Auto-Guided Movements on Minimally Invasive Surgery for Surgeon Assistance

E. Bauzano, V.F. Muñoz and I. Garcia-Morales

Abstract— **This paper focuses on autonomous movements to aid the surgeon to perform certain tasks. Robotic assistants have solved the drawbacks of Minimally Invasive Surgery (MIS) and provide additional skills to the surgeons. However, some authors argue that these systems could lengthen the operating time. The solution is the automation of certain maneuvers that help the surgeon during a surgical maneuver. This work proposes control architecture for a surgical robot capable of performing autonomous movements. In this way, a trajectory planner based on a behavior concept computes the required velocity vector of the surgical instrument hold by the robot. This planner has been implemented and tested on the control architecture of the surgical assistant CISOBOT, designed and developed at the University of Malaga.**

I. Introduction

THE enormous complexity and costs limit the clinical impact of the robotic assistants for Minimally Invasive Surgery (MIS) despite of its well known advantages. Some authors argue that the use of robots in surgery, although providing more precision, could lengthen operating time [1].

One solution to this problem consists of automating certain surgical maneuvers. *Visual Servoing* is one of the most common techniques to perform automated tasks. Control of surgical instrument movements involves calculating instant linear and angular velocity references in each control period. These references are obtained by analyzing the images from a non-calibrated stereo vision system [2] or acquiring those images through a regular laparoscopic surgery camera [3]. This technique enables safe movements of the tools, for example, on cardiac surgery, so the instruments are synchronized with the heart beat [4]-[5].

Other works are devoted to assist the surgeon with robots during the intervention. This way, some developments have performed autonomous stitching and knot tying procedures [6]; others automatically guide a robotic tip in colonoscopy [7], provide automatic transformations to a robot assistant from laparoscopic navigation to open-surgery motion [8], or give autonomous decisions on *teleoperation* with high communication latency or low bandwidth [9]. There are also more complex systems like EndoPAR which automates a knot tying procedure in heart surgery [10], and allows the surgeon to operate as if there was no heartbeat. Furthermore, there are also studies for human-machine skill transfer on robot assistants so they may perform automatic knot tying procedures [11], or automatic navigation on cochleostomy without pre-operative information [12].

This paper proposes a solution for an auto-guided robotic system equipped with two arms, one for the laparoscopic camera and the other for additional instruments. In particular, this work will be focused on the automatic movement of the surgical instrument arm. This arm must avoid the collision with the surgeon tool during its navigation. The main goal of this development is devoted to replacing a human assistant for the surgeon in certain laparoscopic surgery procedures.

The structure of this article is divided into five sections. After this introduction, section II states the control architecture proposed to solve auto-guided movements on MIS robotic assistants. With this general solution, section III explains the developed methodology for moving the robot to one of the surgeon's tool tip. This technique has been applied to the task of taking gauze to the surgeon, as presented on section IV. Finally, section V discusses some possible improvements as well as related future works.

II. Control Architecture

This section introduces the architecture for CISOBOT, a robotic system developed at the University of Malaga with two arms, one for the laparoscopic camera and the other for the additional instrument. Firstly, a brief description of the surgical workspace is given to explain the situation where the robot has to develop the task and all movement constraints that limit its freedom. Once the problem has been presented, the specifications the robot must accomplish are discussed. These requirements should let the robotic assistant navigate into the abdominal cavity in a safely way.

A. Problem statement

The environment where the robot interacts with the patient as well as the surgeon consists of a closed space, the abdominal cavity, as it is shown in Fig. 1. Both, the *camera C* and the *robot tool R*, are inserted through their respective points of insertion G_C and G_R over the abdomen, so-called *fulcrum points*. Moreover, this environment also includes the surgeon's tools he or she uses for the surgery procedures. The *primary tool P* is considered the target for the robot,

Manuscript received March 10, 2010. This work was supported in part by the Spanish National projects DPI2007-62257 and the Regional one P07-TEP-02897.

V. F. Muñoz is with the System Engineering and Automation Department, University of Malaga, Spain (corresponding author to provide phone: +34 951 95 22 49; e-mail: vfmm@uma.es).

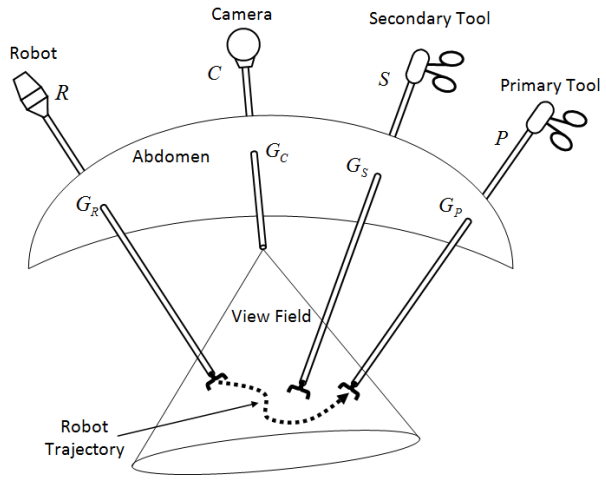

Fig. 1. The camera is focused over the surgical workspace, whereas the robot tool goes where the surgeon's target tool is located.

whereas the *secondary tool S* is the obstacle. The surgeon may displace both tools *P* and *S* during the robot navigation.

The abdominal cavity where the robot tool may move is defined by a cone-shaped view field, which contains the scene seen on the screen by the surgeon. The robot can just analyze this space, so both, its instrument *R* and the primary tool *P* must remain inside of this cone. This work assumes that movements of the robot tool *R* as well as the surgeons' tools *P* and *S* are performed within the view field.

The fulcrum points G_C and G_R are movement constraints for the camera and the robot tool. Therefore, movements inside the patient are limited to four degrees of freedom. In this way, in order to control the position of both, the laparoscopic camera and the surgical robot tool, it is used a motion controller to consider these constraints [13]-[14].

This paper suggests moving the robot tool *R* to the surgeon's primary tool *P* location. The secondary tool *S* is defined as the unique obstacle to be avoided during an automatic task. The trajectory has to be calculated on-line, because the surgeon's tools are not static since he or she continues the intervention normally. All these considerations will be taken into account on the control architecture proposed on next subsection.

B. *Architecture scheme*

Once the functionality needed for the robot has been described, an architecture scheme which resumes the capabilities of the system can be introduced as shown in Fig. 2. The main element is the *Local Planner*, whose mission is to guide the robot tool to the surgeon's primary tool (with velocity $\vec{v}_P$) avoiding the secondary tool (with velocity $\vec{v}_S$).

For this purpose, some information is needed from the environment. In addition to the own kinematics of the robot given by the *Robot Location* feedback, the *Surgeon Location* must be known in order not only for his or her current tools position, but also for estimating their trajectory thanks to the velocity and acceleration parameters. This prediction is done by the *Surgeon Trajectory Estimator* through the secondary tool velocity $\vec{v}_S$, and allows the robot to update its trajectory $\vec{v}_R$ in order to avoid a collision with the secondary tool *S*.

With all this data, the *Local Planner* may command both arms, the one with the laparoscopic camera and other with the special instrument. A *Visual Servoing* system uses this input, as well as the *Environment Changes* detected by the camera, in order to focus the location of interest automatically. It may also move the camera if the other special instrument needs for its planning.

Finally, the *Planned Trajectories* are sent to the arms so the instrumental of the robot moves accordingly. The *Spherical Control* [13] as well as the *Passive Wrist Emulation Control* [14] are both necessary to know the fulcrum points, which are used for the planned trajectory. As seen in section II.A, this study is centered in the robot tool, whereas the Visual Servoing system is supposed to be valid.

III. Auto-Guided Methodology

The automated movement proposed in this paper consists of reaching a goal position defined by the surgeon's primary tool tip *P*, as it was stated in Fig. 1. In this way, the surgeon's secondary tool *S* is an obstacle the robot must avoid. As it was previously stated, both surgeons' tools may be displaced during the robot movement.

Fig. 3 shows the proposed *Local Planner* scheme presented in Fig. 2 which has been used to compute the required velocity $\vec{v}_R$ for the robot tool *R* (see Fig. 1) in order to reach the primary tool *P* by avoiding the secondary tool *S*. The velocities of the robot $\vec{v}_R$ and both surgeons' tools, $\vec{v}_P$ and $\vec{v}_S$, are used on a Fuzzy Logic algorithm for deciding the best strategy to get closer to the target without collisions.

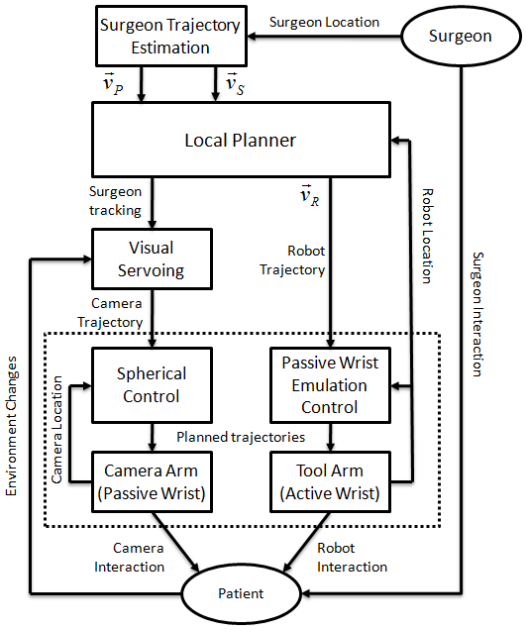

Fig. 2. Architecture scheme for performing automatic tasks of a robot assistant.

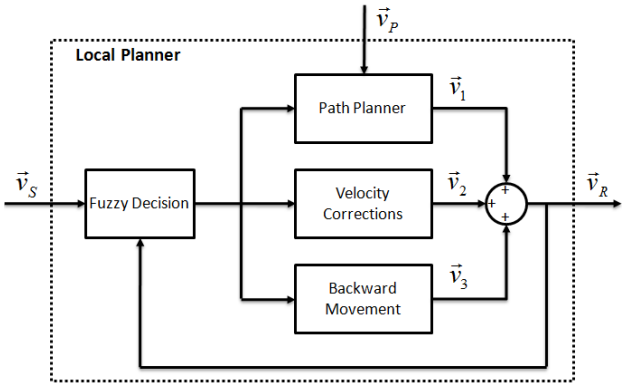

Fig. 3. Local Planner scheme proposed to avoid dynamic obstacles.

This work proposes the combination of three behaviors in order to plan the required robot velocity:

- The "*Path Planner*" behavior finds a trajectory to the target avoiding static obstacles using the Artificial Potential Field methodology (APF), but cannot deal with movable obstacles. However, it can react to primary tool movements so it may be used as a *pursuit behavior*.
- The "*Velocity Corrections*" behavior changes the robot tool velocity $\vec{v}_R$ depending on the secondary tool $\vec{v}_S$. These corrections are more relevant with high velocities and when the robot is nearby the secondary tool.
- The "*Backward Movement*" behavior covers the possibility that the robot tool and the obstacle tool are so close they may collide. This situation is solved by adapting the robot movement to the obstacle in order to avoid the collision, as the surgeon will probably want to displace the robot to free his or her vision space.

As it has been already stated, the decision of which behavior is the best on each situation is taken by a Fuzzy algorithm. Depending on the directions of the robot velocity $\vec{v}_R$ and the secondary tool $\vec{v}_S$, the Fuzzy decision will use the best combination of the behaviors previously stated. The truth table of this Fuzzy decision appears on Table I.

TABLE I
TRUTH TABLE FOR BEHAVIOR DECISION

$\vec{v}_O$ \ $\vec{v}_R$	1	2	3	4
1	A	FA	HC	C
2	FA	A	C	HC
3	HC	C	A	FA
4	C	HC	FA	A

A = go Away, FA = Far Away, C = get Closer, HC = High Close.

In this table, velocities have labels assigned. Each label indicates the place of vectors $\vec{v}_R$ and $\vec{v}_S$ in the quadrants of a circle which centre is coincident with the origin of these velocities. For example, if both velocities are 1 then their directions are the same, whereas if $\vec{v}_R$ is 1 and $\vec{v}_S$ is 3, this means that those directions are opposed. The output of Table I is a fuzzy value used to select the most relevant behavior.

As both, the APF methodology and Fuzzy Controllers are used in mobile robots because they can be computed in real time, the Local Planner of Fig. 3 may fulfill this requirement.

A. The "Path Planner" behavior

This behavior is devoted to find a path to the target avoiding static obstacles. This work has applied the Artificial Potential Field (APF) methodology to fulfill this task. APF associates a repulsive potential field U_{rep} for each obstacle of the environment, as well as an attractive potential field to the target U_{att}. Thus, expression (1) states that the resulting potential field generates a virtual force $\vec{F}_1$ which both, attracts the robot to the goal with force $\vec{F}_{att}$ and repels it from the collision with the obstacles through $\vec{F}_{rep}$ force.

$$\vec{F}_1 = -\nabla U_{att} - \nabla U_{rep} = \vec{F}_{att} + \vec{F}_{rep} \qquad (1)$$

The virtual potential functions of this work are based on the expressions of [15]. They have a high value only on surrounding obstacle area, whereas the attractor has a potential field which is proportional to the distance from the robot to the target. The gradient of these expressions gives the force generated by the potential field.

The main problem of APF methodology appears by means of local minima points that may be found on the workspace. This work extends the potential field of [15] by applying the solution proposed by [16] and uses a force field capable of escape from these local minima points.

The APF methodology is commonly applied to path planning of point mobile robots. However, MIS problem requires that the robot tool *R* is not just a point, but a line which departs from the fulcrum point G_R to its tool tip, as it has already been shown in Fig. 1. One consequence is that the target also becomes a line which departs from the robot fulcrum point G_R and ends on the primary tool tip. Furthermore, as both instruments are long enough to be considered like one-dimensional objects, they may only collide in one point. As shown in Fig. 4, the minimal distance ρ between the robot tool and the secondary tool defines the point over each tool M_R and M_P that would collide in case they approach. Point M_R is called the *guide point*, because it is used to move the robot to the target.

Secondly, the default potential function generates equipotential surfaces with a cylindrical shape. However, this work has chosen the use of conical surfaces with their vertices on the fulcrum point of the secondary tool, as it can be shown in Fig. 4. The reason is that the movements are faster near the tool tips, so higher potentials are needed to maintain the distance between robot and the secondary tools.

Therefore, the APF methodology can be just applied to the

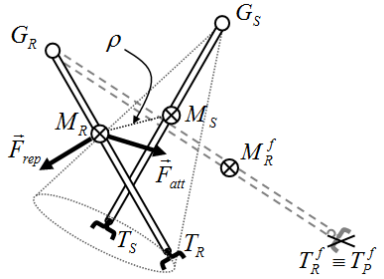

Fig. 4. Virtual force for generating trajectories to the target.

guide point M_R instead all the robot tool longitude. This point has a target associated to its final position M_R^f when the robot tool has reached the target. Thus, to calculate the next trajectory location the algorithm steps are:

Procedure for Automatic Movements

1) Locate the guide point M_R

2) Calculate its target M_R^f

3) Apply the forces given by (1) *in order to know the needed velocity of the guide point*

4) Move the robot instrument to fit the new location and the fulcrum point constraint

End Procedure

First step can be geometrically deduced by solving the equation system of two lines that cross themselves, whereas second step just calculates the position of M_R^f by proportional distances (see Fig. 4). As for third step, the expressions of attraction force $\vec{F}_{att}$ and repulse force $\vec{F}_{rep}$ are given by the Evolution of the APF (EAPF) stated by [16]:

$$\vec{F}_{att} = 2K_a \vec{r}_{goal}$$

$$\vec{F}_{rep} = \begin{cases} -K_r \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 \left(\frac{\vec{r}_{goal}}{\rho^2} + \frac{1}{2}\hat{n} \right) & \text{if } \rho < \rho_0 \\ 0 & \text{if } \rho \geq \rho_0 \end{cases} \quad (2)$$

Fig. 4 shows both forces presented in (2), being K_a and K_r their respective gains. Repulsive force $\vec{F}_{rep}$ only acts on a bound surrounding the obstacle tool below distance ρ_0 and has two components: one part depends on the distance to the target $\vec{r}_{goal} = \overrightarrow{M_R M_R^f}$ as well as the distance to the obstacle tool ρ, and the other is a constant with a direction $\hat{n} = (\hat{r}_{goal} \times \hat{\rho}) \times \hat{\rho}$ perpendicular to $\hat{\rho}$ and always pointing to the target. As a result, this constant forces the robot to move even if it has reached a local minima situation. This component may be also used to avoid any geometrical singularity configuration.

To adapt the potential field of the secondary tool to a cone, a new gain K_r' is defined in (3) which depends on the relation between the secondary tool total length $L = \overline{G_S T_S}$ and the distance from its fulcrum point G_S and the point of minimal distance with the robot M_S (see Fig. 4):

$$K_r' = \frac{\overline{G_S M_S}}{\overline{G_S T_S}} K_r = \frac{l}{L} K_r \quad (3)$$

Finally, the expression of $\vec{v}_1$ is deduced in expression (4) from the virtual force $\vec{F}_1$ of (1), where virtual mass m has been considered to be 1:

$$\vec{v}_1 = \int \frac{\vec{F}_1}{m} dt \rightarrow \vec{v}_1(t + \Delta t) = \vec{v}_1(t) + \frac{\vec{F}_1}{m} \Delta t \quad (4)$$

B. Velocity Corrections

The APF behavior loses its efficiency in dynamic environments because it gives a trajectory just based on the actual state. The fuzzy system will change the robot velocity and try to avoid the collision by reducing the speed in order to allow the surgeon to follow his or her original trajectory [17]. The fuzzy algorithm is based on collision time and collision distance parameters between the robot tool and secondary tool. For example, if the robot is near the secondary tool but they approach themselves very slowly, the robot movement is also slow (or even stopped). However, if the robot distance to the secondary tool is very far, there will be little variation on the velocity calculated by the APF method. All situations appear on the truth table on Table II.

TABLE II
TRUTH TABLE FOR VELOCITY CORRECTIONS

Time \ Distance	Z	AZ	M	F	VF
Z	S	SM	NM	MM	MM
AZ	S	SM	NM	MM	MM
M	SM	SM	NM	FM	MM
L	AS	AS	SM	NM	MM
VL	AS	AS	AS	SM	FM

Z = Zero, AZ = Almost Zero, M = Middle, F = Far, VF = Very Far.
L = Long, VL = Very Long.
S = Stopped, AS = Almost Stopped, SM = Slow Movement, NM = Normal Movement, FM = Fast Movement, MM = Max Movement

The outputs on this table are the speed corrections $\vec{v}_2$ previously presented on Fig. 3 produced by each combination of antecedent data. These rules adapt the speed of the robot tool so that the secondary tool passes before; thus, the cells at the bottom-left of the table, which corresponds to an obstacle that will cross a close point in the robot tool trajectory after a long period of time, produce very slow commands. A value of S (Stop) means that $\vec{v}_2 = -\vec{v}_1$, whereas MM (Max Movement) indicates that $\vec{v}_2 = \vec{0}$, which fits with a maximum velocity generated by the APF behavior already explained in subsection III.A.

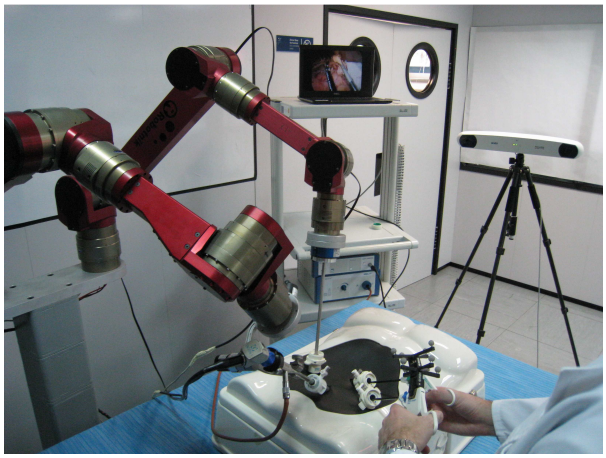

Fig. 5. Experimental robot assistant CISOBOT for the surgeon with the auto-guided movement proposed on this work.

C. Backward movement

With the fuzzy system stated above, almost all situations may be covered. However, collision may also occur if the surgeon moves his or her secondary tool rightly to the robot and forces the contact. This could happen because the surgeon needs to apart for some reason the robot tool for a moment. Thus, the natural solution consists of moving the robot tool rightly into the secondary tool velocity direction. This way, the robot would follow the surgeon as long as it is moving, and should stop when the surgeon does. Only when the secondary tool frees the default APF trajectory, the robot would restart its normal auto-navigation. More specifically, the behavior proposed will be likely a damp between the secondary tool and the robot tool. This way, the virtual force (5) will move the robot back at the secondary tool direction:

$$\vec{F}_3 = B\vec{v}_O \tag{5}$$

The gain B on (5) is a constant which indicates how fast the robot velocity will react to the obstacle. If this parameter is high, then the robot tool will change its velocity very fast to reach the secondary tool velocity $\vec{v}_S$ fast, and similar conclusions can be taken for low values of B. Therefore, the resulting velocity of the robot tool can be obtained by adding the three behaviors exposed (6):

$$\vec{v}_R = c_1\vec{v}_1 + c_2\vec{v}_2 + c_3\vec{v}_3 \tag{6}$$

The parameters c_1, c_2 and c_3 on (6) are the importance of each velocity $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$, and are estimated by the Fuzzy decision explained in the beginning of this section.

IV. Implantation and Experiments

This section describes the experiments considered to validate the proposed methodology for auto-guided movements. For this purpose, it has been used the CISOBOT system, which can be viewed on Fig. 5. This is a two-arm robotic system for holding both, the laparoscopic camera and the surgical instrument. On the left side of the picture, the camera arm has a passive wrist attached to perform the navigation [13] (last degrees of freedom are disabled), whereas the instrument arm on the right has an active wrist. This arm has a force sensor to give information about the fulcrum location and to allow interaction with the surgeon as well as the patient [14]. To complete the implantation, an optical track gives information on the surgeon's tools location. This sensor recovers data on surgeon's position and orientation for both tools simultaneously.

Once the physical system is described, this work proposes to take gauze to the surgeon's primary tool. Therefore, the goal of this experiment is to find a trajectory in real time between the robot and its target by avoiding the secondary tool. The robot tool is supposed to be already inside the abdomen with gauze on its tool tip. Two situations are considered: one with no movement on the secondary tool and another one with a dynamic behavior of the surgeon.

A. Static Surgeon

As it can be shown in Fig. 6, this situation is solved by the APF behavior. Gauze is carried by the robot at its tool tip. The robot tool trajectory represented by asterisks is linear until it *feels* the secondary tool potential field. Once the robot reaches this zone, it changes the trajectory by surrounding the obstacle at a certain distance (called ρ_0 in section III.A). When the robot bypasses the obstacle zone, it continues the linear trajectory until it reaches the target.

Fig. 6 also shows the velocity commanded to the robot. The APF velocity appears with dashed line, whereas the commanded one is solid. The module of the robot velocity is the maximum possible when its tool is far from the secondary tool, but it shrinks when the distance between the robot tool and the obstacle is closer.

B. Dynamic Surgeon

Fig. 7 shows the resulting trajectory of the robot tool with an asterisk line when the secondary tool moves freely. The secondary tool trajectory is drawn as a series of dots over itself, and has been obtained by optical tracking measurements. The robot not only displaces due to its proximity to the obstacle, but also changes its direction and speed. This way, the resulting trajectory has some zones where the robot has to adapt its trajectory because the secondary tool is very close to the robot tool. The velocity graph of Fig. 7 shows that the velocity in this situation is higher than the one planned by the APF behavior when the robot follows the secondary tool velocity.

V. Conclusion

This work is a first step on auto-guided movements for the robot assistant to find free-obstacle paths to a target location. In order to validate the methodology, a two arm robotic system has been used for implementation purposes. An experiment where the robot should take gauze to a surgeon's tool has been developed with success.

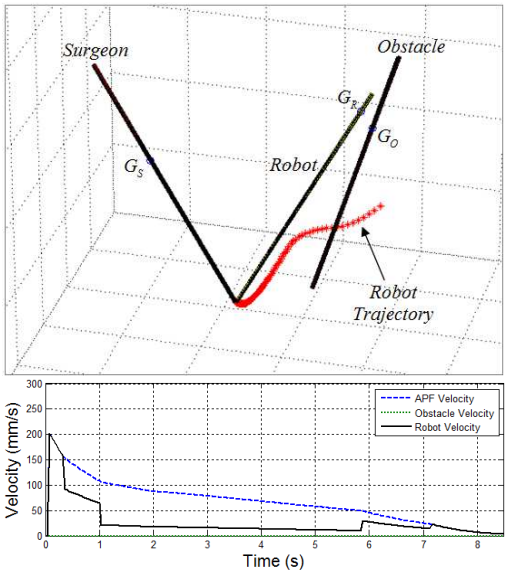

Fig. 6. Robot trajectory when the surgeon's obstacle tool remains static (above) and resulting velocity modules (below).

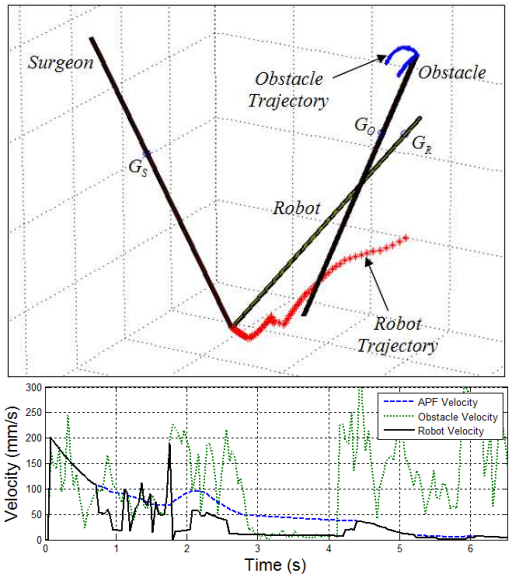

Fig. 7. Robot trajectory when the surgeon's obstacle tool moves (above) and resulting velocity modules (below).

However, as future works the author believes that a deeper interaction between the camera movements and the robot tool would improve auto-guiding tasks. On the other hand, possible collisions with inner tissue must be also avoided in order to generate safe trajectories. Auto-guiding tasks are also very useful for other more complex autonomous maneuvers like knot tying, suture or hold tissue where the robot interacts with both, the surgeon and the patient.

REFERENCES

[1] Rajesh Kumar, Patrick Jensen, Russell H. Taylor, *Experiments with a Steady Hand Robot in Constrained Compliiant Motion and Path Following*. IEEE International Workshop on Robot and Human Interaction, 1999.

[2] P. Hynes, G. I. Dodds, A. J. Wilkinson, *Uncalibrated visual-servoing of a dual-arm robot for surgical tasks*. IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2005.

[3] Krupa, A.; Gangloff, J.; Doignon, C.; de Mathelin, M.F.; Morel, G.; Leroy, J.; Soler, L.; Marescaux, J., *Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing*. IEEE Transactions on Robotics and Automation, 2003.

[4] Gangloff, J.; Ginhoux, R.; de Mathelin, M.; Soler, L.; Marescaux, J., *Model predictive control for compensation of cyclic organ motions in teleoperated laparoscopic surgery*. IEEE Transactions on Control Systems Technology, 2006.

[5] Tobias Ortmaier, Martin Gröger, Dieter H. Boehm, Volkmar Falk, and Gerd Hirzinger, *Motion Estimation in Beating Heart Surgery*. IEEE Transactions on Biomedical Engineering, 2005.

[6] Hyosig Kang; Wen, J.T., *Robotic assistants aid surgeons during minimally invasive procedures*. IEEE Engineering in Medicine and Biology Magazine, vol. 20, no. 1, 2001.

[7] Gang Chen; Minh Tu Pham; Redarce, T., *A semi-autonomous micro-robotic system for Colonoscopy*. IEEE International Conference on Robotics and Biomimetics (ROBIO), 2008.

[8] Ho-seok Song; Ki-young Kim; Jung-ju Lee, *Development of the dexterous manipulator and the force sensor for Minimally Invasive Surgery*. International Conference on Autonomous Robots and Agents (ICARA), 2009.

[9] Dumpert, J.; Lehman, A.C.; Wood, N.A.; Oleynikov, D.; Farritor, S.M., *Semi-autonomous surgical tasks using a miniature in vivo surgical robot*. IEEE International Conference on Engineering in Medicine and Biology Society (EMBC), 2009.

[10] Mayer, H.; Nagy, I.; Knoll, A.; Schirmbeck, E.U.; Bauernschmitt, R., *The Endo[PA]R system for minimally invasive robotic surgery*. IEEE International Conference on Intelligent Robots and Systems (IROS), 2004.

[11] Mayer, H.; Nagy, I.; Burschka, D.; Knoll, A.; Braun, E.U.; Lange, R.; Bauernschmitt, R., *Automation of Manual Tasks for Minimally Invasive Surgery*. International Conference on Autonomic and Autonomous Systems (ICAS), 2008.

[12] Brett, P.N.; Taylor, R.P.; Proops, D.; Coulson, C.; Reid, A.; Griffiths, M.V., *A surgical robot for cochleostomy*. IEEE International Conference of the Engineering in Medicine and Biology Society (EMBS), 2007.

[13] Muñoz, V.F.; Garcia-Morales, I.; Morales, J.; Gomez-DeGabriel, J.M.; Fernandez-Lozano, J.; Garcia-Cerezo, A., *Adaptive Cartesian Motion Control Approach for a Surgical Robotic Cameraman*. IEEE International Conference on Robotics and Automation, 2004.

[14] Bauzano, E.; Munoz, V.F.; Garcia-Morales, I.; Estebanez, B., *Three-layer control for active wrists in robotized laparoscopic surgery*. IEEE International Conference on Intelligent Robots and Systems, 2009.

[15] R. Iraji; M. T. Manzuri-Shalmanit, *A New Fuzzy-Based Spatial Model for Robot Navigation among Dynamic Obstacles*. IEEE International Conference on Control and Automation, 2007.

[16] Enxiu S.; Tao C.; Changlin H.; Enxiu S.; Junjie G., *Study of the New Method for Improving Artifical Potential Field in Mobile Robot Obstacle Avoidance*. IEEE International Conference on Automation and Logistics, 2007.

[17] R. Fernandez; A. Mandow; V. Muñoz; A. Garcia-Cerezo, *Real-Time Motion Control for Safe Navigation*. IFAC Symposium on Intelligent Autonomous Vehicles, 1998.