.RcvBuffer(1).numFrames  % 3 acquisitions per frame
170   % −− 1st synthetic aperture acquisition for aperture 1.
171    Receive(3*i−2).Apod(1:85) = 1.0;
172    Receive(3*i−2).aperture = 1;
173    Receive(3*i−2).framenum = i;
174    Receive(3*i−2).acqNum = 1;
175    Receive(3*i−2).callMediaFunc = 1;
176   % −− 2nd synthetic aperture acquisition for aperture 65.
177    Receive(3*i−1).Apod(22:107) = 1.0;
178    Receive(3*i−1).aperture = 65;
179    Receive(3*i−1).framenum = i;
180    Receive(3*i−1).acqNum = 2;
181   % −− 3rd synthetic aperture acquisition for aperture 129.
182    Receive(3*i).Apod(44:128) = 1.0;
183    Receive(3*i).aperture = 129;
184    Receive(3*i).framenum = i;
185    Receive(3*i).acqNum = 3;
186 end
187
188 % Specify Recon structure arrays.
189 Recon = struct('senscutoff', 0.6, ...
190           'pdatanum', 1, ...
191                 'rcvBufFrame', −1, ...      % use most recently transferred frame
192                 'IntBufDest', [1,1], ...
193                 'ImgBufDest', [1,−1], ...  % auto−increment ImageBuffer each recon
194                 'RINums', [1;2;3]);
195
196 % Define ReconInfo structures.
197 ReconInfo = repmat(struct('mode', 3, ...  % replace IQ data.
198                           'txnum', 1, ...
199                           'rcvnum', 1, ...
200                           'regionnum', 0), 1, 3);
201 % − Set specific ReconInfo attributes.
202 ReconInfo(2).mode = 4; % accumulate and detect IQ data in output buffer.
203 ReconInfo(2).txnum = 2;
204 ReconInfo(2).rcvnum = 2;
205 ReconInfo(3).mode = 5; % accumulate and detect IQ data in output buffer.
206 ReconInfo(3).txnum = 3;
207 ReconInfo(3).rcvnum = 3;
208
209 % Specify Process structure array.
210 pers = 30;
211 Process(1).classname = 'Image';
212 Process(1).method = 'imageDisplay';
213 Process(1).Parameters = {'imgbufnum',1,...   % number of buffer to process.
214                          'framenum',−1,...   % (−1 = lastFrame)
215                          'pdatanum',1,...    % number of PData structure to use
216                          'norm',1,...        % normalization method(1 means fixed)
217                          'pgain',1.0,...     % pgain is image processing gain
218                          'persistMethod','simple',...
219                          'persistLevel',pers,...
220                          'interp',1,... % method of interpolation (1=4pt interp)
221                          'compression',0.5,...% X^0.5 normalized to output word
                               size
```

```matlab
222                               'reject',2,...
223                               'mappingMode','full',...
224                               'display',1,...        % display image after processing
225                               'displayWindow',1};
226
227  % Specify SeqControl structure arrays.
228
229  % jump back to start
230  SeqControl(1).command = 'jump';
231  SeqControl(1).argument = 1;
232  % set profile 1
233  SeqControl(2).command = 'setTPCProfile';
234  SeqControl(2).condition = 'immediate';
235  SeqControl(2).argument = 1;
236  % set profile 5
237  SeqControl(3).command = 'setTPCProfile';
238  SeqControl(3).condition = 'immediate';
239  SeqControl(3).argument = 5;
240  % allow time for switchin to profile 5
241  SeqControl(4).command = 'noop';
242  SeqControl(4).argument = 8000; % time in us
243  % time between long pulses
244  SeqControl(5).command = 'timeToNextAcq';
245  SeqControl(5).argument = 124; % time in usec
246  % 1s  not imaging and not transmitting long pulse
247  SeqControl(6).command =  'timeToNextAcq';
248  SeqControl(6).argument = 1000000;  % 1s
249  % return to MatLab
250  SeqControl(7).command = 'returnToMatlab';
251  % time between apertures for imaging
252  SeqControl(8).command = 'timeToNextAcq';
253  SeqControl(8).argument = 200;  % 200 usec
254  % time between frames for imaging
255  SeqControl(9).command = 'timeToNextAcq';
256  SeqControl(9).argument = 20000;  % 20ms
257
258  % nsc is count of SeqControl objects
259  nsc = 10;
260
261  % n is count of Events
262  n = 1;
263
264  Event(n).info = 'set TPCProfile';
265  Event(n).tx = 0;         % no TX
266  Event(n).rcv = 0;        % no Rcv
267  Event(n).recon = 0;      % no Recon
268  Event(n).process = 0;
269  Event(n).seqControl = [2];
270  n=n+1;
271  % Acquire all frames defined in RcvBuffer
272  for i = 1:Resource.RcvBuffer(1).numFrames
273     if i==Resource.RcvBuffer(1).numFrames
274        Event(n).info = 'set TPCProfile';
275        Event(n).tx=0;            % no tx
276        Event(n).rcv=0;           % no rcv
277        Event(n).recon = 0;      % no reconstruction.
278        Event(n).process = 0;    % no processing
279        Event(n).seqControl=[3,4];
```

```matlab
280        n=n+1;
281        for t= 1:40
282          Event(n).info='transmit long pulse';
283          Event(n).tx=4;
284          Event(n).rcv=0;           % no rcv
285          Event(n).recon = 0;       % no reconstruction.
286          Event(n).process = 0;     % no processing
287          Event(n).seqControl = 5;
288          n=n+1;
289        end
290        Event(n-1).seqControl = 6;
291        Event(n).info='return to Matlab';
292        Event(n).tx=0;
293        Event(n).rcv=0;
294        Event(n).recon = 0;       % no reconstruction.
295        Event(n).process = 0;     % no processing
296        Event(n).seqControl = 7;
297        n=n+1;
298      else
299        Event(n).info = '1st 3rd of aperture.';
300        Event(n).tx = 1;           % use 1st TX structure.
301        Event(n).rcv = 3*i-2;      % use 1st Rcv structure.
302        Event(n).recon = 0;        % no reconstruction.
303        Event(n).process = 0;      % no processing
304        Event(n).seqControl = 8; % time between syn. aper. acqs.
305        n = n+1;
306
307        Event(n).info = 'middle 3rd of aperture.';
308        Event(n).tx = 2;           % use 1st TX structure.
309        Event(n).rcv = 3*i-1;      % use 1st Rcv structure.
310        Event(n).recon = 0;        % no reconstruction.
311        Event(n).process = 0;      % no processing
312        Event(n).seqControl = 8; % time between syn. aper. acqs.
313        n = n+1;
314        Event(n).info = 'last 3rd of aperture.';
315        Event(n).tx = 3;           % use 3rd TX structure.
316        Event(n).rcv = 3*i;        % use 3rd Rcv structure.
317        Event(n).recon = 0;        % no reconstruction.
318        Event(n).process = 0;      % no processing
319        Event(n).seqControl = [9,nsc]; % use SeqControl structs defined below.
320        SeqControl(nsc).command = 'transferToHost';
321        nsc = nsc + 1;
322        n = n+1;
323
324        Event(n).info = 'Reconstruct';
325        Event(n).tx = 0;           % no transmit
326        Event(n).rcv = 0;          % no rcv
327        Event(n).recon = 1;        % reconstruction
328        Event(n).process = 1;      % processing
329        if floor(i/5) == i/5       % Exit to Matlab every 5th frame
330          Event(n).seqControl = 7; % return to Matlab
331        else
332          Event(n).seqControl = 0;
333        end
334        n = n+1;
335      end
336  end
337
```

```matlab
338
339
340  Event(n).info = 'Jump ';
341  Event(n).tx = 0;          % no TX
342  Event(n).rcv = 0;         % no Rcv
343  Event(n).recon = 0;       % no Recon
344  Event(n).process = 0;     % no process
345  Event(n).seqControl = 1;  % jump command
346  n=n+1;
347  Event(n).info = 'sham event';
348  Event(n).tx = 1;
349  Event(n).rcv = 1;
350  Event(n).recon = 0;       % no Recon
351  Event(n).process = 0;     % no process
352  Event(n).seqControl = 0;
353  n=n+1;
354  Event(n).info = 'sham event';
355  Event(n).tx = 0;          % no TX
356  Event(n).rcv = 0;         % no Rcv
357  Event(n).recon = 0;       % no Recon
358  Event(n).process = 0;     % no process
359  Event(n).seqControl = nsc;
360  SeqControl(nsc).command = 'transferToHost';
361  nsc=nsc+1;
362
363
364
365  % User specified UI Control Elements
366  % - Sensitivity Cutoff
367  sensx = 170;
368  sensy = 420;
369  UI(1).Control = {'Style','text',...          % popupmenu gives list of choices
370                   'String','Sens. Cutoff',...
371                   'Position',[sensx+10,sensy,100,20],... % position on UI
372                   'FontName','Arial','FontWeight','bold','FontSize',12,...
373                   'BackgroundColor',[0.8,0.8,0.8]};
374
375  UI(2).Control = {'Style','slider',...         % popupmenu gives list of choices
376                   'Position',[sensx,sensy-30,120,30],... % position on UI
377                   'Max',1.0,'Min',0,'Value',Recon(1).senscutoff,...
378                   'SliderStep',[0.025 0.1],...
379                   'Tag','sensSlider',...
380                   'Callback',{@sensCutoffCallback}};
381  UI(2).Callback = {'sensCutoffCallback.m',...
382                    'function sensCutoffCallback(hObject,eventdata)',...
383                    ' ',...
384                    'sens = get(hObject,''Value'');',...
385                    'ReconL = evalin(''base'', ''Recon'');',...
386                    'for i = 1:size(ReconL,2)',...
387                    '    ReconL(i).senscutoff = sens;',...
388                    'end',...
389                    'assignin(''base'',''Recon'',ReconL);',...
390                    '% Set Control.Command to re-initialize Recon structure.',...
391                    'Control = evalin(''base'',''Control'');',...
392                    'Control.Command = ''update&Run'';',...
393                    'Control.Parameters = {''Recon''};',...
394                    'assignin(''base'',''Control'', Control);',...
395                    '% Set the new cutoff value in the text field.',...
```

```
396                        'h = findobj(''tag'',''sensCutoffValue'');',...
397                        'set(h,''String'',num2str(sens,''%1.3f''));',...
398                        'return'};
399
400   UI(3).Control = {'Style','edit','String',num2str(Recon(1).senscutoff,'%1.3f'), ...
          %text
401                        'Position',[sensx+20,sensy-40,60,22],  ...    % position on UI
402                        'tag','sensCutoffValue', ...
403                        'BackgroundColor',[0.9,0.9,0.9],...
404                        'Callback',{@sensCutOffValueCallback}};
405
406   UI(3).Callback = {'sensCutOffValueCallback.m',...
407                        'function sensCutOffValueCallback(hObject,eventdata)',...
408                        '  ',...
409                        'sens = str2num(get(hObject,''String''));',...
410                        'ReconL = evalin(''base'',  ''Recon'');',...
411                        'for i = 1:size(ReconL,2)',...
412                        '    ReconL(i).senscutoff = sens;',...
413                        'end',...
414                        'assignin(''base'',''Recon'',ReconL);',...
415                        '% Set Control.Command to re-initialize Recon structure.',...
416                        'Control = evalin(''base'',''Control'');',...
417                        'Control.Command = ''update&Run'';',...
418                        'Control.Parameters = {''Recon''};',...
419                        'assignin(''base'',''Control'', Control);',...
420                        '% Set the new cutoff value for the slider',...
421                        'set(findobj(''Tag'',''sensSlider''),''Value'',sens);',...
422                        'return'};
423
424
425   % - Range Change
426   rngx = 20;
427   rngy = 125;
428   UI(4).Control = {'Style','text','String','Range',...
429                        'Position',[rngx+10,rngy,80,20],...
430                        'FontName','Arial','FontWeight','bold','FontSize',12,...
431                        'BackgroundColor',[0.8,0.8,0.8]};
432
433   UI(5).Control = {'Style','slider',...
434                'Position',[rngx,rngy-30,120,30],...
435                    'Max',320,'Min',64,'Value',SFormat.endDepth,...
436                        'SliderStep',[0.1,0.2],...
437                        'Callback',{@rangeChangeCallback}};
438
439   UI(5).Callback = {'rangeChangeCallback.m',...
440            'function rangeChangeCallback(hObject,eventdata)',...
441            '  ',...
442            'simMode = evalin(''base'',''Resource.Parameters.simulateMode'');',...
443            '% No range change if in simulate mode 2.',...
444            'if simMode == 2',...
445            'set(hObject,''Value'',evalin(''base'',''SFormat.endDepth''));',...
446            'return',...
447            'end',...
448            'range = get(hObject,''Value'');',...
449            'assignin(''base'',''range'',range);',...
450            'SFormat = evalin(''base'',''SFormat'');',...
451            'SFormat.endDepth = range;',...
452            'assignin(''base'',''SFormat'',SFormat);',...
```

```
453              'evalin(''base'',''PData.Size(1)=ceil((SFormat.endDepth-SFormat.startDepth
                    )/PData.pdeltaZ);'');',...
454              'evalin(''base'',''[PData.Region,PData.numRegions] = createRegions(PData);
                    '');',...
455              'evalin(''base'',''Resource.DisplayWindow(1).Position(4) =
456              ceil(PData.Size(1)*PData.pdeltaZ/Resource.DisplayWindow(1).pdelta);'');'
                    ,...
457              'Receive = evalin(''base'', ''Receive'');',...
458              'Trans = evalin(''base'', ''Trans'');',...
459              'maxAcqLength=sqrt(range^2+(Trans.numelements*Trans.spacing)^2)-SFormat.
                    startDepth;',...
460              'wlsPer128 = 128/(4*2);',...
461              'for i = 1:size(Receive,2)',...
462              ' Receive(i).endDepth=SFormat.startDepth+wlsPer128*ceil(maxAcqLength/
                    wlsPer128);',...                'end',...
463              'assignin(''base'',''Receive'',Receive);',...
464              '% Update VDAS parameters of Receive objects.',...
465              'evalin(''base'',''updateVDAS('''Receive''')'');',...
466              'evalin(''base'',''TGC.rangeMax = SFormat.endDepth;'');',...
467              'evalin(''base'',''TGC.Waveform = computeTGCWaveform(TGC);'');',...
468              'Control = evalin(''base'',''Control'');',...
469              'Control.Command = ''update&Run'';',...
470              'Control.Parameters =         {''SFormat'',''PData'',''Receive'',''Recon
                    '',''DisplayWindow'',''ImageBuffer''};',.            ..
471              'assignin(''base'',''Control'', Control);',...
472              'assignin(''base'', ''action'', ''displayChange'');',...
473              'h = findobj(''tag'',''rangeValue'');',...
474              'set(h,''String'',num2str(range,''%3.0f''));',...
475              'return'};
476
477
478  UI(6).Control = {'Style','edit','String',num2str(SFormat.endDepth,'%3.0f'), ...%
        text
479                       'Position',[rngx+20,rngy-40,60,22], ...    % position on UI
480                       'tag','rangeValue', ...
481                       'BackgroundColor',[0.9,0.9,0.9]};
482
483  clear i j n sensx sensy rngx rngy prsx prsy
484
485  % Specify factor for converting sequenceRate to frameRate.
486  frameRateFactor = 5;
487  % Save all the structures to a .mat file.
488  save('L12-5_50mmImaging');
```

```matlab
1  % File name SetUpL7_4Imaging.m:
2  % code for sending long pulses with imaging in between
3  % adapted from the Verasonic script SetUpL7_4FlashAngles.m
4  % LONG PULSES:
5  % − frequency: 4.09MHz
6  % − voltage: 100V
7  % − pulse length: 240us / idle time: 11us
8  % − train of pulses length: 5ms / idle time: 6s
9  % IMAGING:
10 % − frequency: 4.09MHz
11 % − voltage: 3V
12 % − pulse length 0.2us
13 % − seven angles
14
15  clear all
16
17 % Specify system parameters.
18  Resource.Parameters.numTransmit = 128;        % number of transmit channels.
19  Resource.Parameters.numRcvChannels = 128;    % number of receive channels.
20  Resource.Parameters.speedOfSound = 1540;
21  Resource.Parameters.simulateMode = 0;
22  Resource.Parameters.fakeScanhead=1;
23
24 % Set imaging parameters
25  na = 7;          % Set na = number of flash angles for 2D.
26  dtheta2D = (36*pi/180)/(na−1); % set dtheta2D to range over +/− 18 degrees.
27
28 % Specify Trans structure array.
29  Trans.name = 'L7−4';
30  Trans = computeTrans(Trans);   % known transducers.
31  Trans.maxHighVoltage = 50;
32
33 % Specify SFormat structure array.
34  SFormat.transducer = 'L7−4';
35  SFormat.scanFormat = 'RLIN';       % rectangular linear array scan
36  SFormat.radius = 0;
37  SFormat.theta = 0;
38  SFormat.numRays = 1;              % Flat Focus
39  SFormat.FirstRayLoc = [0,0,0];    % x,y,z
40  SFormat.rayDelta = 128*Trans.spacing;  % spacing dist. between rays (wvlnghts)
41  SFormat.startDepth = 0;              % Acquisition start depth in wavelengths
42  SFormat.endDepth = 150;              % Acquisition end depth in wavelengths
43
44
45 % Specify PData structure array.
46  PData.sFormat = 1;          % use first SFormat structure.
47  PData.pdeltaX = Trans.spacing;
48  PData.pdeltaZ = 0.5;
49  PData.Size(1) = ceil((SFormat.endDepth−SFormat.startDepth)/PData.pdeltaZ);
50  PData.Size(2) = ceil((Trans.numelements*Trans.spacing)/PData.pdeltaX);
51  PData.Size(3) = 1;          % single image page
52  PData.Origin = [−Trans.spacing*(Trans.numelements−1)/2,0,SFormat.startDepth];
53
54 % only for simulation
55 % Specify Media object.'pt1.m' script defines array of point targets.
56  pt1;
57  Media.function = 'movePoints';
```

```matlab
58
59
60  % Specify Resources.
61  Resource.RcvBuffer(1).datatype = 'int16';
62  Resource.RcvBuffer(1).rowsPerFrame = 4096*(na+1);
63  Resource.RcvBuffer(1).colsPerFrame = Resource.Parameters.numRcvChannels;
64  Resource.RcvBuffer(1).numFrames = 235;    % allows 5s of imaging
65  Resource.InterBuffer(1).datatype = 'complex';
66  Resource.InterBuffer(1).numFrames = 1;       % one intermediate frame needed
67  Resource.InterBuffer(1).rowsPerFrame = PData.Size(1) ;
68  Resource.InterBuffer(1).colsPerFrame = PData.Size(2);
69  Resource.ImageBuffer(1).datatype = 'double';    % image buffer for imaging
70  Resource.ImageBuffer(1).rowsPerFrame = PData.Size(1) ;
71  Resource.ImageBuffer(1).colsPerFrame = PData.Size(2);
72  Resource.ImageBuffer(1).numFrames = 10;
73  Resource.DisplayWindow(1).Title = 'Image Display';
74  Resource.DisplayWindow(1).pdelta = 0.3;
75  Resource.DisplayWindow(1).Position = [250,150, ...
76  ceil(PData.Size(2)*PData.pdeltaX/Resource.DisplayWindow(1).pdelta), ... % width
77  ceil(PData.Size(1)*PData.pdeltaZ/Resource.DisplayWindow(1).pdelta)];    % height
78  Resource.DisplayWindow(1).ReferencePt = [PData.Origin(1),PData.Origin(3)];
79  Resource.DisplayWindow(1).Colormap = gray(256);
80
81
82  % Specify Transmit waveform structures.
83  % for imaging
84  TW(1).type = 'parametric';
85  TW(1).Parameters = [22,17,2,1];    % A=4.09MHz , B=77% duty cycle, C=0.2us, D=1
86  % for long pulses
87  TW(2).type = 'parametric';
88  TW(2).Parameters = [22,8,31,1];    % A=4.09MHz , B=37% duty cycle, C=240us, D=1
89  TW(2).extendBL = 1;
90
91
92  % Specify TX structure array.
93  TX = repmat(struct('waveform', 1, ...
94              'Origin', [0.0,0.0,0.0], ...
95              'focus', 0.0, ...
96              'Steer', [0.0,0.0], ...
97              'Apod', ones(1,Resource.Parameters.numTransmit), ...
98              'Delay', zeros(1,Resource.Parameters.numTransmit)), 1, na+1);
99
100 % - Set event specific TX attributes.
101 if fix(na/2) == na/2        % if na even
102    startAngle = (-(fix(na/2) - 1) - 0.5)*dtheta2D;
103 else
104    startAngle = -fix(na/2)*dtheta2D;
105 end
106 for n = 1:na    % na transmit events for imaging
107   TX(n).Steer = [(startAngle+(n-1)*dtheta2D),0.0];
108   TX(n).Delay = computeTXDelays(TX(n));
109 end
110
111 TX(na+1).waveform = 2;
112 TX(na+1).Apod =[zeros(1,44)  ones(1,40)  zeros(1,44)] ;
113 TX(na+1).Delay = computeTXDelays(TX(n+1));
114 TX(na+1).focus=85; % 25mm
115
```

```matlab
116  % Specify TPC structures.
117  % for imaging
118  TPC(1).name = '2D';
119  TPC(1).maxHighVoltage = 50;
120  TPC(1).highVoltageLimit = 50;
121  % for long pulses
122  TPC(5).name = 'long pulse';
123  TPC(5).maxHighVoltage = 50;
124  TPC(5).highVoltageLimit =50;
125
126
127  % Specify Receive structure arrays.
128  maxAcqLngth2D = sqrt(SFormat(1).endDepth^2 + (Trans.numelements*Trans.spacing)^2)
        - SFormat(1).startDepth;
129
130  wl4sPer128 = 128/(4*2);  % wavelengths in a 128 sample block for 4 smpls per wave
        round trip.
131  wl2sPer128 = 128/(2*2);  % wavelengths in a 128 sample block for 2 smpls per wave
        round trip.
132  Receive = repmat(struct('Apod', ones(1,Resource.Parameters.numRcvChannels), ...
133                    'startDepth', SFormat(1).startDepth, ...
134             'endDepth', SFormat(1).startDepth + wl4sPer128*ceil(maxAcqLngth2D/
                  wl4sPer128), ...                          'TGC', 1, ...
135             'bufnum', 1, ...
136             'framenum', 1, ...
137             'acqNum', 1, ...
138             'samplesPerWave', 4, ... % samplesPerWave for 2D
139             'mode', 0, ...
140             'InputFilter', [0.0036,0.0127,0.0066,-0.0881,-0.2595,0.6494], ...
                                  'callMediaFunc', 1), 1, na*(Resource.
                  RcvBuffer(1).numFrames));
141
142
143  % - Set event specific Receive attributes.
144  for i = 1:Resource.RcvBuffer(1).numFrames
145     Receive(na*(i-1)+1).callMediaFunc = 1;
146     for j = 1:na
147        Receive(na*(i-1)+j).Apod(1:Trans.numelements) = 1.0;
148        Receive(na*(i-1)+j).framenum = i;
149        Receive(na*(i-1)+j).acqNum = j;
150     end
151  end
152
153
154  % Specify TGC Waveform structures.
155  TGC(1).CntrlPts = [400,500,625,700,750,800,850,950];
156  TGC(1).rangeMax = SFormat(1).endDepth;
157  TGC(1).Waveform = computeTGCWaveform(TGC(1));
158
159  % Specify Recon structure arrays.
160  Recon = struct('senscutoff', 0.6, ...
161          'pdatanum', 1, ...
162          'rcvBufFrame',-1, ...
163          'IntBufDest', [1,1], ...
164          'ImgBufDest', [1,-1], ...
165          'RINums',(1:na+1));
166
167
```

```
168  % Define ReconInfo structures.
169  ReconInfo = repmat(struct('mode', 4, ...   % default is to accumulate IQ data.
170                      'txnum', 1, ...
171                      'rcvnum', 1, ...
172                      'regionnum', 0), 1, na+1);
173
174  % Set specific ReconInfo attributes.
175  ReconInfo(1).mode = 3; % replace IQ data
176  for j = 1:na+1  % For each row in the column
177      ReconInfo(j).txnum = j;
178      ReconInfo(j).rcvnum = j;
179  end
180  ReconInfo(na+1).mode = 5; % accum and detect
181
182  % Specify Process structure arrays.
183  cpt = 28;  % define here so we can use in UIControl below persf = 80;
184  persp = 90;
185  DopState = 'freq';
186  Process(1).classname = 'Image';
187  Process(1).method = 'imageDisplay';
188  Process(1).Parameters = {'imgbufnum',1,...            % number of ImageBuffer to
         process.
189                   'framenum',-1,...           % frame number in source buffer (-1 =>
                          lastFrame)
190                   'pdatanum',1,...
191                   'norm',1,...                % normalization method(1 means fixed)
192                   'pgain',1.0,...             % pgain is image processing gain
193                   'persistMethod','simple',...
194                   'persistLevel',20,...
195                   'interp',1,...              % method of interpolation (1=4pt interp)
196                   'compression',0.5,...       % X^0.5 normalized to output word size
197                   'reject',2,...
198                   'mappingMode','full',...
199                   'displayWindow',1,...
200                   'display',1};               % display image after processing
201
202
203  % Specify SeqControl structure arrays.
204
205  % Jump back to start.
206  SeqControl(1).command = 'jump';
207  SeqControl(1).argument = 1;
208  % Change to Profile 1
209  SeqControl(2).command = 'setTPCProfile';
210  SeqControl(2).condition = 'immediate';
211  SeqControl(2).argument = 1;
212  % Change to Profile 5 (long pulse)
213  SeqControl(3).command = 'setTPCProfile';
214  SeqControl(3).condition = 'immediate';
215  SeqControl(3).argument = 5;
216  % Set to allow time for profile change.
217  SeqControl(4).command = 'noop';
218  SeqControl(4).argument = 8000; % time in us
219  % Time between long pulses
220  SeqControl(5).command = 'timeToNextEB';
221  SeqControl(5).argument = 251; % time in us
222  %1s   not imaging and not transmitting long pulse
223  SeqControl(6).command = 'timeToNextAcq';
```

```matlab
224 | SeqControl(6).argument = 1000000; %1s
225 | % return to MatLab
226 | SeqControl(7).command = 'returnToMatlab';
227 | % Time between 2D flash angle acquisitions
228 | SeqControl(8).command = 'timeToNextAcq';
229 | SeqControl(8).argument = 200;  % time in us
230 |
231 | % nsc is count of SeqControl objects
232 | nsc = 9;
233 |
234 | % n is count of Events
235 | n = 1;
236 |
237 | Event(n).info = 'set TPCProfile';
238 | Event(n).tx = 0;          % no TX
239 | Event(n).rcv = 0;         % no Rcv
240 | Event(n).recon = 0;       % no Recon
241 | Event(n).process = 0;
242 | Event(n).seqControl = 2;
243 | n=n+1;
244 | for i = 1:Resource.RcvBuffer(1).numFrames
245 |   if i==Resource.RcvBuffer(1).numFrames
246 |     Event(n).info = 'set TPCProfile';
247 |     Event(n).tx=0;
248 |     Event(n).rcv=0;
249 |     Event(n).recon = 0;       % no reconstruction.
250 |     Event(n).process = 0;     % no processing
251 |     Event(n).seqControl=[3,4];
252 |     n=n+1;
253 |     for t= 1:20
254 |       Event(n).info='Transmit long pulse';
255 |       Event(n).tx=(na+1);
256 |       Event(n).rcv=0;
257 |       Event(n).recon = 0;       % no reconstruction.
258 |       Event(n).process = 0;     % no processing
259 |       Event(n).seqControl = 5;
260 |       n=n+1;
261 |     end
262 |     Event(n-1).seqControl = 6;
263 |     Event(n).info='return to Matlab';
264 |     Event(n).tx=0;
265 |     Event(n).rcv=0;
266 |     Event(n).recon = 0;       % no reconstruction.
267 |     Event(n).process = 0;     % no processing
268 |     Event(n).seqControl = 7;
269 |     n=n+1;
270 |   else
271 |     % Acquire 2D frame
272 |     for j = 1:na
273 |       Event(n).info = 'Acquire 2D flash angle';
274 |       Event(n).tx = j;    % use next TX structure.
275 |       Event(n).rcv = na*(i-1)+j;
276 |       Event(n).recon = 0;       % no reconstruction.
277 |       Event(n).process = 0;     % no processing
278 |       Event(n).seqControl = 8; %
279 |       n = n+1;
280 |     end
281 |   Event(n-1).seqControl = [nsc,nsc+1]; % modify last acquisition Event's
```

```matlab
           seqControl
282     SeqControl(nsc).command = 'timeToNextAcq';     % set time between frames
283     SeqControl(nsc).argument = 20000;              % 20000usec = 20msec (~ 50 fps)
284     SeqControl(nsc+1).command = 'transferToHost'; % transfer frame to host buffer
285     nsc = nsc+2;
286     end
287 Event(n).info = 'recons and 2D process';
288 Event(n).tx = 0;           % no transmit
289 Event(n).rcv = 0;          % no rcv
290 Event(n).recon = 1;   % reconstruction
291 Event(n).process = 1;      % process 2D
292 Event(n).seqControl = 0;
293 n = n+1;
294     if floor(i/3) == i/3
295         Event(n).info = 'return to MATLAB';
296         Event(n).tx = 0;           % no transmit
297         Event(n).rcv = 0;          % no rcv
298         Event(n).recon = 0;   % reconstruction
299         Event(n).process = 0; % Exit to Matlab every 3rd frame
300         Event(n).seqControl = nsc;
301         SeqControl(nsc).command = 'returnToMatlab';
302         nsc = nsc+1;
303         n = n+1;
304     end
305 end
306 Event(n).info = 'Jump ';
307 Event(n).tx = 0;          % no TX
308 Event(n).rcv = 0;         % no Rcv
309 Event(n).recon = 0;       % no Recon
310 Event(n).process = 0;
311 Event(n).seqControl = 1; % jump command
312 n=n+1;
313 Event(n).info = 'FAKE';
314 Event(n).tx = 1;          % no TX
315 Event(n).rcv = 1;         % no Rcv
316 Event(n).recon = 0;       % no Recon
317 Event(n).process = 0;
318 Event(n).seqControl = 0; % jump command
319 n=n+1;
320 Event(n).info = 'FAKE';
321 Event(n).tx = 0;          % no TX
322 Event(n).rcv = 0;         % no Rcv
323 Event(n).recon = 0;       % no Recon
324 Event(n).process = 0;
325 Event(n).seqControl = nsc; % jump command
326 SeqControl(nsc).command = 'transferToHost';
327 nsc=nsc+1;
328
329
330 % User specified UI Control Elements
331 % - Sensitivity Cutoff
332 sensx = 170;
333 sensy = 420;
334 UI(1).Control = {'Style','text',...          % popupmenu gives list of choices
335           'String','Sens. Cutoff',...
336           'Position',[sensx+10,sensy,100,20],... % position on UI
337           'FontName','Arial','FontWeight','bold','FontSize',12,...
338           'BackgroundColor',[0.8,0.8,0.8]};
```

```
339
340  UI(2).Control = {'Style','slider',...          % popupmenu gives list of choices
341          'Position',[sensx,sensy-30,120,30],... % position on UI
342          'Max',1.0,'Min',0,'Value',Recon(1).senscutoff,...
343          'SliderStep',[0.025  0.1],...
344          'Tag','sensSlider',...
345          'Callback',{@sensCutoffCallback}};
346
347  UI(2).Callback = {'sensCutoffCallback.m',...
348          'function sensCutoffCallback(hObject,eventdata)',...
349          ' ',...
350          'sens = get(hObject,''Value'');',...
351          'ReconL = evalin(''base'', ''Recon'');',...
352          'for i = 1:size(ReconL,2)',...
353          '    ReconL(i).senscutoff = sens;',...
354          'end',...
355          'assignin(''base'',''Recon'',ReconL);',...
356          '% Set Control.Command to re-initialize Recon structure.',...
357          'Control = evalin(''base'',''Control'');',...
358          'Control.Command = ''update&Run'';',...
359          'Control.Parameters = {''Recon''};',...
360          'assignin(''base'',''Control'', Control);',...
361          '% Set the new cutoff value in the text field.',...
362          'h = findobj(''tag'',''sensCutoffValue'');',...
363          'set(h,''String'',num2str(sens,''%1.3f''));',...
364          'return'};
365
366  UI(3).Control = {'Style','edit','String',num2str(Recon(1).senscutoff,'%1.3f'), ...
     %text
367          'Position',[sensx+20,sensy-40,60,22], ...    % position on UI
368          'tag','sensCutoffValue', ...
369          'BackgroundColor',[0.9,0.9,0.9],...
370          'Callback',{@sensCutOffValueCallback}};
371
372  UI(3).Callback = {'sensCutOffValueCallback.m',...
373          'function sensCutOffValueCallback(hObject,eventdata)',...
374          ' ',...
375          'sens = str2num(get(hObject,''String''));',...
376          'ReconL = evalin(''base'', ''Recon'');',...
377          'for i = 1:size(ReconL,2)',...
378          '    ReconL(i).senscutoff = sens;',...
379          'end',...
380          'assignin(''base'',''Recon'',ReconL);',...
381          '% Set Control.Command to re-initialize Recon structure.',...
382          'Control = evalin(''base'',''Control'');',...
383          'Control.Command = ''update&Run'';',...
384          'Control.Parameters = {''Recon''};',...
385          'assignin(''base'',''Control'', Control);',...
386          '% Set the new cutoff value for the slider',...
387          'set(findobj(''Tag'',''sensSlider''),''Value'',sens);',...
388          'return'};
389
390  % - Range Change
391  rngx = 20;
392  rngy = 125;
393  UI(4).Control = {'Style','text','String','Range',...
394          'Position',[rngx+10,rngy,80,20],...
395          'FontName','Arial','FontWeight','bold','FontSize',12,...
```

```
396               'BackgroundColor',[0.8,0.8,0.8]};
397
398  UI(5).Control = {'Style','slider',...
399               'Position',[rngx,rngy-30,120,30],...
400               'Max',320,'Min',64,'Value',SFormat.endDepth,...
401               'SliderStep',[0.125,0.250],...
402               'Callback',{@rangeChangeCallback}};
403
404  UI(5).Callback = {'rangeChangeCallback.m',...
405          'function rangeChangeCallback(hObject,eventdata)',...
406          ' ',...
407          'simMode = evalin(''base'',''Resource.Parameters.simulateMode'');',...
408          '% No range change if in simulate mode 2.',...
409          'if simMode == 2',...
410          'set(hObject,''Value'',evalin(''base'',''SFormat.endDepth''));',...
411          'return',...
412          'end',...
413          'range = get(hObject,''Value'');',...
414          'assignin(''base'',''range'',range);',...
415          'SFormat = evalin(''base'',''SFormat'');',...
416          'SFormat.endDepth = range;',...
417          'assignin(''base'',''SFormat'',SFormat);',...
418          'evalin(''base'',''PData.Size(1) =
419          ceil((SFormat.endDepth-SFormat.startDepth)/PData.pdeltaZ);'');',...
420          'evalin(''base'',''[PData.Region,PData.numRegions] = createRegions(PData);''
                  );',...
421          'evalin(''base'',''Resource.DisplayWindow(1).Position(4) =
422          ceil(PData.Size(1)*PData.pdeltaZ/Resource.DisplayWindow(1).pdelta);'');',...
423          'Receive = evalin(''base'', ''Receive'');',...
424          'Trans = evalin(''base'', ''Trans'');',...
425          'maxAcqLength = sqrt(range^2 +
426          (Trans.numelements*Trans.spacing)^2)-SFormat.startDepth;',...
427          'wlsPer128 = 128/(4*2);',...
428          'for i = 1:size(Receive,2)',...
429          '    Receive(i).endDepth = SFormat.startDepth +
430          wlsPer128*ceil(maxAcqLength/wlsPer128);',...
431          'end',...
432          'assignin(''base'',''Receive'',Receive);',...
433          '% Update VDAS parameters of Receive objects.',...
434          'evalin(''base'',''updateVDAS('''''Receive''''')'');',...
435          'evalin(''base'',''TGC.rangeMax = SFormat.endDepth;'');',...
436          'evalin(''base'',''TGC.Waveform = computeTGCWaveform(TGC);'');',...
437          'Control = evalin(''base'',''Control'');',...
438          'Control.Command = ''update&Run'';',...
439          'Control.Parameters =
440          {''SFormat'',''PData'',''Receive'',''Recon'',''DisplayWindow'',''ImageBuffer
                  ''};',...
441          'assignin(''base'',''Control'', Control);',...
442          'assignin(''base'', ''action'', ''displayChange'');',...
443          'h = findobj(''tag'',''rangeValue'');',...                          'set(h,''
               String'',num2str(range,''%3.0f''));',...
444          'return'};
445
446  UI(6).Control = {'Style','edit','String',num2str(SFormat.endDepth,'%3.0f'), ...%
         text
447               'Position',[rngx+20,rngy-40,60,22], ...    % position on UI
448               'tag','rangeValue', ...
449               'BackgroundColor',[0.9,0.9,0.9]};
```

```
450
451  clear i j n sensx sensy rngx rngy
452  % Specify factor for converting sequenceRate to frameRate.
453  frameRateFactor = 5;
454  % Save all the structures to a .mat file.
455  save('L7-4Imaging');
```

```matlab
% File name Control_tests.m:
% code for filtering the Control tests pressure data
% the code is equal for all the protocols (only data are varying)

clear all
close all
clc

% T8 T4 and T10 are 12200x2:
%first column pressure data
%second column zeros
T8=load('test8.txt');
T4=load('test4.txt');
T10=load('test3.txt');


%moving average filter parameters
A=1;
B=ones(1,200)/200;

% create in second column the time vector (equal for all)
i=1;
[r,c]=size(T4);
tmax=r*0.1;
time=zeros(r,1);
for j=0:0.1:tmax-1
    time(i)=j;
    i=i+1;
end
time(:,1)=time(:,1)/60; %convert to minutes

%apply filter
[y4,p4]=filter(B,A,T4(:,1) ); % p4 final condition
y4=filter(B,A,T4(:,1),p4 ); % p4 initial condition
[y8,p8]=filter(B,A,T8(:,1) );
y8=filter(B,A,T8(:,1),p8 );
[y10,p10]=filter(B,A,T10(:,1) );
y10=filter(B,A,T10(:,1),p10 );

%save from sample 201
y1=y8(201:12200);
y2=y4(201:12200);
y3=y10(201:12200);
t=time(1:12000);

%save in file results Control
save('results_Control','y1','y2','y3','t')
```

```matlab
1  % File name Pressure_normalized.m:
2  % Code for normalized mean pressure plot for all protocols (for Comparison)
3  clear all
4  close all
5  clc
6
7  %% control group
8  load('results_Control.mat')  %contains filtered data in y1, y2, y3
9  y1=y1+(100-y1(1));
10 y2=y2+(100-y2(1));
11 y3=y3+(100-y3(1));
12
13 %% Protocol 1
14 load('results_protocol1.mat') %contains filtered data in y4, y5, y6
15 y4=y4+(100-y4(1));
16 y5=y5+(100-y5(1));
17 y6=y6+(100-y6(1));
18
19 %% Protocol 2
20 load('results_protocol2.mat')
21 y7=y7+(100-y7(1));
22 y8=y8+(100-y8(1));
23 y9=y9+(100-y9(1));
24
25 %% Protocol 3
26 load('results_protocol3.mat')
27 y10=y10+(100-y10(1));
28 y11=y11+(100-y11(1));
29 y12=y12+(100-y12(1));
30
31 %% Protocol 4
32 load('results_SonoVue_50')
33 y13=y13+(100-y13(1));
34 y14=y14+(100-y14(1));
35 y15=y15+(100-y15(1));
36
37 %% mean computation
38 control_test=[y1
39 y2
40 y3
41 ];
42 protocol1=[y4
43 y5
44 y6
45 ];
46 protocol2=[y7
47 y8
48 y9
49 ];
50 protocol3=[y10
51 y11
52 y12];
53 protocol4=[y13
54 y14
55 y15];
56
57 mean_control=zeros(12000,1);
```

```matlab
for i=1:12000
   tmp=0;
    for j=1:3
       tmp=tmp+control_test(j,i);
    end
mean_control(i)=tmp/3;
clear tmp
end

mean_protocol1=zeros(12000,1);
for i=1:12000
   tmp=0;
    for j=1:3
       tmp=tmp+protocol1(j,i);
    end
mean_protocol1(i)=tmp/3;
clear tmp
end

mean_protocol2=zeros(12000,1);
for i=1:12000
   tmp=0;
    for j=1:3
       tmp=tmp+protocol2(j,i);
    end
mean_protocol2(i)=tmp/3;
clear tmp
end

mean_protocol3=zeros(12000,1);
for i=1:12000
   tmp=0;
    for j=1:3
       tmp=tmp+protocol3(j,i);
    end
mean_protocol3(i)=tmp/3;
clear tmp
end

mean_protocol4=zeros(12000,1);
for i=1:12000
   tmp=0;
    for j=1:3
       tmp=tmp+protocol4(j,i);
    end
mean_protocol4(i)=tmp/3;
clear tmp
end

figure(6)
plot(t,mean_control)
hold on
plot(t,mean_protocol1,'r')
hold on
plot(t,mean_protocol2,'g')
hold on
plot(t,mean_protocol3,'m')
hold on
```

```matlab
116  plot(t,mean_protocol4,'k')
117  xlabel('time(min)')
118  ylabel('pressure ')
119  legend('Control tests', 'Protocol 1', 'Protocol 2', 'Protocol 3','Protocol 4')
120
121  sd1=std(control_test(:,600));
122  for i=1:20
123     hold on
124     sd=std(control_test(:,i*600));
125     errorbar(t(i*600),mean_control(i*600),sd,'x b')
126  end
127  for i=1:20
128     hold on
129     sd=std(protocol1(:,i*600));
130     errorbar(t(i*600),mean_protocol1(i*600),sd,'o r')
131  end
132  for i=1:20
133     hold on
134     sd=std(protocol2(:,i*600));
135     errorbar(t(i*600),mean_protocol2(i*600),sd,'s g')
136  end
137  for i=1:20
138     hold on
139     sd=std(protocol3(:,i*600));
140     errorbar(t(i*600),mean_protocol3(i*600),sd,'d m')
141  end
142  for i=1:20
143     hold on
144     sd=std(protocol4(:,i*600));
145     errorbar(t(i*600),mean_protocol4(i*600),sd,'* k')
146  end
```