

A Framework for Integrating real-time MRI with Robot Control: Application to Simulated Transapical Cardiac Interventions

Nikhil V. Navkar, Zhigang Deng, Dipan J. Shah, and Nikolaos V. Tsekos

Abstract—The advent of intraoperative real-time image guidance has led to the emergence of new surgical interventional paradigms including image-guided robot assistance. Most often the use of an intraoperative imaging modality is limited to visual perception of the area of procedure. In this work, we propose a framework for performing robot-assisted interventions with real-time Magnetic Resonance Imaging (rtMRI) guidance. The described computational core of this framework, processes on-the-fly rtMRI, integrates the processed information with robot control and renders it on the human-machine interfaces. This information is rendered on a visualization and force-feedback interface for enhanced perception of a dynamic area of procedure and for assisting the operator in the safe and accurate maneuvering of a robotic manipulator. The framework was experimentally tested by applying it to a simulated Transapical Aortic Valve Implantation with a virtual robotic manipulator. rtMRI data was processed on-the-fly in a rolling-window scheme and together with a multi-threaded and multi-hardware implementation, the core delivered appropriate speed of 20Hz for visualization and 1000Hz for force-feedback. The experimental results demonstrate significant improvement in the simulated task by both decreasing the duration of the procedure by half and increasing safety in the presence of cardiac and breathing motion by reducing the duration or incidents the operator collides with the tissue.

Index Terms—Magnetic resonance imaging, Heart, Surgical guidance/navigation, Medical robotics, Image-guided treatment.

I. INTRODUCTION

IMPROVEMENTS in the fields of image-guidance and robotic-assistance in interventional medicine have led to new methodologies and surgical systems that promise to improve patient management and, eventually, contribute to the reduction of the overall cost of health care. Toward such an endeavor, intraoperative imaging plays a critical role since it has the potential to offer contextually rich information about the area of the procedure (AoP). An ever-growing body of literature demonstrates the integration of robotic manipulators and intraoperative imaging, real-time or intermittent with the procedure, using x-ray fluoroscopy, ultrasound

(US) or Magnetic Resonance Imaging (MRI) [1]. Examples of demonstrated image-guided and robot assisted procedures includes applications in urology [2], cardiac ablation with fluoroscopy guided Sensei robotic catheter system (Hansen medical, USA) [3], and MRI-guided procedure like the resection of brain tumor with the NeuroArm [4], breast biopsy [5] and brachytherapy [6]. In these works the operator inspects visually the imaging information and maneuvers the robotic manipulator. In addition, computational approaches can be implemented to extract anatomical and functional information from those real-time imaging feeds on-the-fly. This information can then be used to drive human machine interfaces to assist the operator for safe and accurate maneuvering of the robotic manipulator. This would prove beneficial especially in case of complex clinical paradigms that requires operator to maneuver a robot inside a highly dynamic AoP. As an example of these clinical paradigms are the image-guided procedures on the beating heart, such as electrophysiology [3], tissue removal [7], valvuloplasties [8], and angioplasties [9]. These clinical paradigms are the focus of our work.

Existing literature in the field of image-guided robotic surgeries is vast and includes highly innovative approaches. Herein we only refer to the works that involve the use of imaging for the guidance of the interventional tools inside highly dynamic AoPs, such as interventions in the beating-heart. Image-guided manual or robot-assisted cardiac procedures are usually performed with x-ray fluoroscopy or ultrasound guidance. Three-dimensional (3D) ultrasound is in particularly suitable for cardiac procedures due to its real-time volumetric data collection and lack of ionizing radiation. Works have demonstrated the integration of 3D ultrasound with robotic systems and the use of real-time imaging to synchronize the motion of a device and the heart [10], [11]. To improve the information available to the operator intra-operatively, numerous works use pre-operative imaging sets collected with MRI or Computer Tomography (CT) [12], [13], [14]. Indeed, pre-operative imaging usually offers higher spatial resolution and diagnostic information due to better image quality and available contrast mechanisms. Among those pioneering works, Park et al. [12] were the first to describe the use of virtual fixtures in cardiac surgeries (specifically for coronary artery bypass graft), by using preoperative cardiac CT. In the context of surgical applications, virtual fixtures are software-generated virtual entities that assist the operator by constraining the motion of an actuated manipulator [15], [16]. Virtual fixtures are usually used to either prevent collision and harming an

Nikhil V. Navkar is with the Medical Robotics Laboratory and the Computer Graphics & Interactive Media Laboratory at the University of Houston, 4800 Calhoun Road, Houston, TX 77004, USA (e-mail: nvnnavkar@cs.uh.edu).

Zhigang Deng is with the Computer Graphics & Interactive Media Laboratory and the Computer Science Department, at the University of Houston, 4800 Calhoun Road, Houston, TX 77004, USA (e-mail: zdeng@cs.uh.edu).

Dipan J. Shah is with the Methodist DeBakey Heart & Vascular Center, The Methodist Hospital, 6550 Fannin, Houston, TX 77030, USA (e-mail: djshah@tmhs.org).

Nikolaos V. Tsekos is with the Medical Robotics Lab and the Computer Science Department at the University of Houston, 4800 Calhoun Road, Houston, TX 77004, USA (phone: 713-743-3350 fax: 713-743-3335 e-mail: ntsekos@cs.uh.edu).

anatomical structure or allowing movement of a device along a particular path. In image-guided procedures, these constraints are extracted from images. When considering the case of dynamic AoP in cardiac procedures, an important contribution by Ren et al. [13] introduced the concept of dynamic virtual fixtures for minimally invasive robot assisted cardiac surgeries. In this work, the fixtures were used to guide a virtual robotic manipulator to perform a precise cut on the surface of the beating heart. These fixtures were generated by creating a visual/haptic model from preoperative dynamic CT/MR images and registering it using intraoperative ultrasound images. Another characteristic application of dynamic virtual fixtures for cardiac surgeries is that of pulmonary vein isolation using preoperative CT/MR images [14]. The research have paved the way and demonstrated the potential of computer-assisted intraoperative guidance using images, by actively directing the operator for manual or the device for robotic procedures.

Further advancing image-guided and robot-assisted interventions presents many challenges. Among them is the integration of preoperative scans (collected at slow acquisition-speed resulting in high resolution and contrast images) with intraoperative real-time imaging feeds (collected at high acquisition-speed resulting in lower resolution and contrast images). The spatial and temporal registration of preoperative and intraoperative images is not a trivial task and is in particularly challenging in the case of procedures on the beating heart [17]: the deformations secondary to breathing and heart beating can be substantially different in the preoperative and intraoperative image sets. Likewise, tissue deformations due to the interventional tool are absent in the preoperative images. When different modalities are used, an additional complexity arises from the differential contrast, image quality, depiction or lack of the same anatomical landmarks to facilitate registration [18]. In principal, this challenge can be simplified when the same modality is used for both preoperative scan and intraoperative guidance. This has an additional logistic benefit from the clinical point of view: performing both preoperative assessment and intervention during the same session. The single *single-stop-shop* approach has been one of the primary driving forces behind the development of interventional MRI. Indeed, with continuous advancements in real-time MRI (rtMRI), numerous groundbreaking research has demonstrated the potential of the modality to guide interventions including robot-assisted cardiac procedures [19], [20], [21]. A second, pertinent challenge is the immersion of the operator to the information available from the intraoperative images for guiding the maneuvering of the interventional tool.

Within this context, this work focuses on the development and assessment of an approach for extracting information from intraoperative rtMRI and rendering it on-the-fly to the operator via a visual interface (VI) combined with a force-feedback interface (FFI). In particular, it describes a computational framework, shown in Fig. 1, that integrates the hardware (imaging modality, robotic manipulator and the VI and/or FFI) and software (image processing, semi-autonomous or manual robot control, and visual-force-feedback rendering algorithms). To achieve this, the proposed framework, integrates rtMRI with the robot control and uses the image information during

the procedure to enhance the visual perception of the AoP and the control of robotic manipulator.

The concept of generating virtual fixtures on-the-fly from rtMRI was presented in our previous work [22]. The primary feature of this earlier work was the control of the motion of the end-effector of a straight manipulator along a dynamic curve that linked the anatomical entrance with the targeted tissue site. Using this earlier methodology, it was recognized that a generalized approach could be implemented. First, the motion of the manipulator can be constrained within a forbidden region in the form of 4D corridors representing dynamic tissue boundaries that are extracted from rtMRI. Second, a control for a generic bendable tubular manipulator could be implemented. Conceptually, the control of the entire body of a bendable robot, as compared to controlling only the tip of a straight robotic manipulator, provides improved maneuverability and accessibility in the moving tissue. Therefore, in this work the framework is generalized to control the entire body of a bendable manipulator to safely maneuver inside a 4D corridor, from entrance site to targeted anatomy, as compared to controlling only the tip of a straight robot to advance along a dynamic curve [22]. Herein, we also show that the forbidden region fixture can be tuned to behave as a guidance fixture.

While the described framework can be used for a wide range of clinical paradigms, in this work it is investigated for trans-apical aortic valve implantation (TA-AVI); a procedure that has been the subject of MRI-guided manual or robot-assisted studies before [19], [20], [21]. This paradigm was selected as a representative case of a dynamic AoP, since it involves maneuvering a robotic manipulator inside a dynamically changing environment, i.e. that of beating left-ventricle (LV), and positioning it into an also moving target, i.e. the aortic annulus, where a prosthetic valve is placed [23]. For this work, the safety criterion was set for maneuvering the manipulator so that no part would collide with the endocardium and papillary muscles while steering away from the mitral valve. Section II-A describes the overview of the framework and its interaction with the hardware, and sections II-B to II-F describes the computational implements of the framework in form of parallel running threads.

II. METHODS

A. Overview

Figure 1 shows the generic architecture of the framework for man-in-the-loop control of a robotic system using intraoperative rtMRI. The framework is a pipeline that processes rtMRI data, generates and updates on-the-fly a dynamic virtual model comprising of spatiotemporal information of the AoP and renders the information to the operator via VI and FFI interfaces. Data processing and information rendering are tailored to direct the operator for safe maneuvering of a robotic device inside a dynamically changing AoP and to accurately reach the targeted anatomy. As shown in Fig. 1, all processing is performed in the multi-module *processing core* that consists of task-dedicated parallel-running threads, connected to each other as well as with the physical entities (MR scanner, robot, and operator's interfaces). Specifically, the three primary

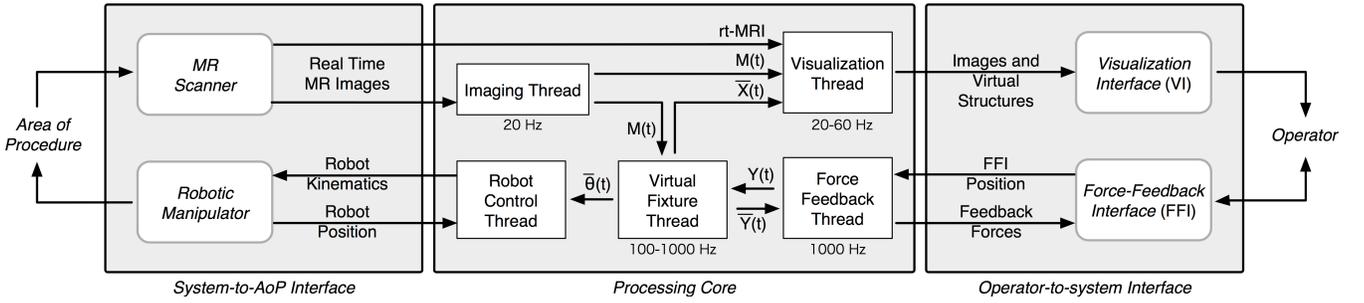


Fig. 1. Generic architecture of the framework illustrating the interconnection of its three primary components: the interface of the system to the AoP, i.e. MR scanner and robotic manipulator, its processing core, and the interface of the operator to the system, i.e. the VI and the FFI. The processing core consists of parallel running dedicated threads (imaging, visualization, virtual fixtures, robot control and force-feedback).

components of the system and the interconnection of the dedicated threads are:

1) *System-to-AoP Interface*: All information needed are extracted from fast MR Imaging that collects in an interleaved fashion a limited number of non-triggered and oblique-to-each-other slices to monitor the motion of endocardial structures and landmarks of interest. In our particular studies, we collected three slices with a speed of 50 ms per slice. While MRI offers the above mentioned benefits, it also brings an important challenge for intra-operative guidance: the limited speed of data acquisition (in our case 150 ms per 3 slices). This inherent-to-the-modality limitation has also determined specific aspects of this framework since its different modules run or require running at different speeds (as shown in Fig. 1). This issue was also addressed by the multi-thread approach. In addition to the MR scanner, the system-to-AoP interface also includes a robotic manipulator that is controlled by the processing core.

2) *Processing Core*: As MR images are collected, they are transferred on-line to the imaging thread where they are processed on-the-fly to generate a dynamic model $M(t)$ of the AoP. The $M(t)$ is represented in the form of a virtual entity, the ‘access corridor’, that is a 4D tubular-like structure extending from the site of entrance to the targeted anatomy. Within the access corridor a robotic manipulator can maneuver safely (i.e. without collision and harming vital structures or healthy tissue). The $M(t)$ is then sent to both the visualization and the virtual fixture threads. The visualization thread renders the dynamic model $M(t)$ superimposed with the MR images on the VI. The virtual fixture thread performs two tasks. The first is to evaluate whether the robot-maneuvering commands $Y(t)$ entered from the FFI (via the force-feedback thread) comply with the safety criteria, i.e. intended kinematic structure of the robot does not collide with any anatomical structure. The second task is to generate appropriate instructions $\bar{\theta}(t)$ and $\bar{Y}(t)$ for the robot control and force feedback threads, respectively. The purpose of the force feedback thread is to compute and render feedback-forces on the FFI to inform the operator about constraints of the access corridor for man-in-the-loop maneuvering of the robotic manipulator.

3) *Operator-to-system Interface*: Interfacing of the operator to the system is performed by means of the VI and FFI. The VI displays the output of the visualization thread, i.e. MR images along with the computed virtual structures, on a high-

definition LCD screen. Via a two way communication, the FFI provides information in form of feedback forces and receives intended control instructions.

B. Imaging Thread

The task of the imaging thread is to generate on-the-fly the dynamic model $M(t)$ of the AoP. This entails processing the MR data in three sequential steps that perform: (1) tracking of the anatomical structures of interest (endocardium and aorta) from MR images in the form of boundary points, (2) from those boundary points, generate control curves in the form of dynamic traces along the tissue boundaries for guiding the procedure and (3) from the control curves generate dynamic 3D access corridors. The input of the imaging thread is a continuous feed of rtMRI composed of N_{SL} number of oblique-to-each-other slices that are collected without cardiac triggering in an interleaved fashion, i.e. $1-2-\dots N_{SL}-1-2-\dots N_{SL}-\dots$ etc [24]. Since MR is rather slow (50 ms/image), only a limited number of slices is selected to balance the needs for assessing the motion of tissue in 3D while maintaining a high speed of data collection. Fig. 2 shows representative results of each step that is discussed below.

1) *Boundary Points from rtMRI*: In this work, as a compromise between need of speed and imaging of a 3D structure, i.e. the LV, we used a $N_{SL} = 3$ achieving a refreshing rate of 50 ms per frame. The panel of images in Fig. 2a shows one time frame of the rtMRI feed illustrating the three oblique-to-each-other slices $I_k(t)$ ($k = 1, 2, 3$) prescribed by the operator to image the LV and aorta to maximize the spatial content of tissue-tracking information of the AoP. The rtMRI feed is processed as we described before [24]. In brief, as shown in Fig. 2a, to accelerate the processing of rtMRI, instead of conventional segmentation of 2D MR images, the algorithm processes the signal intensity of projections of bands that transverse the LV blood pool. Those bands are also selected by the operator pre-operatively. The algorithm segments the 1D profiles of the signal intensity in the projection bands calculating the boundary points $X_{i,j}(t)$ that correspond to tissue boundaries between the high intensity LV blood pool and hypointense endocardium, in the LV, and the aortic annulus. It is noted an important feature of MRI that the boundary point $X_{i,j}(t)$, where $X_{i,j}(t) \in R^3$, is calculated relative to the coordinate system of the MR scanner; together with intra-operative MR images, all other physical (i.e. robot) or virtual

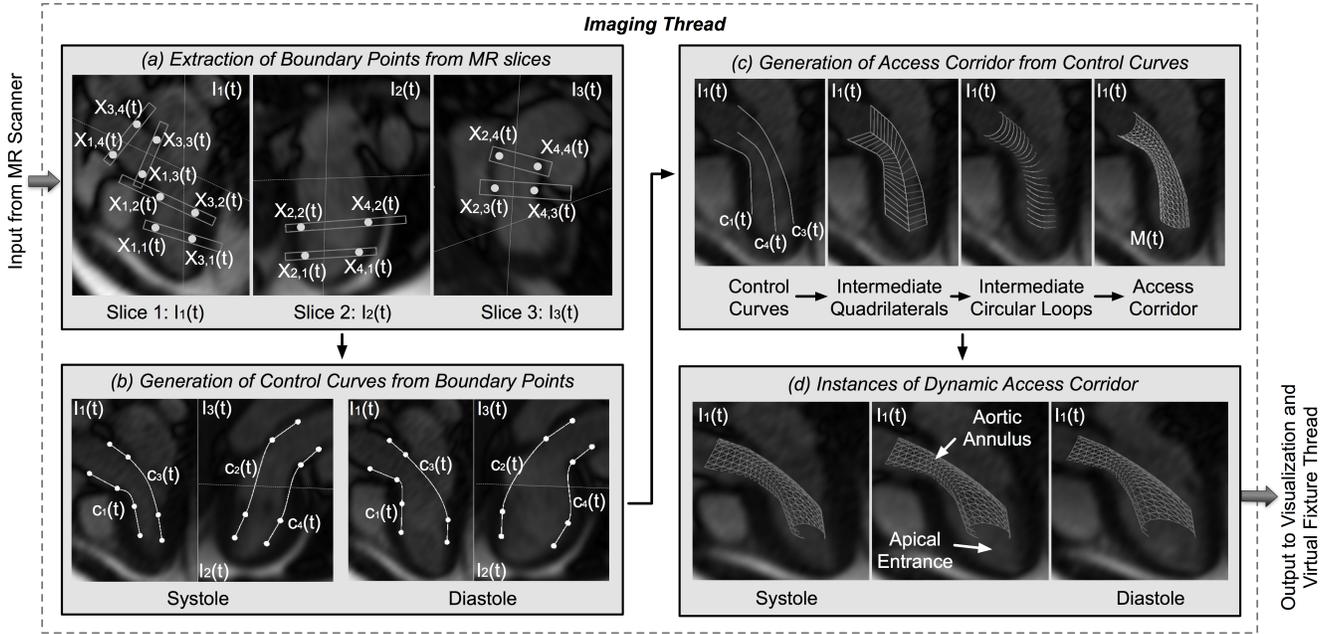


Fig. 2. Example operation of the imaging thread for the generation of the dynamic access corridor from the real-time MR images.

(i.e. access corridors, virtual robot) entities are also relative to this coordinate system.

2) *Control Curves from Boundary Points*: In this step, the extracted boundary points are interconnected with splines to generate dynamic control curves $c_i(t)$; Fig. 2b illustrates examples of the four control curves $c_1(t)$ to $c_4(t)$ superimposed to the three slices. The control curves provide a dynamic guidance trace along the tissue boundaries starting from the site of entrance (i.e. apex) to the target (aortic annulus). The points of the control curves are generated by interpolation of the neighboring boundary points in two steps. First, the points are linearly interpolated if there is no tissue along the line segment connecting the two boundary points. Otherwise, Kochanek-Bartels curves [25] are used to interpolate the boundary points based on the shape of the tissue boundary. The control curves are generated so that the number of interpolated points remains the same on each curve and frame. In the particular case of TA-AVI, the boundary of the endocardium of LV is generated by linear interpolation between the points $X_{i,1}(t)$ and $X_{i,2}(t)$, where $i = 1$ to 4, on slices $I_1(t)$ and $I_2(t)$. The tissue boundary inside the aortic root is generated by linear interpolation between the boundary points $X_{i,3}(t)$ and $X_{i,4}(t)$ on slices $I_2(t)$ and $I_3(t)$. Small bulges (less than 1mm) along the surface of the tissue (such as trabeculations towards the bottom of the septum) are ignored at this step. The region between the point $X_{i,2}(t)$ and $X_{i,3}(t)$ is interpolated using Kochanek-Bartels curves [25]. The generated curves $c_1(t)$ and $c_3(t)$ are on the imaging plane of slice $I_1(t)$, while the curves $c_2(t)$ and $c_4(t)$ are on the imaging plane of slices $I_2(t)$ to $I_3(t)$. The graphical user interface of the VI also allows the operator to manually alter the tangential properties of these curves. The operator can adjust these properties on-the-fly as desired with the mouse and visual inspection. In our experimental studies, we observed that the operators adjusted

the shape of the curves to better account for the specific anatomy of the patient at the AoP (usually once during a study). The control curves were most often deflected more towards the interventricular septum, relative to the mitral valve and papillary muscles.

3) *Access Corridors from Control Curves*: In this step, access corridors are generated to control curves through the series of processes, as example shown Fig. 2c. First the interpolated points from each curve are interconnected to form intermediate quadrilaterals, which are further subdivided to form intermediate loops [26]. The code offers the options that the points on the intermediate loops be interconnected and form either a triangular or a quadrilateral mesh. The subdivision in this step causes narrowing of the mesh along the tissue boundary. This makes the mesh conservative in nature, by avoiding small bulges (less than 1 mm) along the tissue boundaries. As the boundary points are dynamic, adhering to the motion of tissue, the dynamic access corridor inherits this property and thus the corridor also follows the heartbeat, breathing motion, and any deformation caused by tool-tissue interaction. The panel in Fig. 2 shows representative frames (from the VI display) of three selected instances in the cardiac cycle depicting the 3D access corridor superimposed onto one of the three rtMRI slices.

In our implementation, the three computational steps of the imaging thread are applied to each slice as it is acquired and fed to the thread. It operates in a moving-window fashion and the thread does not wait for the collection of a new set of three slices to process them. As soon as a new image arrives it is processed together with the previous two slices and the access corridor is updated within 0.50 ms. As a result a frame of the access corridor is computed from three images: one 50 ms old, one 100 ms old, and one 150 ms old. It is noted that, previous studies have demonstrated that the motion of

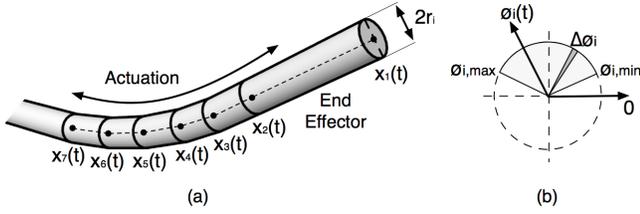


Fig. 3. (a) Tubular structure representing a bendable robotic manipulator. $\mathbf{X}(t) = (\mathbf{x}_i(t); i = 1, 2 \dots N)$ and radius is r_i . The actuation mechanism for controlling the end effector is located outside the patient body. (b) Definitions of the i^{th} actuator substituting a rotational angle $\phi_i(t)$, where $\phi_{i,min} \leq \phi_i(t) \leq \phi_{i,max}$.

the boundary points extracted from the feeds collected with 50 ms or 150 ms per image show high correlation [24]. As the thread processing time (0.50 ms per image) is much less than the image acquisition time (50 ms per image), the effective frequency of this thread is assigned to be the same with the rtMRI frequency, i.e. 20Hz.

C. Virtual Fixture Thread

To implement the operation of virtual fixture thread we (1) used the paradigm of a bendable tubular manipulator (Fig. 3a) and (2) adopted a scheme to describe the operator commands and the robot actuation. The selected structure in Fig. 3a is a generalized representation of manipulators used or under development for a wide range of procedures, including but not limited to endo-cardiac, vascular and natural orifice transluminal endoscopic surgery ([8], [27], [9]). We note that the presented analysis refers to the part of the manipulator that is actuated to perform a bend. In the current examples, this corresponds to a certain length of the distal portion of the manipulator. We treat only this portion of the manipulator, i.e. the inserted length of the tool from where the actuation starts; the remaining part is controlled by some other means. Such a structure can be discretized in N cylindrical elements interconnected with links, as shown in Fig. 3a. Considering that the robot is registered to the MR scanner [8], its structure can be described with the array $\mathbf{X}(t)$ of vectors such that $\mathbf{X}(t) = (\mathbf{x}_i(t) \in R^3; i = 1 \text{ to } N)$, where the vector $\mathbf{x}_i(t)$ is the coordinates of the i^{th} link in the coordinate system of the MR scanner. Herein, we will refer to the $\mathbf{X}(t)$ structure as an element of the set *robotStates*.

If the links of the robotic manipulator are actuated by M actuators, then the state of these actuators can be represented with the array $\theta(t) = (\phi_i(t); i = 1 \text{ to } M)$, where $\phi_i(t)$ is the rotational angle of the i^{th} actuator (Fig. 3b). We will refer to $\theta(t)$ as an element of the set *actuatorStates*. Rotation of an actuator is discretized using small rotational step $\Delta\phi_i$; this step size is equal to the desired minimum accuracy of the actuator for the procedure. The rotation angle of an actuator $\phi_i(t)$ is usually limited by the physical constraints imposed by the design of the robotic manipulator, i.e. the actuator can rotate between $\phi_{i,min} \leq \phi_i(t) \leq \phi_{i,max}$ (Fig. 3b). Thus the number of *actuatorStates* is given by $\prod((\phi_{i,max} - \phi_{i,min})/\Delta\phi_i)$, where $i = 1 \text{ to } M$. Corresponding to each actuator state $\theta(t)$, the state of the robotic manipulator $\mathbf{X}(t)$ is computed and

Algorithm 1 Virtual Fixture Thread

input: $\mathbf{M}(t), \mathbf{Y}(t)$
output: $\bar{\theta}(t), \bar{\mathbf{X}}(t), \bar{\mathbf{Y}}(t)$

- 1: **if** interventional environment changes **then**
- 2: $collisionResolved = true$
- 3: **end if**
- 4: **if** the position $\mathbf{X}(t) = f_{RA}(f_{AC}(\mathbf{Y}(t)))$ of the robotic manipulator collides with dynamic access corridor $\mathbf{M}(t)$ **then**
- 5: **if** $collisionResolved$ **then**
- 6: Perform collision response to compute a suitable position $\bar{\mathbf{X}}(t)$ for the robotic manipulator and its corresponding actuator state $\bar{\theta}(t)$.
 Compute $\bar{\mathbf{Y}}(t) = f_{AC}^{-1}(\bar{\theta}(t))$.
 $collisionResolved = false$.
- 7: **else**
- 8: Set $\bar{\theta}(t), \bar{\mathbf{X}}(t), \bar{\mathbf{Y}}(t)$ to the values computed in previous iteration.
- 9: **end if**
- 10: **else**
- 11: Allows the robotic manipulator to move in the operator-defined direction. Set $\bar{\mathbf{Y}}(t) = \mathbf{Y}(t)$.
 Compute $\bar{\theta}(t) = f_{AC}(\bar{\mathbf{Y}}(t))$ and $\bar{\mathbf{X}}(t) = f_{RA}(\bar{\theta}(t))$.
 $collisionResolved = true$.
- 12: **end if**

stored in *robotStates*.

There is an one-to-one correspondence, described by the bijective function f_{RA} , between the structure $\mathbf{X}(t) \in robotStates$ of the robotic manipulator and the state $\theta(t) \in actuatorStates$ of the actuators, i.e. $f_{RA} : robotStates \rightarrow actuatorStates$. The exact expression of the bijective function f_{RA} depends upon the design of the specific kinematic structure of the robotic manipulator. Via the FFI, the operator requests a particular motion described by a set of instructions $\mathbf{Y}(t)$ that we refer to as the *commandStates*. A second bijection function f_{AC} can then be defined to describe the one-to-one correspondence between the state of the FFI $\mathbf{Y}(t) \in commandStates$ and the state of actuators $\theta(t) \in actuatorStates$, i.e. $f_{AC} : actuatorStates \rightarrow commandStates$.

The virtual fixture thread operates as described in the panel of Algorithm 1. It first detects whether there will be a collision of the robotic manipulator $\mathbf{X}(t)$ with the AoP (represented by dynamic access corridor $\mathbf{M}(t)$) using its ‘collision detection’ routine (panel Algorithm 2). If a collision is detected, then the thread resolves it by computing a safe location for the robotic manipulator to move. The computation of the new safe location is performed by the ‘collision response’ routine (panel Algorithm 3). In case of no collision, the thread allows the robotic manipulator to move in the operator-defined direction. The outputs $\bar{\theta}(t), \bar{\mathbf{X}}(t)$, and $\bar{\mathbf{Y}}(t)$ of the virtual fixture thread are then used by the robot control thread (section II-F), visualization thread (section II-E), and force-feedback thread (section II-D), respectively. The algorithms for collision detection and collision response are implemented using parallel processing and are described as follows:

Algorithm 2 Collision Detection

input: $\mathbf{M}(t)$, $\mathbf{X}(t)$
output: *collision*

- 1: *collision* = false
- 2: **for each** vertex \mathbf{v} of mesh $\mathbf{M}(t)$ representing dynamic access corridor **in parallel**
- 3: **for each** cylindrical segment $(\mathbf{x}_i, \mathbf{x}_{i+1})$ of robotic manipulator in state $\mathbf{X}(t)$ **in parallel**
- 4: **if** shortest distance between the vertex \mathbf{v} and the line segment $(\mathbf{x}_i, \mathbf{x}_{i+1})$ is less than r_i **then**
- 5: *collision* = true
- 6: **end if**
- 7: **end for**
- 8: **end for**

1) *Collision Detection Routine:* This algorithm evaluates whether anyone of the elements of the entire body of the robotic manipulator may collide with the access corridor.

2) *Collision Response Routine:* The algorithm of this subroutine first computes all possible positions, which the robotic manipulator can achieve inside the AoP for a given insertion length during the procedure. An objective function f_1 is used to determine the optimal position for the robotic manipulator from all the computed positions. In this work, we set $f_1(\mathbf{X}(t); \mathbf{Y}(t))$ equal to the distance between the tip of the end effectors corresponding to the robot state $f_{RA}(f_{AC}(\mathbf{Y}(t)))$ desired by the operator and a element $\mathbf{X}(t)$ of *robotStates*.

At the beginning of the execution of virtual fixture thread, the state *collisionResolved* of Algorithm 1 is set to *true*, and once a collision is detected its value changes to *false* until $\bar{\mathbf{X}}(t)$ converges to $\mathbf{X}(t)$ (i.e. in the presence of feedback forces generated on the FFI). The *collisionResolved* becomes *true* when the collision is resolved. This ensures stability of the framework as no new position $\bar{\mathbf{X}}(t)$ is computed when its converging to a safe position. If the interventional environment changes (i.e. either the insertion length of the tool changes or a new frame of access corridor is received from the imaging thread) during this transition *collisionResolved* is again set to *true* to compute the new position $\bar{\mathbf{X}}(t)$ to reflect the change.

D. Force-Feedback Thread

The force feedback thread acts as a link between the computational core and the hardware of the FFI. It reads the encoder readings from the motors of the FFI and creates a command state $\mathbf{Y}(t) = (\lambda_i(t); i = 1 \text{ to } N_{FFI})$ where N_{FFI} is number of FFI motors), where $\lambda_i(t)$ is the rotational angle of the i^{th} motor of the FFI. The command state is then sent to the virtual fixture thread. After processing it, the virtual fixture calculates the corrected command state $\bar{\mathbf{Y}}(t) = (\bar{\lambda}_i(t); i = 1 \text{ to } N_{FFI})$ and send it back to the force-feedback thread. The feedback forces are generated such that the FFI moves from the command state $\mathbf{Y}(t)$ to the corrected command state $\bar{\mathbf{Y}}(t)$. Feedback-forces are generated in form of torques on each i^{th} motor by applying a virtual spring-damper on the corrected position $\bar{\lambda}_i(t)$ and the one commanded by the operator $\lambda_i(t)$. It should be noted that, when there is no collision, i.e. $\bar{\mathbf{Y}}(t)$

Algorithm 3 Collision Response

input: $\mathbf{M}(t)$, $\mathbf{Y}(t)$
output: $\bar{\mathbf{X}}(t)$

- 1: **for each** state $\theta(t)$ of actuators in *actuatorStates* **in parallel**
- 2: Compute the state $\mathbf{X}(t) = f_{RA}(\theta(t))$ of the robotic manipulator and store it in *robotStates*.
- 3: **end for**
- 4: **for each** state $\mathbf{X}(t)$ in *robotStates* **in parallel**
- 5: **if** robotic manipulator $\mathbf{X}(t)$ collides with dynamic access corridor $\mathbf{M}(t)$ **then**
- 6: Set the function $f_1(\mathbf{X}(t), \mathbf{Y}(t)) = \infty$
- 7: **else**
- 8: Compute the function $f_1(\mathbf{X}(t), \mathbf{Y}(t))$
- 9: **end if**
- 10: **end for**
- 11: **for each** state $\mathbf{X}(t)$ in *robotStates* **do**
- 12: Find a state $\bar{\mathbf{X}}(t)$ for which $f_1(\mathbf{X}(t), \mathbf{Y}(t))$ is minimum.
- 13: **end for**

$= \mathbf{Y}(t)$, no feedback forces are felt on the FFI. In case of collision, the feedback forces direct the FFI to a position such that the robotic manipulator is inside the safe corridor.

E. Visualization Thread

The visualization thread renders the dynamic model $\mathbf{M}(t)$ and the virtual robotic structure superimposed with the rtMRI. The graphical rendering is implemented by using OpenGL and speed of the thread varies between 40Hz to 80Hz depending upon the number of MR images (1, 2, or 3) being rendered.

F. Robot Control Thread

The robot control thread is used to control the robotic manipulator with M actuators. The thread receives the command $\bar{\theta}(t)$ from the virtual fixture thread and allows manipulator to achieve the configuration $\bar{\mathbf{X}}(t)$, where $\bar{\mathbf{X}}(t) = f_{RA}(\bar{\theta}(t))$.

III. EXPERIMENTAL STUDIES**A. Experimental Setup**

The framework was evaluated for performing simulated MRI-guided TA-AVI using a virtual robotic manipulator. To optimize execution, the different modules of the processing core (Fig. 1) were implemented on a standard PC (Intel 2.4 GHz Processor and 16 GB RAM) with General Purpose Graphics Processing Unit (GPGPU; NVIDIA Tesla C1060 GPU) and an embedded controller board (DS1103 PPC dSPACE with 1GHz CPU, 16A/D, 8D/A and 6 separate encoder channels). Specifically, imaging, visualization and virtual fixture threads were implemented on the PC. The parallel processing for collision detection (Algorithm 2) and collision response (Algorithm 3) of the virtual fixture thread were executed on the GPGPU, using the CUDA parallel computing platform. The force feedback thread was implemented on the

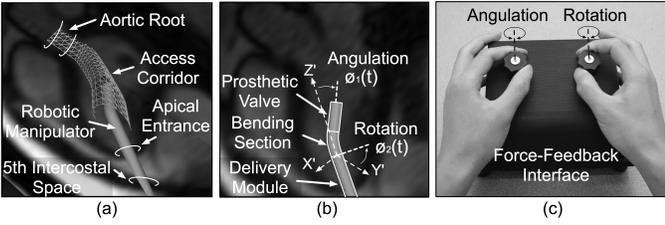


Fig. 4. (a) Schematic illustration of the task. (b) Virtual robotic manipulator. (c) Two degree-of-freedom force-feedback interface.

embedded controller board. The PC and the embedded controller board were physically connected with a fiber-optic cable (100 Mbps). This implementation synchronized acquisition with no observed latencies.

For guiding the simulated procedure, we used rtMRI data sets of 540 frames each composed of three oblique-to-each-other slices (I_1 , I_2 , and I_3) collected without cardiac triggered and with free-breathing using a true fast imaging with steady-state precession (true-FISP) pulse sequence (TR = 49.3 ms; TE = 1 ms; flip angle = 64° ; pixel size = 1.25×1.25 mm²; FOV = 275×400 mm²; slice thickness = 6 mm) on a Siemens 1.5T Avanto MR scanner on three healthy volunteers. To address logistics with the availability of the MR scanner, experiments were performed off-line (i.e. during the experiments, the processing core was not physically connected to the scanner for receiving the rtMRI feed). To mimic on-line conditions, we implemented a virtual MR scanner thread on the PC that supplied the imaging thread with a stream of the above-mentioned real-time MR images every 50 ms (i.e. the exact timing of their collection). The stream was repeated as long as needed to perform the experimental task.

The virtual manipulator used in the study (shown in Fig. 4a and Fig 4b) was generated with a chain of cylindrical meshes and was assigned spatial dimensions relative to the MR scanner coordinate system for realistic fusion with the MR images. The manipulator is an 8-mm diameter cylindrical structure composed of three parts in tandem: a straight 10-cm long section (the delivery module), a 1.5-cm long bending section and a 2 cm long straight section, that represented a prosthetic valve. The dimensions were those of a clinically used tool [8]. The delivery module was inserted through the apex, which acts a pivotal point. At its distal end (i.e. at its interface to the bending section) a coordinate system $X'Y'Z'$ (Fig. 4b) was assigned, as we described previously in [22], for describing the motion of the other two sections of the manipulator. In reference to section II-C, the manipulator was discretized with $N = 4$: the prosthetic valve section with one cylindrical element ($r_1 = 4$ mm and length = 2 cm) and the bending section with 3 cylindrical segments ($r_2 = r_3 = r_4 = 4$ mm and length = 0.5 cm). The manipulator had three DoF: translation $L(t)$, rotation $\phi_2(t)$ ($0^\circ \leq \phi_2(t) \leq 360^\circ$) around the axis of the straight delivery module, and angulation $\phi_1(t)$ ($0^\circ \leq \phi_1(t) \leq 30^\circ$) orthogonal to this axis. Feedback forces were applied to the distal end of the tool (i.e. the bending element and the prosthetic valve section), via the two angular DoF. Therefore, we used two virtual actuators ($M = 2$) with the state of actuation $\theta(t) = \{\phi_1(t), \phi_2(t)\}$

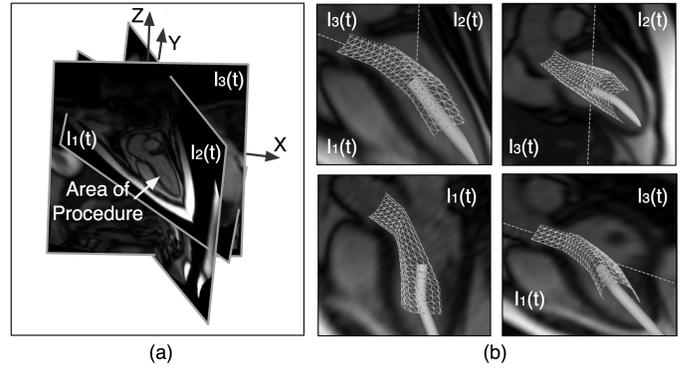


Fig. 5. Rendering of visual information on the VI comprising of (a) the three oblique-to-each-other planes (I_1 , I_2 , and I_3) and (b) a different viewing angle of those planes together with the virtual robotic manipulator and the access corridor.

and step size $\Delta\phi_1 = \Delta\phi_2 = 1^\circ$. The function f_{RA} was used to map the *actuationStates* to the *robotStates* and was implemented such that the distal end rotates around the Z' axis and angulates on a plane orthogonal to $X'Y'$ plane and passing through origin of the distal coordinate system (shown in Fig. 4b).

To control the virtual manipulator we used an FFI composed of two devices. One, was an in-house developed 2-DoF device shown in Fig. 4c, that was used to control the angulation and rotation of the virtual manipulator (Fig. 4b). Each knob was directly connected to a DC motor (Maxon 264571) and the output of its encoder and input of its amplifier were connected to the controller board. To adjust insertion and retraction of the virtual tool we used a pedal controller (Logitech Flight System G940) connected to the dedicated PC (passive without force-feedback). This device was specifically tailored for the desired experiments since it offered force-feedback endowed control of the two DoF that we studied, as well as allowed us to have a one-to-one mapping between the elements of the operator input, represented by the command state $\mathbf{Y}(t)$, and the actuation state, represented by the state $\theta(t)$.

The visual cues were comprised of rtMRI data (i.e. I_1 , I_2 , and I_3), the access corridor ($\mathbf{M}(t)$), and the structure of the virtual robotic manipulator ($\mathbf{X}(t)$ or $\bar{\mathbf{X}}(t)$) that were graphically rendered on the high-definition LCD of VI by the visualization thread using OpenGL. All rendered objects were coregistered to the MR coordinate systems using the slice orientation information extracted on-the-fly (i.e. as individual images were fed to the imaging thread) from the DICOM header of the rtMRI images (as shown in Fig. 5a). The VI offers the user-selectable options of multiple displays of the oblique MR images with different orientation and/or combination of slices (Fig. 5b) for operator-customized visualization of the 3D environment.

B. Task

The subjects (five non-medical graduate students) were first introduced to the operation of the system through a 30-minute preparatory session that included familiarization with the anatomy of the AoP, the motion of the virtual manipulator,

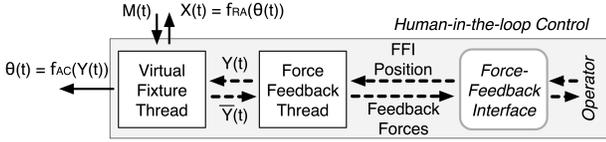


Fig. 6. Operation in mode III illustrating human-in-the-loop control.

and hand-eye coordination training with the VI/FFI of the system. Subsequently, each subject was asked to use the VI/FFI to maneuver the virtual manipulator in a simulated TA-AVI procedure (shown in Fig. 4a) that included three steps: (1) insertion of the virtual manipulator from the apical entrance and deployment into the aortic root, (2) holding this position for 2 seconds to simulate deployment of a valve, and (3) retraction of the tool to its initial position. In these studies each subject performed the task with four different modes of operation of the framework:

- *Mode I (MRI)*: The operator only sees the MR feed (either combination of the three slices I_1 , I_2 , and/or I_3) rendered continuously on the VI without any forces exerted by the FFI.
- *Mode II (MRI + Access Corridor)*: The operator sees the MR images and the access corridor $M(t)$ rendered together. The corridor provides the operator with visual cues to maneuver the manipulator inside the dynamic environment of the LV.
- *Mode III (MRI + Access Corridor + Force-Feedback)*: The operator sees the MR images and the access corridor while guided by feedback forces exerted by the FFI. In this mode, the position $\mathbf{X}(t) = f_{RA}(f_{AC}(\mathbf{Y}(t)))$ and the actuation states $\theta(t) = f_{AC}(\mathbf{Y}(t))$ of the manipulator are computed directly from the operator command state $\mathbf{Y}(t)$. In the presence of collision $\bar{\mathbf{Y}}(t)$ is computed, as described in section II-C, i.e. by the virtual fixture thread and sent to force-feedback thread (shown in Fig. 6). Whenever the virtual manipulator is maneuvered to cross the surface of the access corridor, collision is detected and feedback forces are applied by the FFI to bring the virtual manipulator to a safe location within the access corridor.
- *Mode IV (MRI + Access Corridor + Force-Feedback)*: As described in section II-A, the virtual robotic manipulator (visually render as $\bar{\mathbf{X}}(t)$) is always confined within the access corridor. This gives the operator a visual cue that the manipulator stays within the conduit defined by the access corridors. If $\mathbf{X}(t)$ goes outside the tissue boundaries, forces are generated by force feedback thread which bring $\mathbf{X}(t)$ back to safe position $\bar{\mathbf{X}}(t)$.

Each mode was repeated three times for statistical significance (i.e. each subject performed 12 procedures), with 10 minutes idle time between each task and with a random order of performance. During each study, we were collecting a series of data, sampled every 10 ms and stored into a log-file. Its entry was time-stamped and included the following (1) the position of virtual robotic manipulator ($\mathbf{X}(t)$ and $\bar{\mathbf{X}}(t)$), (2) the operator command $\mathbf{Y}(t) = \{\text{Angulation } \phi_1(t), \text{Rotation } \phi_2(t)\}$,

(3) the response to the operator command from the framework $\bar{\mathbf{Y}}(t) = \{\text{Angulation } \bar{\phi}_1(t), \text{Rotation } \bar{\phi}_2(t)\}$, (4) the length $L(t)$ of the straight portion of the manipulator (i.e. from the apical entrance to the distal origin of the distal coordinate system), (5) the state of collision of the virtual manipulator with the access corridor in binary form, i.e. either *true* or *false* and (6) the indices of the presented slices in the real-time stream of MR images. At the conclusion of a study, the operators were interviewed to for quantitative feedback on the system interface.

IV. EXPERIMENTAL RESULTS

The log files were first analyzed to extract measures related to the performance of the procedure. Specifically, for each operator and each mode of operation, the following values were calculated: (1) *task completion time*, i.e. durations for performing the entire task and its three steps (insertion, deployment, and retraction) and (2) *impingement distance*, i.e. the distance the distal end of the manipulator penetrates beyond the surface of the access corridor, and (3) *collision time*, i.e. the duration the distal end of the manipulator was at a state of collision with the access corridor. The percentage of time spent at different impinge distances during collision was calculated, with respect to total collision time. For this purpose, all recorded impingement distances of a trial were categorized into five bins each 2 mm wide (equal to multiple of pixel size of the real-time MR images). The results are reported as mean and standard deviation (SD) per individual corresponding to 3 trial repetitions per mode and as group corresponding to 15 trials per mode, i.e. 5 subjects in the group with 3 trials per subject.

Table I summarizes the results of the collected experimental data for all five subjects. The table is organized for comparison between the four modes of operation, reporting the subjects both as individuals and as a group. When considering the individual subjects, there is a consistent reduction of the duration for performing the task when inspecting the rows from mode I to mode IV (with the exception of Subject #2 in modes II and III). The same trend is also seen at the group level that manifests a reduction of the task completion time with 18.51 ± 5.78 for mode I, 15.23 ± 6.22 for mode II, 10.22 ± 1.81 for mode III, and 7.67 ± 1.75 for mode IV. As we may expect, as the information content rendered to the operator increases (i.e. from mode I to mode IV), the subject requires an ever decreasing time to perform the task. Moreover, and consistent with the previous observation is that the SD of the task completion time for the group shows a similar trend: the SD for modes III (1.81 s) and IV (1.75 s) are approximately 60% less that those for modes I (5.78 s) and II (6.22 s). The convergence of SD, support the notion that, in the presence of force-feedback (i.e. modes III and IV), all subjects required nearly same amount of time to perform the task as compared with absence of force-feedback. We used Analysis of Variance (ANOVA) to determine if there was a significant difference between the task completion time of the modes while controlling for variability that may have been caused by the different trials. Results of the ANOVA showed

TABLE I
ANALYSIS OF THE EXPERIMENTAL RESULTS.

<i>Modes of Operation of the Framework</i>		Mode I		Mode II		Mode III		Mode IV	
Visualization of Access Corridor on VI (On / Off)		Off		On		On		On	
Rendering of Feedback Forces on FFI (On / Off)		Off		Off		On		On	
Robotic Manipulator Representation ($\mathbf{X}(t)$ / $\bar{\mathbf{X}}(t)$)		$\mathbf{X}(t)$		$\mathbf{X}(t)$		$\mathbf{X}(t)$		$\bar{\mathbf{X}}(t)$	
Average task completion time (s)	Subject #	Mean	SD	Mean	SD	Mean	SD	Mean	SD
	#1	19.38	1.16	18.10	0.60	11.40	0.65	6.59	1.11
	#2	13.49	5.94	9.41	2.13	10.11	1.23	8.92	0.70
	#3	23.10	6.01	21.34	11.37	11.78	0.63	9.68	1.14
	#4	20.94	8.34	14.05	2.23	10.62	1.02	5.96	1.82
	#5	15.65	0.47	13.23	2.12	7.20	0.27	7.18	0.43
	Group	18.51	5.78	15.23	6.22	10.22	1.81	7.67	1.75
Percent of residence duration during collision (%)	0 mm to 2 mm	51.25		50.75		86.53		0.00	
	2 mm to 4 mm	29.38		32.80		12.52		0.00	
	4 mm to 6 mm	13.25		12.93		0.95		0.00	
	6 mm to 8 mm	4.83		2.62		0.00		0.00	
	More than 8 mm	1.29		0.90		0.00		0.00	

a significant difference ($p < 0.0001$) between modes, while the effect of trial was not significant ($p = 0.0865$). A Tukeys pairwise post hoc comparison further showed that modes I and II differed significantly from mode III and mode IV in mean time to complete the procedure.

In reference to the residence time over the impingement distances (reported in the lower section of Table I), we observe that for all modes more than 80% of time the tool was within 4 mm from the access corridor, while this time was shifted toward smaller values when moving from mode I to mode IV. We also observe that in the presence of only visual cues (modes I and II), there is no significant improvement for mode II as compared to mode I. This maybe related to the fact (identified by interviewing the subjects) that the operator is not able to perceive the depth of the virtual manipulator with respect to the MR image projected parallel onto the VI screen. During those studies, it became evident that the dynamic nature of the environment does not allow the operator to perceive depth information at the same time from multiple views (Fig. 5b). This further underscores the benefit of force-feedback for telemanipulation tasks in 3D space in which depth cannot be visually perceived. In the presence of force-feedback, mode III, the percent of residence duration for shorter proximity increases drastically (i.e. for the [0 to 2 mm] bin from 50.75% to 86.53% with a concomitant result for the [2 to 4 mm] from 32.80% to 12.52%, and for bins with >4 mm from 15.55% to 0.95%). For this experimental set up and group of subjects, those data support the notion that an intervention could safely performed with mode III when the tissue is more than 4 mm away from the access corridor (as the tool would stay away from the tissue for 99% of time). Obviously, the access corridor can be narrowed by the operator (or the software) by moving the vertices of mesh $\mathbf{M}(t)$ opposite to the direction of the vertex normal. If the corridor is further narrowed down to the radius of the manipulator, mode III behaves as a guidance virtual fixture. Notably, in mode IV during collision, the virtual manipulator (represented by $\bar{\mathbf{X}}(t)$) always stays inside the access corridor.

Additional analysis was performed to evaluate the framework in delivering the performance it was designed for and

in regard whether: (1) the access corridor ever collided with the endocardium, (2) the virtual manipulator was confined within the access corridor, and (3) the FFI exerted forces when there was collision. We performed this analysis manually, i.e. by loading the log file back in memory and playing-back each repetition of the task, and visually inspecting each frame. This was a tedious procedure that was repeated by three volunteers and deemed necessary for ensuring basic functionality of the framework. Fig. 7 illustrates representative results from a mode IV playback. The panel in Fig. 7b shows five frames of the recorded stream that correspond to insertion (t_1 and t_2), deployment (t_3) and retraction (t_4 and t_5). It is noted that in those frames are observable the deformation of the access corridor, secondary to both heart beating (t_3 corresponds to systole and t_5 to diastole) and free breathing (at instance t_1 , the apex is at a lower position as compare to t_3). This analysis verified the intended operation of the framework as summarized in the following qualitative results: (1) in the absence of collision, the virtual manipulator always followed the operator-commanded maneuver and no feedback forces were felt on the FFI ($\phi_1(t) - \bar{\phi}_1(t) = 0$ and $\phi_2(t) - \bar{\phi}_2(t) = 0$), (2) in the presence of collision, feedback forces always exerted, while resulting to the FFI moving from state $\mathbf{Y}(t)$ to state $\bar{\mathbf{Y}}(t)$, and (3) for all time instances, the access corridor never collided with the endocardium and aortic root, while the virtual manipulator ($\bar{\mathbf{X}}(t)$) stayed within the access corridor. As a result, the described framework ensured that the manipulator never touched the tissue.

V. DISCUSSION

This work describes a system for performing interventions with real-time MRI guidance. Viewed as the first step toward a generalized framework, the architecture of the computational core (shown in Fig. 1 and described in section II) aim to integrate seamlessly physical and cyber components: (i) the former being the imaging scanner, the robotic manipulator and the VI and/or FFI and (ii) the latter being the dedicated algorithms for image processing, image-extracted dynamic virtual fixtures, semi-autonomous or manual robot control, and visual-force-feedback rendering. To achieve this

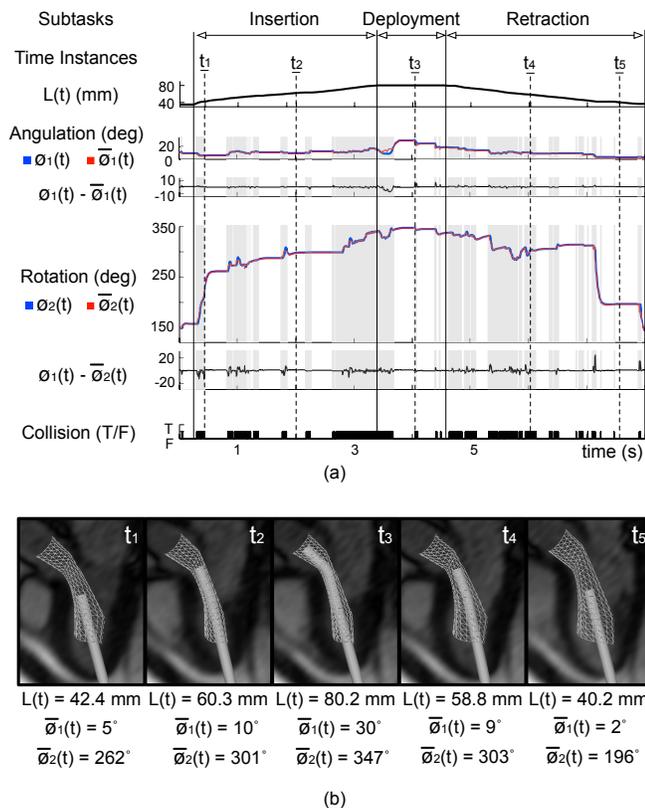


Fig. 7. (a) Representative results of the recorded stream of the simulated surgical task of transapical aortic valve implantation under mode IV of the framework. (b) Five selected frames from the recorded stream.

aim, approaches were introduced in regard to three specific aspects of this framework: (1) the relatively low speed of MR data collection (20 Hz), (2) the high refreshing frequency needed for a force-feedback interface (1000 Hz), and (3) the computationally intensive image-based control of a multi-DoF robotic manipulator.

Those three aspects, together with the necessity of a system that facilitates on-the-fly operation, resulted to adopting a multi-thread approach. As demonstrated by the experimental studies, this approach was suitable for the above-mentioned purposes since the generated dynamic visual and force-feedback cues resulted to guidance with reduced (mode III) or eliminated (mode IV) collisions. The framework in Fig. 1 was designed to be modular so it can be adapted for use with different modalities, robotic manipulators or FFI. In this work, the computational core was tailored for guidance with rtMRI by appropriately implementing the imaging thread. MRI was pursued since it offers certain unique features for real-time image guidance and in particular for robot-assisted cardiac procedures [19], [20], [21]. Among those features that make MRI suitable for guiding interventions are (1) a plethora of soft tissue contrast mechanism for assessing tissue morphology and function, (2) absence of ionizing radiation allowing unlimited imaging, (3) an inherent coordinate system, relative to which interventional tools and manipulators can be registered and tracked, (4) computer-controlled setting of the imaging planes, including oblique orientations and 3D volumes and (5) the

ability to change on on-the-fly the parameters of MR data acquisition, such as the orientation of the imaging plane(s) and acquisition parameters directly from the control of the manipulator [28].

However, MRI has certain limitations; in addition to the safety and patient accessibility [29], a critical limitation directly pertinent to intraoperative guidance is the inherent-to-the-modality low signal sensitivity that limits the speed of rtMRI. Indeed, current rates are in the range of 30-50 ms per image [21]. This presented a challenge and to address it, we adopted the use of a limited number of planes that were prescribed by the operator to image specific areas related to the procedure (Section II-B). We used an improved approach of a moving window first described in [24]. This was further modified to generate access corridors in form of parameterized surfaces to be used in collision detection (Algorithm 2).

The presented work has certain limitations, which we feel do not limit the functionality of the described approach. First, the majority of the studies were performed off-line; i.e. the processing core was not connected on-line with the MR scanner. At this developmental stage, this was deemed necessary to provide adequate time for assessing and fine-tuning the system without the logistic challenges associated with having volunteers in a heavily booked clinical MR scanner. To ensure accurate timing and realism, the previously collected rtMRI feeds were supplied to the processing core from a virtual MR scanner that fed the imaging thread with one image every 50 ms (i.e. the time needed to collect and transport image). With this implementation, performing the studies off-line did not limit the operation of the described framework. Second, the positions and orientations of the slices and projection bands were selected pre-operatively manually from CINE images, as also described before [24]. Our pilot studies, as well as prior work [28], indicate that a more efficient approach would be to automatically adjust the imaging planes and the bands on-the-fly. This is one of the future directions of this work, i.e. to implement a manipulator-driven real-time MR guidance with which the robot control thread also controls on-the-fly the MR scanner to adjust the acquisition parameters as we described in [28]. Third, the work did not include an actual robotic manipulator or a specific FFI, and the computation of the control commands assumed a generic actuation unit. When an actual robot and FFI are used, then their corresponding kinematic structures need to be entered into the modules of the robot control and force-feedback threads. In this work the motion of the robot was assumed to be instantaneous. The code can be expanded to account for the response frequency, actuation delays and kinematic constrains of the robot for computing the stability and transparency of the framework. However, it is noted that, this will not affect the computation of fixtures from the MR images. Fourth, the inclusion of a robotic manipulator inside an MR scanner may distort the images and hence the signal intensity projections. In such a scenario, tool-image enhancement techniques, like those proposed by Guttman et al. [21], could be applied. Alternatively, the robotic manipulator can tracked inside the MR scanner using fiducial RF coils [30], while signal distortions that may induce can be suppressed while collecting the image.

VI. CONCLUSION

Concluding, this work describes a framework for real-time MRI-guided minimally-invasive robot-assisted procedures that was demonstrated for a simulated transapical valvuloplasty on the beating heart. The framework integrates two separate channel of information flow: the real-time visualization of the AoP and telemanipulated control of the robotic manipulator. This integration of real-time imaging, robotic manipulators and visual/force-feedback interfacing may further enhance the armamentarium of methods for enhancing current and developing new interventional procedures.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) award CNS-0932272. All opinions, findings, conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of our sponsors. We would like to thank Karen Chin of the Methodist Hospital at Houston for her contribution in the collection of MR data sets, and Habib Zaid for fabricating the force-feedback interface.

REFERENCES

- [1] D. Perrin, N. Vasilyev, P. Novotny, J. Stoll, R. H. P. Dupont, I. Salgo, and P. del Nido, "Image guided surgical interventions," *Current Problems in Surgery*, vol. 46, pp. 730 – 766, 2009.
- [2] J. Cadeddu, D. Stoianovici, R. Chen, R. Moore, and L. Kavoussi, "Stereotactic mechanical percutaneous renal access," *Journal of Endourology*, vol. 12, no. 2, pp. 121 – 125, 1998.
- [3] W. Saliba, V. Reddy, O. W. J. Cummings, J. B. M. Haissaguerre, J. Kautzner, P. Peichl, P. Neuzil, V. Schibgilla, G. Noelker, J. Brachmann, L. D. Biase, C. Barrett, P. Jais, and A. Natale, "Atrial fibrillation ablation using a robotic catheter remote control system: initial human experience and long-term follow-up results," *Journal of the American College of Cardiology*, vol. 51, no. 25, pp. 2407 – 2411, 2008.
- [4] S. Pandya, J. Motkoski, C. Serrano-Almeida, A. Greer, I. Latour, and G. Sutherland, "Advancing neurosurgery with image-guided robotics," *Journal of Neurosurgery*, vol. 111, no. 6, pp. 1141 – 1149, 2009.
- [5] B. Larson, N. Tsekos, and A. Erdman, "A robotic device for minimally invasive breast interventions with real-time MRI guidance," in *IEEE Symposium on Bioinformatics and Bioengineering*, 2003, pp. 190 – 197.
- [6] A. Patriciu, D. Petrisor, M. Muntener, D. Mazilu, M. Schar, and D. Stoianovici, "Automatic brachytherapy seed placement under MRI guidance," *IEEE Transaction on Biomedical Engineering*, vol. 54, no. 8, pp. 1499 – 1506, 2007.
- [7] A. Gosline, N. Vasilyev, A. Veeramani, M. Wu, G. Schmitz, R. Chen, V. Arabagi, and a. P. D. P. Nido, "Metal mems tools for beating-heart tissue removal," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 1921 – 1926.
- [8] M. Li, A. Kapoor, D. Mazilu, and K. Horvath, "Pneumatic actuated robotic assistant system for aortic valve replacement under MRI guidance," *IEEE Transaction on Biomedical Engineering*, vol. 58, no. 2, pp. 443 – 451, 2011.
- [9] W. Saliba, J. Cummings, S. Oh, Y. Zhang, T. Mazgalev, R. Schweikert, J. Burkhardt, and A. Natale, "Novel robotic catheter remote control system: feasibility and safety of transeptal puncture and endocardial catheter navigation," *Journal of Cardiovascular Electrophysiology*, vol. 17, no. 10, pp. 1102–1105, 2006.
- [10] R. Mebarki, A. Krupa, and C. Collewet, "Automatic guidance of an ultrasound probe by visual servoing based on b-mode image moments," in *Medical Image Computing and Computer Assisted Intervention*, 2008, pp. 339 – 346.
- [11] S. Yuen, S. Kesner, N. Vasilyev, P. D. Nido, and R. Howe, "3d ultrasound-guided motion compensation system for beating heart mitral valve repair," in *Medical Image Computing and Computer Assisted Intervention*, 2008, pp. 711 – 719.
- [12] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixture for robotic cardiac surgery," in *Medical Image Computing and Computer Assisted Intervention*, 2001, pp. 1419 – 1420.
- [13] J. Ren, R. Patel, K. McIsaac, G. Guiraudon, and T. Peters, "Dynamic 3-d virtual fixtures for minimally invasive beating heart procedures," *IEEE Transaction on Medical Imaging*, vol. 27, no. 8, pp. 1061–1070, 2008.
- [14] K. Kwok, G. Mylonas, L. Sun, M. Lerotic, J. Clark, T. Athanasiou, A. Darzi, and G. Yang, "Dynamic active constraints for hyper-redundant flexible robots," in *Medical Image Computing and Computer Assisted Intervention*, 2009, pp. 410 – 417.
- [15] J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *International Symposium of Robotics Research*, 2005, pp. 49 – 64.
- [16] J. J. Abbott, "Virtual fixtures for bilateral telemanipulation," Ph.D. dissertation, The Johns Hopkins University, Baltimore, Maryland, August 2005.
- [17] M. Wierzbicki, M. Drangova, G. Guiraudon, and T. Peters, "Validation of dynamic heart models obtained using non-linear registration for virtual reality training, planning, and guidance of minimally invasive cardiac surgeries," *Medical Image Analysis*, vol. 8, no. 3, pp. 387 – 401, 2004.
- [18] C. Gendrin and P. Markelj, "Validation for 2d/3d registration ii: The comparison of intensity- and gradient-based merit functions using a new gold standard data set," *Medical Physics*, vol. 38, no. 3, p. 14911502, 2011.
- [19] K. Horvath, M. Li, D. Mazilu, M. Guttman, and E. McVeigh, "Real-time magnetic resonance imaging guidance for cardiovascular procedures," *Seminars in thoracic and cardiovascular surgery*, vol. 19, pp. 330–335, 2007.
- [20] E. McVeigh, M. Guttman, R. Lederman, M. Li, O. Kocaturk, T. Hunt, S. Kozlov, and K. Horvath, "Real-time interactive MRI-guided cardiac surgery: Aortic valve replacement using a direct apical approach," *Magnetic Resonance in Medicine*, vol. 56, pp. 958 – 964, 2006.
- [21] M. Guttman, C. Ozturk, A. Raval, V. Raman, A. Dick, R. DeSilva, P. Karmarkar, R. Lederman, and E. McVeigh, "Interventional cardiovascular procedures guided by real-time mr imaging: an interactive interface using multiple slices, adaptive projection modes and live 3d renderings," *Journal of Magnetic Resonance Imaging*, vol. 26, no. 6, pp. 1429–1435, 2007.
- [22] N. V. Navkar, Z. Deng, D. J. Shah, K. E. Bekris, and N. Tsekos, "Visual and force-feedback guidance for robot-assisted interventions in the beating heart with real-time MRI," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 689 – 694.
- [23] T. Walther, T. Dewey, M. A. Borger, J. Kempfert, A. Linke, R. Bechtd, V. Falk, G. Schuler, F. W. Mohr, and M. Mack, "Transapical aortic valve implantation: Step by step," *The Annals of Thoracic Surgery*, pp. 276 – 283, 2008.
- [24] N. Navkar, E. Yeniaras, D. Shah, N. Tsekos, and Z. Deng, "Generation of 4d access corridors from real-time multislice MRI for guiding transapical aortic valvuloplasties," in *Medical Image Computing and Computer Assisted Intervention*, 2011, pp. 249 – 257.
- [25] D. Kochanek and R. Bartels, "Interpolating splines with local tension, continuity, and bias control," in *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 1984, pp. 33 – 41.
- [26] C. Loop, "Smooth subdivision surfaces based on triangles. (master thesis) department of mathematics, university of utah," Utah, 1987.
- [27] L. Lyons, R. Webster, and R. Alterovitz, "Motion planning for active cannulas," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 801–806.
- [28] E. Christoforou, E. Akbudak, A. Ozcan, M. Karanikolas, and N. Tsekos, "Performance of interventions with manipulator-driven real-time mr guidance: implementation and initial in vitro tests," *Magnetic Resonance Imaging*, vol. 25, pp. 69–77, 2007.
- [29] N. Tsekos, A. Khanicheh, E. Christoforou, and C. Mavroidis, "Magnetic resonance compatible robotic and mechatronics systems for image-guided interventions and rehabilitation: A review study," *Annual Review of Biomedical Engineering*, vol. 9, pp. 351 – 387, 2007.
- [30] A. Krieger, G. Metzger, G. Fichtinger, E. Atalar, and L. Whitcomb, "A hybrid method for 6-dof tracking of MRI-compatible robotic interventional devices," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 3844 – 3849.