# Synthetic Tracked Aperture Ultrasound Imaging:

## Virtual Fixtures for Co-Robotic Control

Kalyna Apkarian and Rodolfo Finocchi

Spring 2016

## Table of Contents

## Team Members

Kalyna Apkarian
Rodolfo Finocchi

## Mentors

Kai Zhang
Dr. Emad Boctor
Dr. Russell Taylor

## Motivation

Ultrasound imaging is widely used in the clinical setting to visualize a patient's anatomy quickly, easily, and at a low cost. However, the main problem with ultrasound imaging is due to the aperture size of the transducer, which limits image quality in deep tissues. Synthetic aperture ultrasound imaging (STrAtUS) is a technique that synthesizes data from multiple sub-apertures using tracking data and has been shown to provide an improvement in image quality. The current synthetic aperture system uses a UR5 robot arm to autonomously scan the desired trajectory on the patient. This presents problems for clinical translation due to patient safety, ease of use by the sonographer, and force control requirements for anatomy-specific imaging.

## Goals

Our goal is to bring the current system from autopilot to co-robotic freehand. This will be done by implementing guidance virtual fixtures, force sensing, and compliance force control. This will assist the operator in completing a safer and more accurate procedure.

## Relevance

Achieving these goals will essentially bridge the gap between an autonomous robot and direct control by the user. First and foremost, this project will ensure patient safety. Secondly, it will allow for ease of use by the sonographer. Additionally, the force sensing and control component will allow for higher quality imaging of more complex regions, such as the abdomen, by guaranteeing a constant amount of force exerted.

## Technical Approach

### System architecture

The information flow of the system is outlined in Figure 1. Data flows from the force torque sensor, given readings from the user, to the PC, and back to the robot controller. The force sensor highlighted in green is a 6-axis force-torque sensor (Robotiq FT150), and is placed at the wrist of the UR5 arm. Originally in our proposal, we thought that two force sensors may be necessary, but we were able to achieve the desired outcomes with just the Robotiq sensor. This is used both for admittance control and for contact force control as an option in the virtual fixture formulation. This formulation will be explained in detail below.

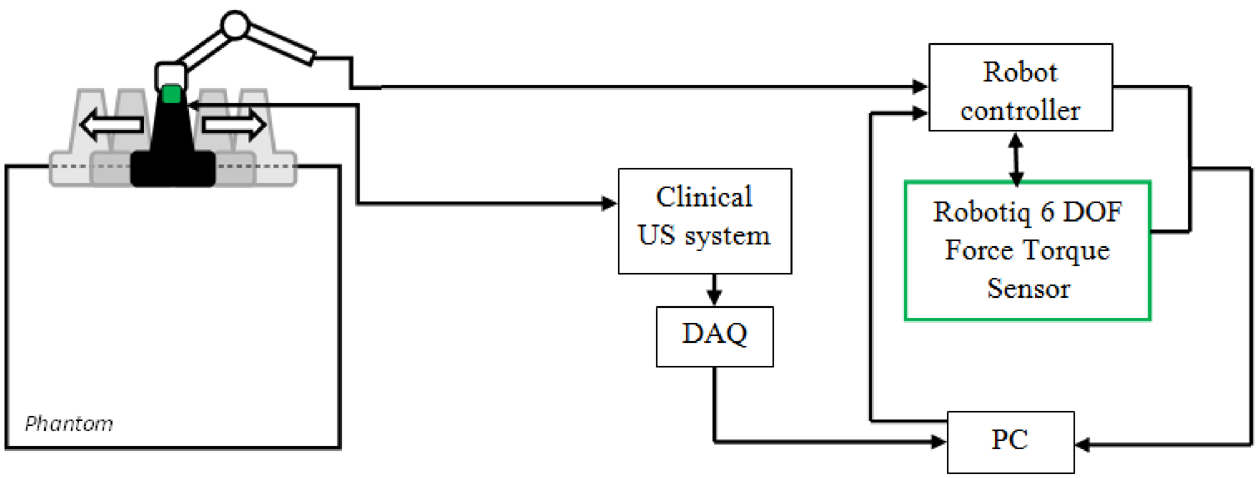


**Figure 1.** System architecture.

### System Flowchart

The flowchart below in Figure 2 represents how our virtual fixtures and force sensing are integrated with the current STrAtUS system. It demonstrates how the improved system with co-robotic control will be used together with input from the sonographer to create the output of the B-mode image. Each of the components that are essential to this co-robotic control and how they serve to assist in image formation will be outlined in detail in the section on virtual fixtures.

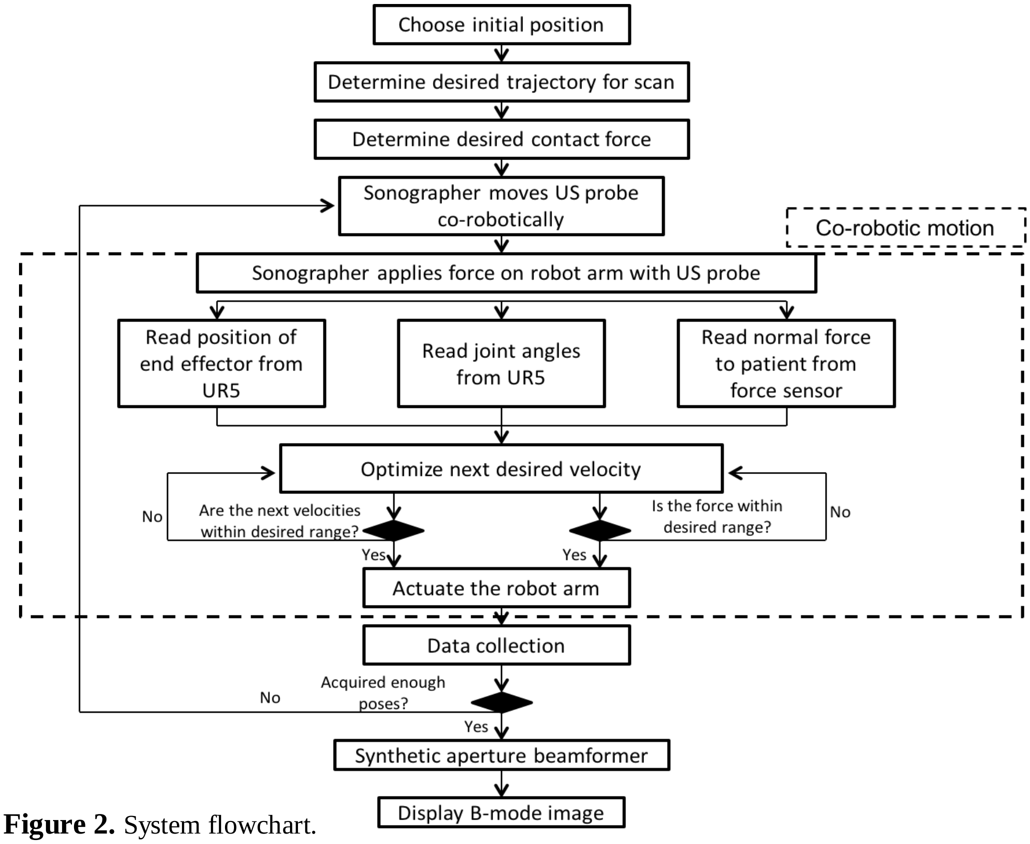


**Figure 2.** System flowchart.

## Admittance Force Control

In order to control the robot's motion, force readings applied at the ultrasound (US) probe attached to the UR5's end effector are translated into velocity commands with the following admittance control law:

$$\dot{x}_{des} = K(f)$$

where $f$ is the readings of the FT150 force sensor and $K$ is a diagonal matrix of non-linear gains whose value is determined by trial and error. This matrix scales the input force in such a way that makes the robot's end effector translate and rotate in space in a smooth manner given the user's hand motions. Forces or torques applied in one direction will make the robot move in that direction at a velocity proportional to the force's magnitude.

## Virtual Fixtures

Virtual fixtures are a class of robot control algorithms that are used to augment motion commands from the user, thus enhancing precision, stability, and patient safety. These virtual fixtures can be implemented given information on the instantaneous kinematics of the robot and the geometric constraints specified based on the desired behavior of the system. In our system, we would like to ensure that the user only scans the desired area and nowhere else. We also want to provide the option to control the contact force once a desired trajectory to scan is selected. Additionally, we want to impose limits on the joint velocities to ensure patient and user safety. One of the most common ways to do this as presented in the literature is with the constrained optimization approach, which we have implemented in our system as follows.

Our goal is to solve for the optimal joint velocities according to the equation $\dot{q}_{des} = \text{argmin}_{\dot{q}_{des}} \| J\dot{q}_{des} - \dot{x}_{des} \|$, where $\dot{q}_{des}$ is a 6-vector of the desired joint velocities, $J$ is the 6x6 Jacobian matrix that relates joint space to task space, and $\dot{x}_{des}$ is the 6-vector of the end-effector Cartesian velocities determined by the admittance control law explained above. As it is, this allows for unconstrained admittance control of the robot. We would like to impose constraints on this optimization problem such that the motion of the ultrasound probe is restricted to the desired area. These constraints take the form $H * \dot{q}_{des} \leq h$, where $H$ is the constraint coefficient matrix and $h$ is the constraint vector. These matrices are built up and arranged based on which geometric constraints the user specifies. There are three main geometric constraints that we have implemented: move along a line, move within a plane, and move rotationally. In addition to these, we have implemented force control constraints to set a desired contact force and maintain it while scanning. As mentioned, we also limit joint velocities in this formulation. These constraints are each outlined in detail below.

<u>Move along a line</u>

We define a line in 3D space along which we constrain the probe to remain while being moved by the user. This line is given by $L = \overrightarrow{L_0} + \hat{l} * s$, where $\overrightarrow{L_0}$ is a point on the line and $\hat{l}$ is the unit vector indicating the direction of the line, as can be seen in Figure 3[1]. With each incremental motion, $P_{cl}$, the closest point on the line L to the current position $\vec{x}_p$ is computed. The signed errors are then calculated according to $\overrightarrow{\delta_p} = \vec{x}_p - P_{cl}$ and $\delta_r = 0$, in this case, since there is no rotation. The vector $\overrightarrow{\delta_p} + \Delta\vec{x}_p$ is then projected onto the plane perpendicular to the line L according to the following method. A rotation matrix is computed to transform the plane to the world frame. The rotation matrix has the form

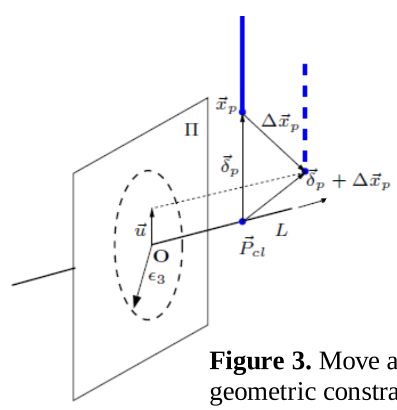

**Figure 3.** Move along a line geometric constraint. Source: 1.

[1] Li, Ming, Ankur Kapoor, and Russell H. Taylor. "A constrained optimization approach to virtual fixtures." *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005.

$R = [\widehat{v_1} \quad \widehat{v_2} \quad \hat{l}]$, where $\widehat{v_1} = \frac{\hat{l} \, x \, \hat{l}'}{\|\hat{l} \, x \, \hat{l}'\|}$ and $\widehat{v_2} = \frac{\hat{l} \, x \, \widehat{v_1}}{\|\hat{l} \, x \, \widehat{v_1}\|}$, where $\hat{l}'$ is any arbitrary unit vector not in the same direction as $\hat{l}$. This rotation matrix allows a vector to be written in the world frame by pre multiplying it by $(R * [\cos(\alpha_i); \quad \sin(\alpha_i); \quad 0])^T$. Next, we require the projection of $\overrightarrow{\delta_p} + \Delta\vec{x}_p$ to be less than some error, $\varepsilon$, approximated by an n-dimensional polygon. This constraint is then written as $[(R * [\cos(\alpha_i); \quad \sin(\alpha_i); \quad 0])^T \quad 0 \quad 0 \quad 0] \cdot (\overrightarrow{\delta_p} + \Delta\vec{x}) \leq \varepsilon$ for i=1:n, where $\alpha_i = \frac{2*\pi*i}{n}$. Rewriting in the desired form of the general problem, $H * \dot{q}_{des} \leq h$ gives

$$H = \begin{bmatrix} (R * [\cos(\alpha_1); \quad \sin(\alpha_1); \quad 0])^T & 0 & 0 & 0 \\ & \vdots & & \\ (R * [\cos(\alpha_n); \quad \sin(\alpha_n); \quad 0])^T & 0 & 0 & 0 \end{bmatrix}$$

$$\vec{h} = \begin{bmatrix} \varepsilon \\ \vdots \\ \varepsilon \end{bmatrix} - H * \vec{\delta}$$

In practice, we have set the rotational components of the linear velocity vector to zero to ensure that the motion is strictly translational and the probe stays locked perpendicular to the line.

Move within a plane:

Another geometric constraint, move within a plane, is simply an extension of move along a line. It can be used to either prevent penetration of a certain plane or restrict motion to within a plane. For STRATUS, we use it to constrain motion along a plane when force constraints will be introduced. It follows the same form of move along a line, where we instead look for the normal direction to the plane, $\hat{d}^t$, instead of the unit vector direction of the line. The closest point on the plane is similarly found, the translational error calculated, and then the problem takes the form

$$H = \begin{bmatrix} \hat{d}^t & 0 & 0 & 0 \\ -\hat{d}^t & 0 & 0 & 0 \end{bmatrix}$$

$$\vec{h} = \begin{bmatrix} 0 \\ -\varepsilon \end{bmatrix} - H * \vec{\delta}$$

Rotational motion:

The rotational constraint is essentially the same as the constraint to move within the plane, except that the rotational component is unlocked in one direction. In practice, this means that only two rotational components are set to zero, while the rotation along the axis normal to the length of the probe is not. This is done to ensure that the probe still does not go out of plane, but the probe can be rotated away from 90º to accommodate a more realistic anatomy.

Joint velocity limits:

This constraint simply takes the form $\dot{q}_{des} \le \dot{q}_{max}$ and $-\dot{q}_{des} \le \dot{q}_{min}$, so $H = \begin{bmatrix} I_{6x6} \\ -I_{6x6} \end{bmatrix}$ and $h = \begin{bmatrix} \dot{q}_{max} \\ \dot{q}_{min} \end{bmatrix}$. In practice, we have set the joint velocity limits at ± 0.5 rad/s, to ensure that the robot cannot move too quickly for the user to respond. This constraint is used no matter what geometric constraint is specified by the user.

Force limits:

For safety purposes and to avoid any harm to a patient, the probe-to-patient contact force applied by the robot needs to be limited. This value, denoted as $f_{lim}$, is set to 25N to allow the sonographer to comfortably manipulate the probe through a wide range of forces but ensure patient safety. In practice, the current force value measured by the Robotiq FT150, $f_{curr}$, plus any change in force, $\Delta f$ must be less than $f_{lim}$. By the definition of material compliance $K = \frac{\Delta x}{\Delta f} \rightarrow \Delta f = \frac{\Delta x}{K}$. Given the relationship between incremental motions and velocity, $\dot{x} = \frac{\Delta x}{\Delta t}$, this becomes $\Delta f = \frac{\dot{x}\Delta t}{K}$. Therefore,

$$f_{curr} + \frac{\dot{x}\Delta t}{K} \le f_{lim} \rightarrow \dot{x} \le \frac{K(f_{lim} - f_{curr})}{\Delta t}$$

In this case, the admittance gain matrix K plays the role of the material compliance. This comes into the joint velocity optimization framework described above in the form of the following constraint coefficient matrix $H_{lim}$ and the corresponding constraint vector $h_{lim}$:

$$H_{lim} = Ad_g \cdot I_{6x6}$$

$$h_{lim} = \frac{K(f_{lim} - f_{curr})}{\Delta t}$$

where $Ad_g$ denotes the adjoint matrix that resolves velocities associated with the force measured by the FT150 to the frame of the robot. The transformation that relates the force sensor coordinate frame of the robot end effector to that of the robot base is given by the transformation $g = \begin{bmatrix} R & p \\ 0_{1x3} & 1 \end{bmatrix}$ where $R$ is the rotation and $p$ is the displacement between the two. From this, $Ad_g = \begin{bmatrix} R & \hat{p}R \\ 0_{3x3} & R \end{bmatrix}$ where $\hat{p}$ denotes the skew symmetric matrix associated with the vector $p$.

Approach desired force:

In certain cases, it might be desirable to have the robot control the contact force autonomously such that remains relatively constant throughout an imaging procedure, such as one involving a "Move within a plane" virtual fixture. When the program is set to keep the desired force constant, any change $\Delta f$ from the current force reading, $f_{curr}$ must stay close to the desired force, $f_{des}$ within a margin of error $\epsilon_{force}$. Similar to the formulation for force limits, this required behavior can be defined by:

$$f_{des} - \epsilon_{force} \leq f_{curr} + \frac{\dot{x}\Delta t}{K} \leq f_{des} + \epsilon_{force}$$

Solving for $\dot{x}$ leads to:

$$\frac{K(f_{des} - \epsilon_{force} - f_{curr})}{\Delta t} \leq \dot{x} \leq \frac{K(f_{des} + \epsilon_{force} - f_{curr})}{\Delta t}$$

The constraint coefficient matrix $H_{desired}$ and the corresponding constraint vector $h_{desired}$ are calculated as follows:

$$H_{desired} = \begin{bmatrix} -Ad_g \cdot I_{6x6} \\ Ad_g \cdot I_{6x6} \end{bmatrix}$$

$$h_{desired} = \begin{bmatrix} \frac{K\left(\epsilon_{force} + f_{curr} - f_{des}\right)}{\Delta t} \\ \frac{K\left(\epsilon_{force} - f_{curr} + f_{des}\right)}{\Delta t} \end{bmatrix}$$

In practice, $f_{des}$ can be set manually by the user or set while our program is running by using the "Set Desired Contact Force" button of our Graphical User Interface (GUI) explained in the next section.

Multiple Constraints:

On various occasions, the optimization problem might be subjected to multiple constraints. For example, in the case where in-plane contact force control is desired, three constraints are applied: move within a plane, approach desired force, and joint velocity limits. To do so, the constraint coefficient matrix $H$ and the corresponding constraint vector $h$ are structured as follows:

$$H = \begin{bmatrix} H_{desired} & 0 & 0 \\ 0 & H_{plane} & 0 \\ 0 & 0 & H_{joint} \end{bmatrix}$$

$$h = \begin{bmatrix} h_{desired} \\ h_{plane} \\ h_{joint} \end{bmatrix}.$$

The optimization problem therefore becomes

$$\dot{q} = \underset{\dot{q}}{\text{argmin}} \| J\dot{q} - \dot{x} \|$$

$$\text{subject to: } \begin{bmatrix} H_{desired} & 0 & 0 \\ 0 & H_{plane} & 0 \\ 0 & 0 & H_{joint} \end{bmatrix} \begin{bmatrix} J(\theta) \\ J(\theta) \\ I_{6x6} \end{bmatrix} \dot{q} \leq \begin{bmatrix} h_{desired} \\ h_{plane} \\ h_{joint} \end{bmatrix}$$

where $J$, the robot's Jacobian matrix works to convert the Cartesian spatial constraints defined in the approach desired force and move within a plane virtual fixtures into their corresponding configuration space constraints.

**<u>Graphical User Interface</u>**

A GUI was designed as a way to integrate all of the above information in a straightforward way. As can be seen in Figure 4, the user first must choose to connect with the robot. As soon as a connection is established, the robot can be moved according to the <u>unconstrained</u> admittance control problem. At this point, the buttons “Line”, “Plane”, and “Rotate” become enabled. At any time, the user can select to move along a line, move within a plane, or move rotationally.

If either of the last two are chosen, they can choose to set the desired contact force to the current contact force at any time and activate force assistance. Also, if imaging, the user can click the button to begin collecting imaging data from the Ultrasonix machine at any time. Clicking the reset button at the bottom will remove all constraints and return to unconstrained

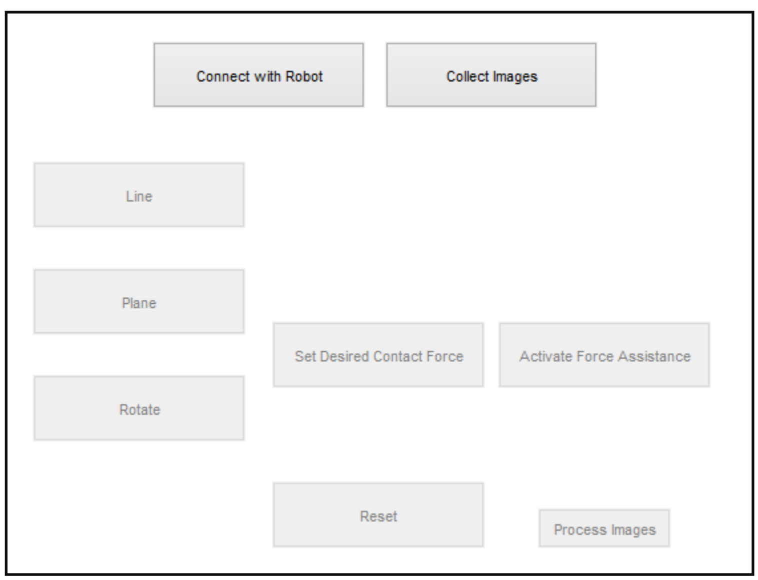


**Figure 4**. Graphical User Interface.

admittance control. One important thing to note is that if there is ever a constrained problem that is overly stringent and cannot be solved, the code will default to this unconstrained admittance control.

**Image Formation**

In the last step of the STRATUS imaging algorithm, the location and pose of the robot's end effector, collected from the forward kinematics of the robot in real time, is used to reconstruct the final image. The frame of each probe pose is first transformed into a single uniform coordinate system that relates all pose data into the image frame. The radiofrequency data collected from the Ultrasound machine for each transformed pose can then be summed up to create the final image.

## Results

The plots in Figures 5, 6 and 7 demonstrate the trajectory of the probe as reported by the robot tracking data while using the specified constraints. In each of these cases, the error was set to ± 0.5 mm in each direction. Figure 5 shows that the probe can only move in one direction in 3D space, along the direction of the line, while motion in all other directions is essentially locked.

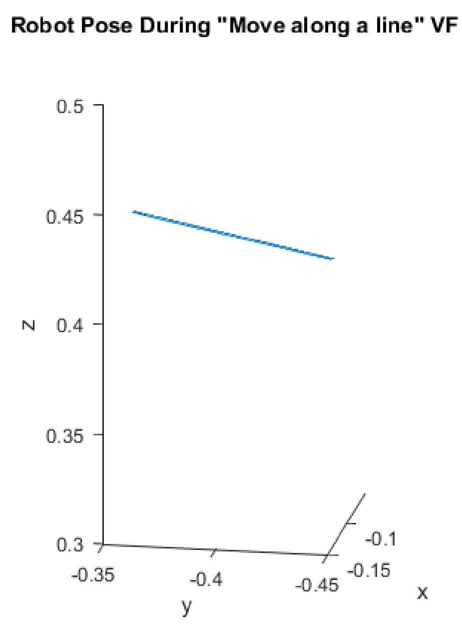


**Figure 5**. “Move along a line”


**Figure 6.** “Move along a plane”

Figure 6 shows the probe moving in an in-plane trajectory. It is similar to the previous behavior, except that one additional degree of freedom is unlocked. This feature allows the user to collect in-plane imaging data on non-planar structures. Once again, it is possible to see the accuracy with which the probe location remains on the plane with little to none deviation. This option can also be integrated with contact force control to maintain the force between the probe and patient as constant as possible while imaging or to have the probe return to the desired force setting after being lifted up for readjustment.

Figure 7 shows the result of activating the rotational virtual fixture. This feature allows the probe to be moved in a planar manner with an additional degree of freedom, rotation around the direction normal to the probe's axial direction, unlocked. This allows for more anatomically accurate scanning of rounded structures, such as the abdomen. This can also be combined with contact force control to have the probe apply a controlled force magnitude against the patient.

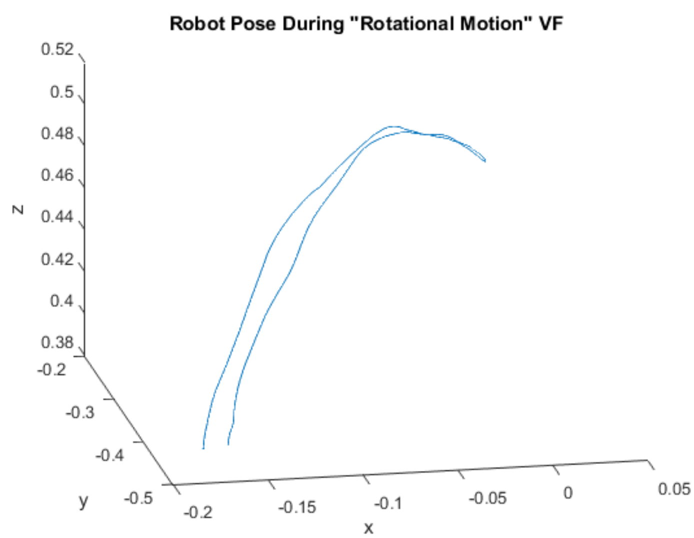


**Figure 7**. "Rotational motion"

For validation, we ran our STRATUS/virtual fixture system to image a planar general ultrasound phantom. A lateral, in-plane motion was applied and 105 poses were collected. Figure 8 shows a single pose B-mode ultrasound image without any modification. Figure 9 shows the results of using the STRATUS reconstruction algorithm using synthetic apertures extended either 10 (top image) or 60 mm (bottom image). By integrating the information from all poses, the field-of-view of the image expanded by 65.5 mm, and makes it possible to visualize many features not visible in the single-pose image.

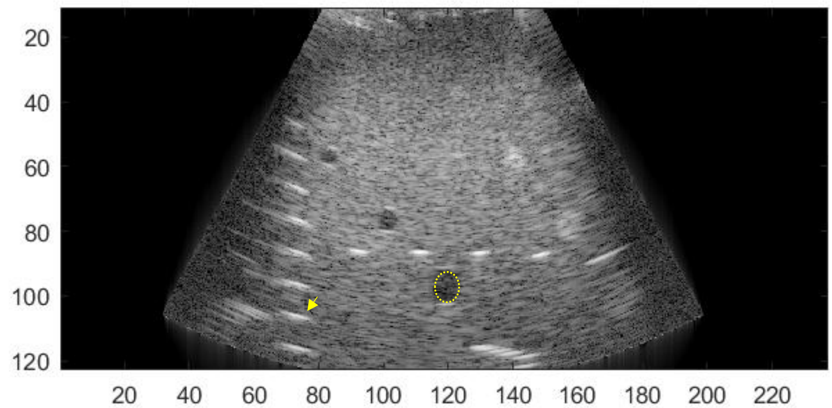

**Figure 8.** Single pose B-mode ultrasound image. The dynamic range was set to -70 dB, the display scale unit is millimeter. The pointed point target was used to compute FWHM and SNR, and the dot region was selected to calculate contrast.

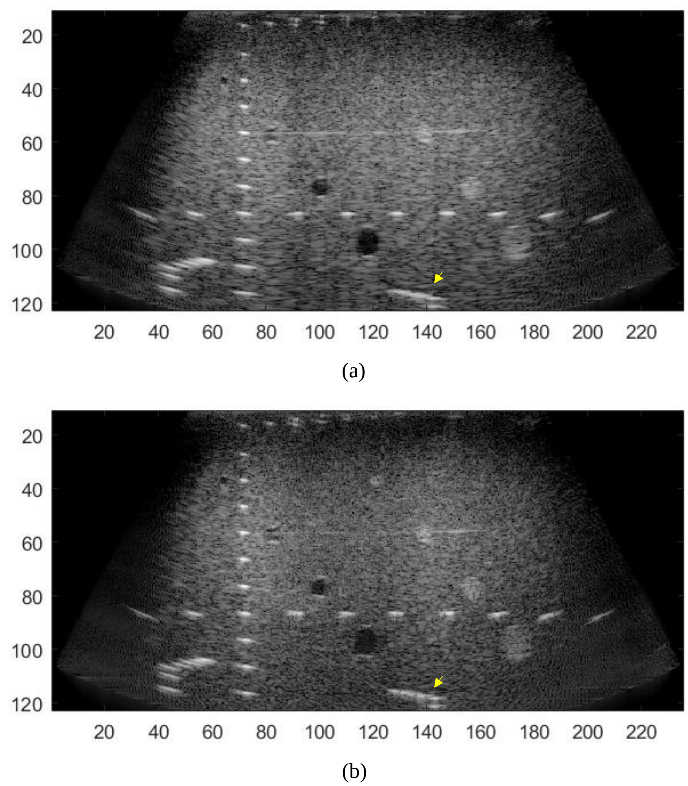


**Figure 9.** STRATUS images synthesized the range of 10 mm (a) and 60 mm (b) motion data. The yellow arrow region indicates that the separation of point fiducials became more obvious as the synthesis range increases.

| | Single | STRATUS: 10 mm | STRATUS: 60 mm |
|---|---|---|---|
| **FWHM (mm)** | 3.87 | 3.96 | 2.37 |
| **Contrast (dB)** | -7.14 | -13.24 | -10.67 |
| **SNR (dB)** | 25.01 | 27.20 | 29.35 |

**Table 1.** Increase in overall image quality with STRATUS as compared to single B-mode scan.

A quantitative comparison of the single B-mode image with the STATUS image is shown in Table 1. The full width at half maximum (FWHM), an inverse measure of resolution decreases for the 60 mm image. Additionally, contrast improves as does signal-to-noise ratio (SNR). Overall, the combination of STRATUS imaging and Virtual Fixtures showed an image quality improvement over conventional ultrasound across all metrics, which could change the paradigm of ultrasound imaging.

## Management Summary

Overall, we have accomplished most of our deliverables, as can be seen below. We finished our minimum and expected deliverables before the originally scheduled dates, and finished one of the maximum deliverables in a reasonable amount of time. We decided not to focus our time on the last maximum deliverable so that we could ensure that our code was clean and commented, and that we had working demo code for the poster session. We also had to spend some time modifying the current synthetic aperture image reconstruction code, even though it was not an explicitly stated deliverable, to demonstrate the functionality of our system with the existing STRATUS system. However, even though we did not get results for our final maximum deliverable, we believe that the system as it is should work on an abdominal phantom, but would be a good starting point for the next student taking over the project.

<u>Deliverables</u>

Minimum **(Planned by 3/14/16)**

- Code implementing virtual fixtures **(Complete 3/4/16)**
- Code implementing compliance force control **(Complete 3/1/16)**
- Comparison of actual trajectory of robot with planned trajectory **(Complete 3/7/16)**
- Demonstration of translational path in water tank using co-robotic control **(Complete 3/7/16)**

Expected **(Planned by 4/8/16)**

- Demonstration of translational path on general US phantom without contact force control **(Complete 3/11/16)**
- Demonstration of translational path on general US phantom with contact force control **(Complete 4/1/16)**

Maximum **(Planned by 4/29/16)**

- Demonstration of rotational path in water tank using co-robotic control **(Complete 4/27/16)**
- Demonstrate control on more anatomically accurate path using rotation and force control on abdominal phantom **(Future work)**

Due to some similarities with the Master's thesis project being completed simultaneously by one of our team members, Rodolfo Finocchi, he was assigned the task of developing the algorithms required for admittance control, contact force control and force limiting.

Kalyna Apkarian's role consisted of gaining a deep understanding of virtual fixtures and how to use them. She led the team in the development of the software required for the implementation of our necessary virtual fixtures.

All actual software development and testing, as well as data collection was done together during our bi-weekly meetings.

This project taught us a number of important lessons that will affect us in our future work. First, we learned the importance of clearly documenting code, particularly when working in a team developing separately. Next, we learned how to critically read scientific literature and understand how to apply its contents to our projects. Lastly, we were able to apply the concepts learned in our robotics coursework to a real-world application, deepening our understanding of the subject matter.

## Technical Appendix

All necessary code will be made available on our Wiki page.