2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Active Motion Compensation in Robotic Cardiac Surgery*

G. P. Moustris, A. I. Mantelos., and C. S. Tzafestas

Abstract—**Motion compensation is a prominent application in robotic beating heart surgery, with significant potential benefits for both surgeons and patients. In this paper we investigate an activate assistance control scheme on a simple tracking task, which helps the surgeon guide the robot on a predefined reference. The control is implemented on top of a shared control system, which serves as a basis for implementing higher level controllers. Experiments with a trained surgeon are also presented, which show the positive effect of the approach.**

I. Introduction

Robotic surgery has seen a great increase in the last decade through the adoption of specialized surgical robots such as the DaVinci system (Intuitive Surgical), the Cyberknife (Accuray), the Magellan system (Hansen Medical), and other. Most of the systems follow a master-slave paradigm and fall under the category of telerobotics, where the surgeon teleoperates the robotic slave manipulators sitting at the master console. Autonomy is not introduced into the control loop since it poses several legal, ethical and technological challenges [1], although it is an active research topic having produced interesting results [2].

Semi-autonomy, i.e. the introduction of autonomy in subtasks in surgery, combining the user input with the computer actions, is a promising application since the main goal is to augment the surgeon on his/her tasks, rather than bypass him/her. A prominent semi-autonomous technique is the so-called motion compensation in robotic beating-heart surgery. Motion compensation refers to the apparent cancellation of organ motion in the surgical field through image processing and robot control algorithms. Typically, the motion of the surgical field (e.g. heart beat, respiratory motion etc) is captured by an imaging device in real-time, is rectified, and presented to the surgeon as still. Concurrently, the robot maintains a steady pose with respect to the field, essentially tracking its motion and moving along with it. This function however, is transparent to the surgeon on the master console, who effectively operates on a static image without perceiving the motion of the slave robot. This approach is of particular interest in coronary artery bypass graft surgery (CABG). Typically in CABG, the patient undergoes cardiopulmonary bypass (CPB), being connected to the heart-lung machine to stop the heart from beating, in order for the surgeon to operate on it. In off-pump CABG (OPCAB), the surgeon operates directly on the beating heart, without the use of CPB. However, mechanical or vacuum stabilizers are placed on the surgical field (myocardium), which reduce the heart motion. Even though OPCAB has been shown to have significant positive effects on the patient, the stress induced by the stabilizers on the heart can cause hemodynamic disorders [3]. Furthermore, there is also a residual motion in the field since the stabilizers do not cancel the motion entirely. Robotic motion compensation emerges as a promising solution to these problems because it can obviate the need for stabilizers. The control scheme falls under the shared control paradigm since both the controller and the surgeon use the robot at the same time. Motion compensation presents challenges on two ends. The first is the image capture and rectification of the motion itself, as it can present very fast dynamics. On the other end, the control of the robot is also demanding by having to track very fast moving targets.

Motion compensation generally comprises three main tasks; mechanical synchronization, image stabilization and shared control. The notion of heartbeat synchronization was introduced in [4] using a high speed camera (995fps) to track a laser point on an image. Similarly in [5], four markers were tracked on the surface of a heart using a 500fps camera. The fast dynamics have dictated the use of Model Predictive Controllers [5,6] and the introduction of biological signal in the prediction loop [7–9]. Other modalities such as ultrasound [10,11] and forces sensors [12] have also been presented. Image rectification and stabilization algorithms for motion compensation are presented in [13–16].

Shared control has received little attention, although it serves a critical role by allowing the surgeon to operate on the still image and concurrently compensate the cardiac movement. To this end, this paper presents a theoretical unifying framework under which the mechanical synchronization, the image stabilization and the shared control emerge and combine naturally. This is achieved by using the Strip-Wise Affine Map (SWAM) [17], which shifts the control to a stabilized canonical space where the robot and the image are still. It also serves as a basic motion compensation layer, on top of which higher-level controllers can be implemented. In this paper, we propose and implement an active-assistance controller, aiming to help the surgeon guide the robot for a simple tracking task on the heart, thus increasing the accuracy and removing motion noise.

*The work leading to these results has received funding by the European Union under project MOBOT with grant FP7-ICT-2011-9-600796.
G. P. Moustris, A. I. Mantelos., and C. S. Tzafestas are with the School of Electrical & Computer Engineering, National Technical University of Athens, Intelligent Robotics & Automation Laboratory, Zographou Campus, Athens, GR 15773 (phone: +30-210-7721527, fax:+30-210-7722489, e-mails: {gmoustri@mail, mantelos@cc.ece, ktzaf@cs}.ntua.gr)

II. Theoretical Framework

A. Preliminary Concepts

The typical setup for motion compensation in robotic cardiac surgery is presented in Fig. 1. The surgical field (e.g. heart) is manipulated by the slave robot and overviewed by an endoscopic camera. This camera image will be called the "physical image". The reference features are extracted from the image, to be used in subsequent tasks. The physical image is then rectified and presented to the surgeon, who sits on the master console operating the controls. The stabilized image will be called the "canonical image". The surgeon perceives the surgical field in the canonical image as still, and produces the human input which is combined with the reference signal by the shared controller. Closing the loop, the shared control signal is fed to the slave robot.

We identify three spaces attached to each subsystem; the *physical world space* W_p i.e. the Cartesian space where the slave robot and surgical field dwell; the *physical image space* I_p, i.e. the image space of the world space through the camera, and; the *canonical image space* I_c, which is the rectified image space where the physical image is still. The three spaces are related by two transformations. Specifically, $P: W_p \to I_p,\ P \in GPL(W_p)$ is the projective transformation involving the camera matrix. For simplicity, in this work the pinhole camera model is used. The canonical and physical images are related by a non-linear rectification map $\Psi: I_p \to I_c$ which must be bijective in order to prevent unnatural distortions of the surgical field presented to the surgeon.

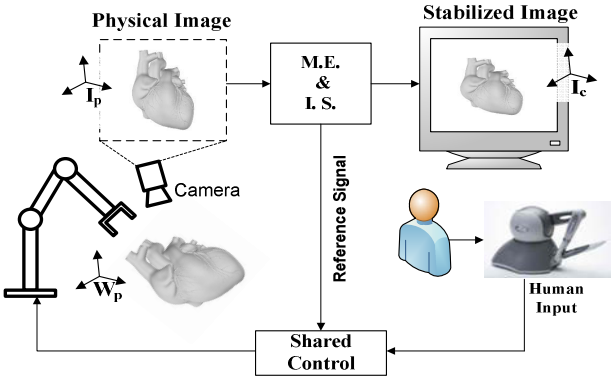

Fig.1. Typical setup for motion compensation in robotic cardiac surgery

In order to produce the reference signal, the system tracks the motion of a *reference manifold* $M_p(u,t)$ in the surgical field. The manifold M_p is parameterized by coordinates $u_i: W_p \to \mathrm{R},\ i=1,2$, that is, $u_i = u_i(q_p),\ q_p \in W_p$. For example, M_p could be a point on the heart surface (0-manifold), a line (1-manifold) or a patch (2-manifold). For the mechanical synchronization task, the goal is to maintain a fixed pose to the reference manifold. Formally, let $T_S(\boldsymbol{\tau}_s)$ describe a frame rigidly attached to the slave robot tool, expressing the pose (position and orientation) with respect to the world frame W_p. We denote the position of T_S as ${}^p q_s$. The vector $\boldsymbol{\tau}_s$ is the robot input. Also, let $T_G(t)$ be a frame of a goal pose on the reference manifold at point $q_G(t)$. T_G depends on time since the reference manifold changes shape due to cardiac pulsation. R_G is the *goal orientation* and q_G is the *goal point* in M_p. The general synchronization problem asks for the input $\boldsymbol{\tau}_s$ such that the frames remain fixed to each other. A simpler version of the problem is to consider only the Cartesian position (i.e. only q_G), and ignore the rotational part. This reduces to a Cartesian tracking problem.

For the image stabilization task, let F_p be the physical image of the field through the camera, and $IM_p = P(M_p)$ be the image of the reference manifold. Formally, F_p is a function from I_p to some color space and IM_p is a subset of $dom(F_p)$. F_p is mapped to the canonical image space through Ψ, i.e. $dom(F_c)=\Psi(dom(F_p);IM_p)$. In a similar manner $IM_c=\Psi(IM_p)$, $IM_c \subset dom(F_c) \subset I_c$ is the reference manifold in the canonical image. Note that Ψ is not a static transformation but depends on IM_p as a parameter. This is natural since the purpose of image stabilization is to map the physical image to the canonical image in a way such that the reference manifold in the physical image is mapped to a *fixed* sub-domain in I_c i.e. to IM_c. However since IM_p deforms through time, Ψ must depend on the actual IM_p at each instance. For motion compensation however, one needs to account for the image transformation of the robot as well, since the surgical tool is visible in the stabilized image. In the general case where the robot's pose is considered, the entire robot image block must be transformed so as to stabilize the orientation as well as the position. In the simpler case of Cartesian synchronization, only the robot's position in the image must be compensated.

In this work we consider the Cartesian compensation problem for a planar *1*-manifold reference i.e. a planar curve. The canonical reference manifold IM_c is a fixed straight line in the canonical image. Thus, the physical curve is always mapped to a straight line. In such a case, if the surgeon wants to cut along a curve on the pulsating heart, the task is reduced to tracking a static straight line in the canonical image.

In order to produce the motion synchronization and image stabilization algorithms, we introduce the *canonical world space* W_c, which is a Cartesian 3D space containing the stabilized robot and reference manifold. The physical and canonical world spaces are related by the bijective map, $\Phi: W_c \to W_p$, as seen in the commutative diagram of Fig. 2.

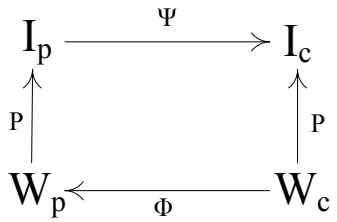

Fig. 2. Commutative diagram relating the four spaces and transformations

The control of the robot takes place in the canonical world space, where the reference manifold M_c is a straight line.

Essentially the surgeon operates in W_c looking at its image in I_c.

From the diagram, it can be verified that $\Psi = P \circ \Phi^{-1} \circ P^{-1}$. The inverse projective map P^{-1} exists because the reference manifold lays on the plane z=0. The map Φ is an extended version of the strip-wise affine map, analyzed in the following section.

B. Strip-Wise Affine Map

The strip-wise affine map (SWAM) is a piecewise linear homeomorphism between the two world spaces. It acts by inducing a strip decomposition on the *x-y* plane and then applying an affine map between them. It has been successfully applied to mobile robot tracking control since it reduces path tracking to straight line tracking [18,19]

The map takes a *polygonal line* from the physical space, and maps it to the x_c-axis in the canonical space. This fits well into the control as the reference manifold is extracted from the physical image and projected to the physical space as a collection of points, rather than a smooth curve. Thus let $\{w_i\}$, $i=1...n$, $w_i = ({}^p x_i, {}^p y_i, 0)$ are the vertices of the polygonal line, lying on the physical plane z=0. Each vertex is projected to a point α_i on the real axis in the canonical world according to its normalized length. Furthermore, let $q_p = (x_p, y_p, z_p)$ be a point in W_p and $q_c = (x_c, y_c, z_c)$ a point in W_c. The SWAM maps q_c to q_p under the equation,

$$q_p = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} y_c S \cos\theta_s + f_x(x_c) \\ y_c S \sin\theta_s + f_y(x_c) \\ z_c \end{bmatrix} = \Phi(q_c) \qquad (1)$$

where θ_s is some angle, called the *shifting angle*. The functions f_x and f_y are given by,

$$\begin{aligned} f_x(x_c) &= \sum_{k=0}^{n} [{}^p x_k + S \cdot (x_c - a_k)\cos\theta_k]\psi_k, \\ f_y(x_c) &= \sum_{k=0}^{n} [{}^p y_k + S \cdot (x_c - a_k)\sin\theta_k]\psi_k \end{aligned} \qquad (2)$$

The angles θ_k are the angles of each edge $[w_k\ w_{k+1}]$ with respect to the physical x-axis, and $\psi_k(x_c)$ is a unit rectangle pulse function in W_c with support set $[a_k, a_{k+1})$ (for a more thorough presentation of the SWAM see [17]). Observe that (2) is a piece-wise linear parameterization of the reference manifold such that $M_p = (f_x, f_y, 0)$, and thus x_c is the coordinate parametrizing it. Equations (1),(2) show that if the robot moves on a vertical plane parallel to the x_c-axis in the canonical space, its image in the physical space moves parallel to the reference manifold by a standard offset $g = (y_c S\cos\theta_s, y_c S\sin\theta_s, z_c)$ from point $(f_x(x_c), f_y(x_c), 0)$ laying on M_p. Defining the goal point as,

$$q_G(x_c) = \begin{bmatrix} f_x(x_c) & f_y(x_c) & 0 \end{bmatrix}^T \qquad (3)$$

(1) takes the compact form,

$$q_p = g(y_c, z_c) + q_G(x_c) \qquad (4)$$

Using the SWAM the control of the robot is transferred to the canonical world where the objective for the surgeon is to track the x-axis. Using this map as a basis, the active assistance controller is implemented, as described in the next section.

III. Active Assistance

Let W_m be the Cartesian physical space at the master console. In order to bind the master and slave systems we identify W_m with W_c, up to a scaling factor on the *x-y* plane. Thus, if ${}^p q_m = ({}^p x_m, {}^p y_m, {}^p z_m)$ is the position of the master robot in W_m and ${}^c q_m = ({}^c x_m, {}^c y_m, {}^c z_m)$ its respective position in W_c, then $({}^c x_m, {}^c y_m, {}^c z_m) = (L\,{}^p x_m, L\,{}^p y_m, {}^p z_m)$, where L is some real constant. Similarly, let ${}^p q_s$ be the physical position of the slave robot in W_p, and ${}^p q_r, {}^c q_r$ be the physical and canonical images of a *reference point* (Fig. 3).

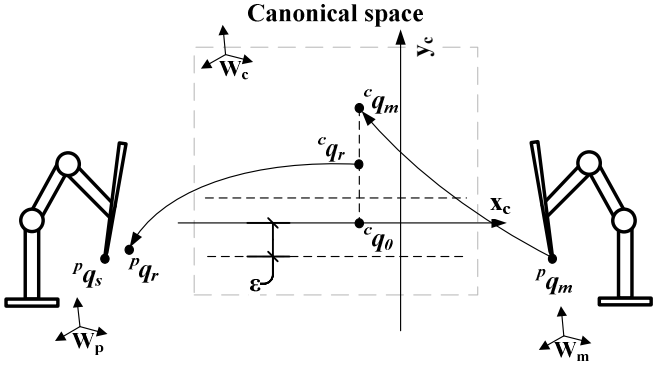

Fig. 3. Definition of the robot and reference images in the physical and canonical spaces.

On the slave side, the goal of the robot is to follow the physical reference point ${}^p q_r$. In this work we have used a simple PID controller to perform this task. However, we additionally introduce a higher-level assistive controller, which is implemented in the canonical space. The goal of this control is to assist the surgeon in tracking the reference manifold i.e., the canonical x_c-axis.

If the canonical reference and master points are identical, viz. ${}^c q_r \equiv {}^c q_m$, then the control corresponds to a pure teleoperation scheme. To insert the proposed active assistance control stage, we firstly dissociate the two points (reference and master) in the canonical space, and then attach a second-order dynamics to the system by making ${}^c q_m$ and ${}^c q_0$ attractive for ${}^c q_r$. This is implemented using a linear spring-mass-damper model. Point ${}^c q_0 = ({}^c x_m, 0, {}^c z_m)$ is

the projection of ${}^c q_m$ on x_c. Note that the attractors affect only the y_c-axis, whereas x_c is transferred unaltered to the physical slave space. Formally, let m be a virtual mass attached to the canonical reference point ${}^c q_r$. We assume that the spring-damper models on the two attractors are identical, with K, B being the spring constant and damping coefficient respectively. The forces exerted on the mass m by the two attractors are:

$$\begin{aligned} {}^cF_m &= -K({}^c y_r - {}^c y_m) - B({}^c \dot{y}_r - {}^c \dot{y}_m), \\ {}^cF_0 &= -K\,{}^c y_r - B\,{}^c \dot{y}_r, \\ m\,{}^c \ddot{y}_r &= {}^cF_T = {}^cF_m + {}^cF_0 \end{aligned} \tag{5}$$

${}^cF_T$ is the total force acting on m.

System (5) has an undesirable effect on the transparency of the teleoperation. The attractor ${}^c q_0$ exerts forces on the reference point even if ${}^c q_r$ is away from the line (the x_c-axis that is being tracked in the canonical space). In this case, the surgeon will observe the slave robot with a motion drag in the entire surgical field. Since the aim of the control is to assist the surgeon in tracking the x_c-axis, we insert a zone of width 2ε centered about the axis, in which the attractive forces are combined, i.e.:

$$\begin{aligned} {}^cF_T &= \lambda({}^c y_m)\,{}^cF_m + \left(1-\lambda({}^c y_m)\right){}^cF_0, \\ \lambda({}^c y_m) &= \begin{cases} 1 & ,{}^c y_m \le -\varepsilon \text{ or } \varepsilon \le {}^c y_m \\ ({}^c y_m - \delta)/(\varepsilon - \delta) & ,\delta \le {}^c y_m < \varepsilon \\ 0 & ,-\delta \le {}^c y_m < \delta \\ -({}^c y_m + \delta)/(\varepsilon - \delta) & ,-\varepsilon \le {}^c y_m < -\delta \end{cases} \end{aligned} \tag{6}$$

In this model, if the canonical master is outside the zone, it is the only attractor of the reference point. Similarly, if ${}^c y_m \in (-\delta, \delta)$, i.e. in the *dead zone*, the reference is attracted only by the line. In the middle zones, the forces are combined linearly. Using (5),(6) and expanding,

$${}^c \ddot{y}_r = -\frac{K}{m}\,{}^c y_r - \frac{B}{m}\,{}^c \dot{y}_r + \lambda({}^c y_m)\left(\frac{K}{m}\,{}^c y_m + \frac{B}{m}\,{}^c \dot{y}_m\right) \tag{7}$$

Substituting $z_1 = {}^c y_r$, $z_2 = {}^c \dot{y}_r$, the state equations are:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & -\frac{B}{m} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{K}{m} & \frac{B}{m} \end{bmatrix} \begin{bmatrix} {}^c y_m \\ {}^c \dot{y}_m \end{bmatrix} \lambda({}^c y_m) \tag{8}$$

Transforming the input according to the surjection on R^2,

$$({}^c y_m, {}^c \dot{y}_m) \cdot \lambda({}^c y_m) = (u_1, u_2) \tag{9}$$

the new input is (u_1,u_2) and (8) is transformed into a second order controllable linear form,

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & -\frac{B}{m} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{K}{m} & \frac{B}{m} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{10}$$

Of course, for $K,B>0$ the system is asymptotically stable. System (10) is solved incrementally in the control loop using the Euler integration method.

Let $\boldsymbol{z}(t) = (z_1(t),\ z_2(t))$, $\boldsymbol{u}(t) = (u_1(t),\ u_2(t))$ be the system state and input at time t; the state at $t+\Delta T$ is given by,

$$\begin{aligned} z_1(t+\Delta T) &= z_1(t) + \Delta T \cdot z_2(t) \\ z_2(t+\Delta T) &= z_2(t) + \Delta T \cdot f(\boldsymbol{z}(t), \boldsymbol{u}(t)) \end{aligned} \tag{11}$$

where f equals the right hand side of (7), that is,

$$f = -\frac{K}{m} z_1(t) - \frac{B}{m} z_2(t) + \frac{K}{m} u_1(t) + \frac{B}{m} u_2(t) \tag{12}$$

Then, by definition ${}^c y_r(t+\Delta T) = z_1(t+\Delta T)$.

IV. Implementation

The master-slave system consists of the PHANToM Omni and Desktop for the master-slave sides respectively. The robots have identical mechanical structure, with DC motor actuators for the first three joints. The Tool Center Point (TCP) is located at the intersection of the last three joints that form the gimbal. Since only Cartesian compensation is considered, utilizing the first three joints, the gimbal was locked and rigidly attached to the second link in each robot.

To simulate a lancet on the Slave robot, a metallic needle was attached to the gimbal stylus (see Fig. 4), mirroring a similar metallic tip on the master. The robots were calibrated using checkerboards, which also defined the physical and canonical world frames. In front of the slave robot lays a semi-transparent screen, defining the surgical field. Underneath, there is a projector showing the reference line on the screen. The surgeon views the field through a pole-mounted camera, which was calibrated and registered to W_p

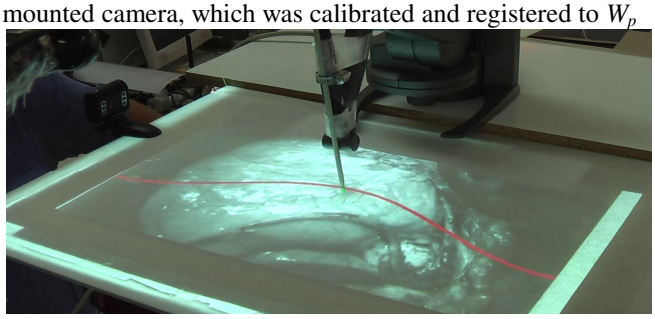

Fig. 4. View of the slave robot and the surgical field. The robot tracks a red line on a heart image projected onto a semi-transparent screen from a projector underneath the screen.

In order to reduce the computational overhead and latency of the system, the robots were connected to two different

computers while the camera was feeding the input to a third. The system runs in three parallel loops, distributed across the three computers; the Servo loop; the Graphics loop and; the Communications loop.

The servo loop, operating at 1 KHz, controls the slave robot querying its configuration and implementing the PID controller for the tracking of the reference position, setting the forces on the first three joints. The PID follows the update rate of the communication loop (100 Hz) since all three loops are asynchronous. In this loop, the assistive controller also runs, at approximately 250Hz, solving the dynamics using the forward Euler method.

The Graphics loop performs the image acquisition and rectification on the master console. It acquires the camera image, processes it and presents it to the surgeon stabilized. The loop was runs in MATLAB and OpenCV, achieving a refresh rate of approximately 30Hz. To speed up the computation, the image resolution was reduced to 320x240 pixels and converted to grayscale, following the detection of the red reference line. The graphics loop essentially applies the Ψ transform on the physical image.

Finally, the communications loop utilizes UDP sockets at an update rate of 100 Hz. Since UDP lacks an error correction mechanism, the data packets were framed with a predefined header to detect corruption.

V. Experimental Results

For the experiments, a trained surgeon was asked to operate the master console, in order to track a pulsating red line embedded into an endoscopic image of the heart. The entire image was deformed according to the reference line in order to give the impression of a beating heart. The line followed a periodic movement, driven by a pre-recorder ECG signal set to various frequencies. To capture the robot motion, a green marker was attached to the needle tip. Overlooking the scene was a HD camera recording the experiment in 1080p resolution. Prior to the experiments both the surgeon's and the HD camera were calibrated and registered to the physical world frame.

Two groups of experiments were performed; the first corresponding to a pulsation frequency of 12 bpm and the second to 15 bpm (these rates where the maximum allowed from our current h/w implementation in order for the system to perform in real-time, since the graphics processing injects significant delays in the control loop). In the first group, four runs were carried out; *without* compensation, *with* simple compensation, active compensation *without* dead zone (ε=15mm, δ=0) and active compensation *with* dead zone (ε=15mm, δ=7.5mm). The second group included two runs; active compensation *with* dead zone and *no* compensation at all. The surgeon viewed the field on a monitor, showing the physical and canonical images (Fig. 5).

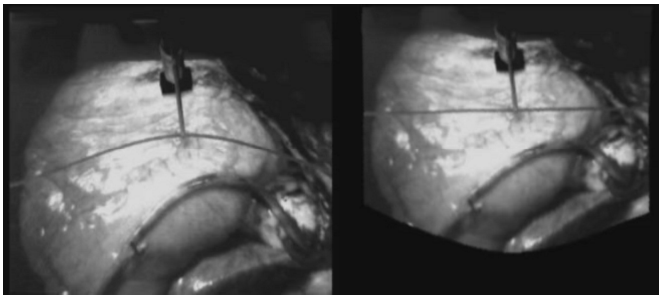

Fig. 5. View of the surgical field through the camera. (Left) Physical image. (Right) Canonical image. The slave robot is visible in both images.

The surgeon was asked to follow the line from end to end for about 1.5 minutes. The slave robot's tip movement was extracted in post-processing from the HD camera video. A simple color filter was used in each frame to extract the tip's green marker, as well as the red line. Following, the pixel trajectory was filtered to reduce noise and back-projected to the Cartesian space to perform the analysis.

To assess the effect of active compensation, the error distance of the tip to the line was calculated for each frame and a statistical analysis was performed. The normalized and cumulative histograms of the calculated errors for the first group are shown in Fig. 6.

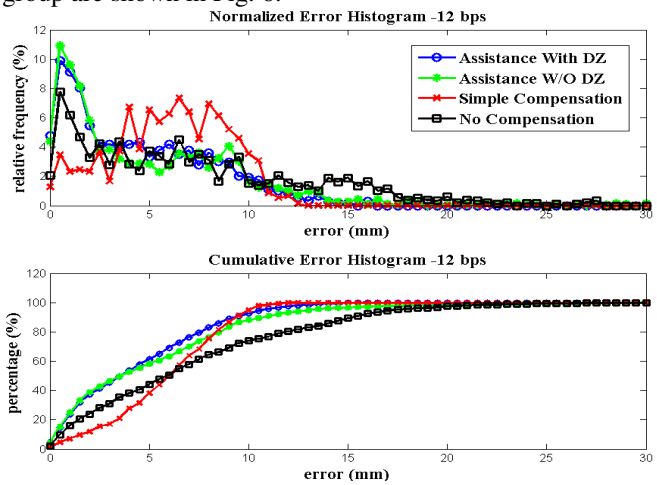

Fig. 6. Normalized and Cumulative error histograms for the first group. (blue) assistance with dead zone. (green) assistance without dead zone. (red) simple compensation. (black) no compensation

A summary of the statistical results of the first group is presented in Table I.

TABLE I
STATISTICAL RESULTS FOR THE FIRST GROUP

	AWDZ[a]	AWODZ[b]	SC[c]	NC[d]
mean (mm)	4.487	5.131	6.031	7.171
rel. diff.[e]	-12.57%	-14.91%	-15.90%	-

[a]Assistance with Dead Zone, [b]Assistance W/O Dead Zone, [c]Simple Compensation, [d]No Compensation, [e]Relative Difference between consecutive columns.

Reading the data in Table I, from left to right, we observe that the active assistance *with* dead zone reduces the mean error, with respect to active assistance *without* dead zone, by 12.57%, which in turn reduces the error by 14.91% w.r.t simple compensation, which finally reduces the error by

15.90% w.r.t no compensation. To investigate the effect of the heartbeat frequency in the active assistance, the second group of experiments was performed. Aggregate data for the two groups are presented in Table II.

TABLE II
AGGREGATE RESULTS FOR THE TWO GROUPS

	12 bpm		15 bpm	
	AWDZ	NC	AWDZ	NC
mean (mm)	4.487	7.171	4.400	8.022
rel. diff.	-37.43%	-	-45.15%	-

Table II shows a consistent decrease of the mean error across the two frequencies. Notice that the error in the active assistance, in absolute terms, remains virtually the same. This implies that the assistive controller is robust with respect to the pulsation frequency, and is approaching its lower threshold i.e. it effectively cancels the effects of motion irrespective of the frequency. The residual error of approximately 4.5mm can be attributed to the specific implementation since the computations insert a measurable delay in the loop which affects the accuracy on the slave end.

VI. CONCLUSION AND FUTURE WORK

We have presented an active assistance controller which effectively augments the surgeon in a simple tracking task, in robotic beating heart surgery. The approach unifies the motion synchronization and image stabilization tasks, serving as a basis for higher level control schemes. The algorithms were implemented in a prototype teleoperated system used in experiments with a trained surgeon. The experiments showed a positive effect of the assistive controller, which persisted across different frequencies of heart pulsation. Future work aims at extending the reference to an arbitrary surface, using all six degrees of freedom of the robot and investigating more complex controller in this direction.

REFERENCES

[1] C. Elliott, L. Paterson, G. Clarke, B. Whitby, and W. Bardo, Autonomous Systems: social, legal and ethical issues, The Royal Academy of Engineering, 2009..

[2] G.P. Moustris, S.C. Hiridis, K.M. Deliparaschos, and K.M. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," Int. J. Med. Robotics Comput. Assist. Surg., vol. 7, Dec. 2011, pp. 375–392.

[3] P.P.M. de Oliveira, D.M. Braile, R.W. Vieira, O. Petrucci Junior, L.M. Silveira Filho, K.A. de S. Vilarinho, E.S. de O. Barbosa, and N. Antunes, "Hemodynamic disorders related to beating heart surgery using cardiac stabilizers: experimental study," *Revista brasileira de cirurgia cardiovascular: órgão oficial da Sociedade Brasileira de Cirurgia Cardiovascular*, vol. 22, Dec. 2007, pp. 407–415.

[4] Y. Nakamura, K. Kishi, and H. Kawakami, "Heartbeat synchronization for robotic cardiac surgery,", in Proc. of the IEEE Int. Conf. on Robotics and Automation, Seoul, Korea, May 2001, pp. 2014–2019 vol.2.

[5] J. Gangloff, R. Ginhoux, M. de Mathelin, L. Soler, and J. Marescaux, "Model predictive control for compensation of cyclic organ motions in teleoperated laparoscopic surgery," IEEE Trans. Control Syst. Technol, vol. 14, 2006, pp. 235–246..

[6] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M.M.A. Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control," IEEE Trans. Robot., vol. 21, 2005, pp. 67–79.

[7] T. Ortmaier, M. Groger, D.H. Boehm, V. Falk, and G. Hirzinger, "Motion estimation in beating heart surgery," IEEE Trans. Biomed. Eng., vol. 52, 2005, pp. 1729–1740.

[8] O. Bebek and M.C. Cavusoglu, "Predictive control algorithms using biological signals for active relative motion canceling in robotic assisted heart surgery," in Proc. of the IEEE Int. Conf. on Robotics and Automation, Orlando, Florida, May 2006, pp. 237–244.

[9] V. Duindam and S. Sastry, "Geometric motion estimation and control for robotic-assisted beating-heart surgery," in Proc. of IEEE Conf. on Intelligent Robots and Systems, San Diego, USA, 2007, pp. 871-876

[10] S. Yuen, S. Kesner, N. Vasilyev, P. Del Nido, and R. Howe, "3D Ultrasound-Guided Motion Compensation System for Beating Heart Mitral Valve Repair," in Medical Image Computing and Computer-Assisted Intervention, LCNS vol. 5241, 2008, pp. 711-719.

[11] S.B. Kesner and R.D. Howe, "Design and control of motion compensation cardiac catheters," in Proc. of the IEEE Int. Conf. on Robotics and Automation, Anchorage, Alaska, May 2010, pp. 1059–1065.

[12] B. Cagneau, N. Zemiti, D. Bellot, and G. Morel, "Physiological Motion Compensation in Robotized Surgery using Force Feedback Control," in Proc. of the IEEE Int. Conf. on Robotics and Automation, Roma, Italy, April 2007, pp. 1881–1886

[13] P. Mountney and G.-Z. Yang, "Soft Tissue Tracking for Minimally Invasive Surgery: Learning Local Deformation Online," in Proc. Medical Image Computing and Computer Assisted Intervention, 2008, pp. 364–372.

[14] D. Stoyanov, G.P. Mylonas, F. Deligianni, A. Darzi, and G.Z. Yang, "Soft-Tissue Motion Tracking and Structure Estimation for Robotic Assisted MIS Procedures," in Proc. Medical Image Computing and Computer Assisted Intervention, 2005, pp. 139–146.

[15] R. Richa, A.P.L. Bó, and P. Poignet, "Towards robust 3D visual tracking for motion compensation in beating heart surgery," Med. Image Anal., vol. 15, Jun. 2011, pp. 302–315.

[16] D. Stoyanov and G.-Z. Yang, "Stabilization of Image Motion for Robotic Assisted Beating Heart Surgery," in Proc. Medical Image Computing and Computer Assisted Intervention, 2007, pp. 417–424.

[17] G. Moustris and S.G. Tzafestas, "Reducing a class of polygonal path tracking to straight line tracking via nonlinear strip-wise affine transformation," Math. Computer Simulat., vol. 79, Nov. 2008, pp. 133–148.

[18] G.P. Moustris and S.G. Tzafestas, "Switching fuzzy tracking control for mobile robots under curvature constraints," Control Eng. Pract., vol. 19, Jan. 2011, pp. 45–53.

[19] G.P. Moustris, K.M. Deliparaschos, and S.G. Tzafestas, "Feedback equivalence and control of mobile robots through a scalable FPGA architecture," Recent Advances in Mobile Robotics, A.V. Topalov, ed., InTech, 2011, pp. 401–426..