# A workspace analysis method to support intraoperative trocar placement in minimally invasive robotic surgery (MIRS)

*M. Lohmann, R. Konietschke, A. Hellings, C. Borst, G. Hirzinger*

*DLR, German Aerospace Center, Institute of Robotics and Mechatronics, Oberpfaffenhofen, Germany*

Kontakt: Martin.Lohmann@dlr.de

***Abstract:***

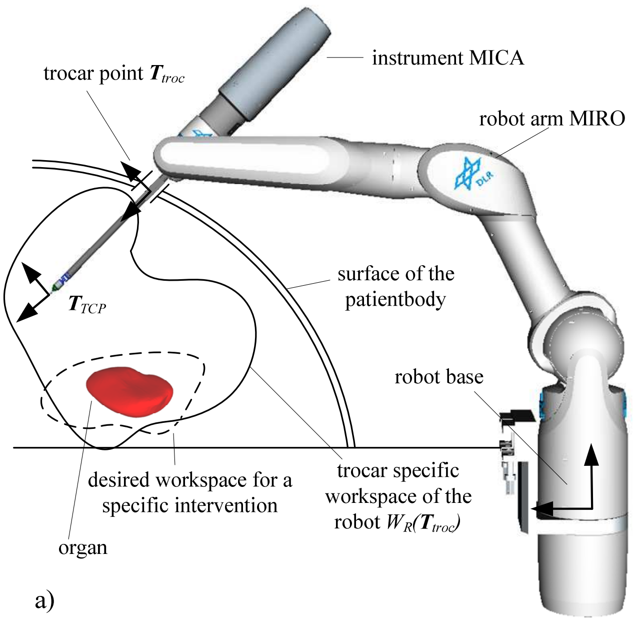
*This paper presents a new method to calculate and display an approximated workspace of a surgical robot in nearly realtime. Displaying this information on a screen in the operation room could support the surgeon during intraoperative trocar placement for teleoperated minimally invasive robotic surgery (MIRS). We give a short overview on existing trocar placement procedures in teleoperated MIRS and describe the possibilities and limitations of workspace analysis methods to support the surgeon during trocar placement. Our new method uses MIRS-specific simplifications to reduce the search space and enable the creation of a reduced workspace map. It was implemented for the DLR MiroSurge system. The implementation can create a reduced workspace map and display a mesh representation of the map in less than 2 seconds. We give a short outlook on how this method could be embedded in trocar placement procedures in the operation theater and what our future plans are with this method.*

*Key words: robotic surgery, minimal invasive surgery, setup planning, workspace analysis*

## 1 Problem

In MIRS interventions one or more instruments which are each guided by a robot arm (in the following "robot" will mean the robot arm including the instrument) are introduced into the patient's body through a trocar. The robot's workspace $W_{rob}$ describes the sum of all the Cartesian 6-dimensional poses of the robot's tool center point (TCP poses, $\boldsymbol{T}_{TCP}$[1]), which can be reached with at least one joint configuration $\boldsymbol{q}$. In MIRS, $W_{rob}$ depends on the position of the trocar $\boldsymbol{T}_{troc}$. Therefore for a robot with trocar kinematics we define $W_{rob}$ as $W_{rob}(\boldsymbol{T}_{troc})$. Choosing proper trocar positions which gain a sufficient reachability for the intended intervention can lead to a very complex problem and exceeding limits of $W_{rob}(\boldsymbol{T}_{troc})$ happens often. Some approaches [1] solve this problem by preoperatively optimizing $\boldsymbol{T}_{troc}$, based on a model of the intervention which is derived e.g. from CT-data of the patient body including the spatial definition of the desired workspace (patient-individual trocar placement). The optimized setup has to be registered and adapted intraoperatively to the real circumstances in the operation room. Other approaches [2] provide a setup standardized for one type of intervention that aims to gain sufficient reachability in most cases (rule of thumb trocar placement). In case of the patient-individual setup, the user can only be sure that the preoperatively planned TCP poses are reachable in the model. This may be inaccurate because reality might differ and the registration of the model to reality comprises sources of error. In case of the rule of thumb setup, the user has to rely on standards which are derived from an average situation and cannot cover every case. Therefore it would be desirable for the user to know $W_{rob}(\boldsymbol{T}_{troc})$ at the moment of trocar placement. With this information it would be possible to estimate, if a desired trocar point offers sufficient reachability for the robot to execute the intended intervention (see figure 1a)). A workspace $W_{rob}$ can be approximated with a workspace map $WM_{rob}$ [3]. This map can e.g. be created brute force by solving the inverse kinematics of all TCP poses defined in a 6-D space with a translational and a rotational resolution and allocating each TCP pose with the information *reachable* or *non-reachable* [3]. With fixed kinematics such a map has to be built only once offline and can be used online as a look-up table for applications such as autonomous task-planning. In case of MIRS, the kinematics depend on $\boldsymbol{T}_{troc}$, so the generation of only one look-up table is not sufficient and a trocar specific $WM_{rob}(\boldsymbol{T}_{troc})$ has to be created.

[1] For spatial description we use homogenous matrices $\boldsymbol{T}^{4x4}$, translational vectors $\boldsymbol{t}^{3x1} = [tx,ty,tz]^T \in IR^3$ and orthonormal rotational matrices $\boldsymbol{R}^{3x3}(\alpha,\beta,\gamma) = [\boldsymbol{Rx}^{3x1}, \boldsymbol{Ry}^{3x1}, \boldsymbol{Rz}^{3x1}]\ (\alpha,\beta,\gamma) \in SO(3)$ (group of all rotations in $IR^3$) with $\alpha,\beta,\gamma$ being rotations around $\boldsymbol{Rx}^{3x1}$, $\boldsymbol{Ry}^{3x1}$, $\boldsymbol{Rz}^{3x1}$. In this paper all spatial descriptions are defined relatively to the robot base.


trocar point $T_{troc}$
instrument MICA
robot arm MIRO
$T_{TCP}$
surface of the patientbody
robot base
desired workspace for a specific intervention
trocar specific workspace of the robot $W_R(T_{troc})$
organ
a)

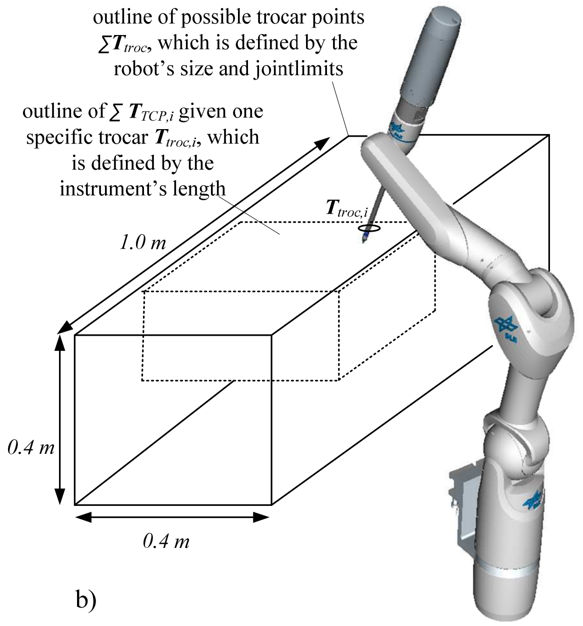

outline of possible trocar points $\sum T_{troc}$, which is defined by the robot's size and jointlimits
outline of $\sum T_{TCP,i}$ given one specific trocar $T_{troc,i}$, which is defined by the instrument's length
$T_{troc,i}$
1.0 m
0.4 m
0.4 m
b)

**Figure 1: a) The robot arm with the minimally invasive instrument MICA [4] inserted into the patient's body through a trocar. b) The outline of a specific $WM_{rob,i}(T_{troc,i})$ compared to the outline of $\sum T_{troc,i}$**

Using brute force sampling, the calculation time to create $WM_{rob}$ of a robot with fixed kinematics as the KUKA LBR is approximately 1 hour with a common PC and $WM_{rob}$ requires memory space of approximately 50 MB [3]. A MIRS specific $WM_{rob}(\boldsymbol{T}_{rob})$ has a smaller outline but needs a higher resolution, than the quoted example. Therefore the calculation time and the memory consumption are comparable. The calculation time is not feasible to create $WM_{rob}(\boldsymbol{T}_{troc})$ online during trocar placement. A method could be to create various maps $\sum WM_{rob,i}(\boldsymbol{T}_{troc,i})$ for all possible trocar points $\sum \boldsymbol{T}_{troc,i}$ to be used as look-up table for trocar planning. As can be seen in figure 1b), for the MiroSurge system $\boldsymbol{T}_{troc}$ can approximately be positioned freely within a volume of *1.0 m x 0.4 m x 0.4 m*. Discretized with *0.01 m*, 160.000 workspace maps would have to be created. The creation and storage of those maps would not be reasonable and create additional costs. Figure 2 shows the visualization of $WM_{rob}$ of a KUKA LBR as introduced by [3]. Voxels are used to display the reachability of the whole volume of $W_{rob}$. This visualization method might not be suitable for the operation room, because it would hide valuable data such as the patient geometry. It is furthermore assumed that not all information about the volume of $W_{rob}(\boldsymbol{T}_{troc})$ is needed by the surgeon. This paper presents a method to create a $WM_{rob}(\boldsymbol{T}_{troc})$ which only describes the outline of $W_{rob}(\boldsymbol{T}_{troc})$. To do so, MIRS-specific simplifications are defined. The aim is to achieve shorter calculation times, less memory consumption and an intuitive visualization of $WM_{rob}(\boldsymbol{T}_{troc})$, so that it can be used for MIRS trocar placement.

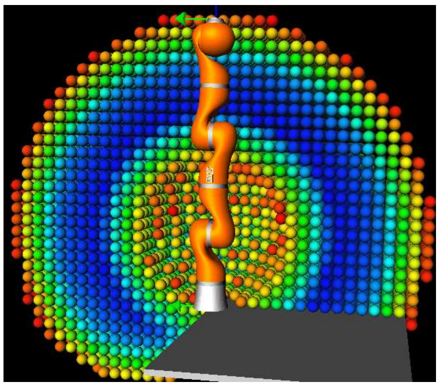


**Figure 2: *$WM_{rob}$ of the KUKA LBR, the percentage of reachable orientations per position is coded in color [3]***

## 2 Methods

### Geometric definition of the reduced MIRS-specific workspace map $WM_{rob}(T_{troc})$

In figure 3a) it can be seen that the TCP can maximally reach the outline of a sphere $HS_1$ if the last three joints (roll, pitch, yaw, see figure 3a)) are restricted to the zero position. This sphere is defined by the length of the instrument $L_{instr}$. By allowing only a movement in the last three joints the system can move the TCP maximally on the surface of the spherical sector $HS_2$, defined by the joint limits of the instrument [5]. For planning autonomous tasks, it is advantageous to have a $WM_{rob}$ approximating the whole volume of $W_{rob}$ translationally and rotationally. A map $WM_{rob}(\boldsymbol{T}_{troc})$ for MIRS does not have to cover all this information. We assume that with some experience a surgeon can approximate the rotational part of $W_{rob}(\boldsymbol{T}_{troc})$ through the movability of the last three joints. Regarding the translational part the important information is the outer border of $W_{rob}(\boldsymbol{T}_{troc})$. Therefore we define the search space to create $WM_{rob}(\boldsymbol{T}_{troc})$ as a discrete space of TCP poses $\boldsymbol{T}_{TCP}$ $(\boldsymbol{t}_{TCP}, \boldsymbol{R}_{TCP})$, which meet the following restrictions:

$$(eq.1-3) \qquad \sqrt{tx_{TCP}^2 + ty_{TCP}^2 + tz_{TCP}^2} = L_{instr} \;; \qquad \boldsymbol{R}z_{TCP} = \left\| \boldsymbol{t}_{TCP} - \boldsymbol{t}_{troc} \right\| ; \quad \boldsymbol{R}x_{TCP} = \boldsymbol{R}x_{troc}$$

In this case only those TCP poses are checked on their reachability, which lie on the outer border of $W_{rob}(\boldsymbol{T}_{troc})$ and which result from a stretched out pose of the instrument.

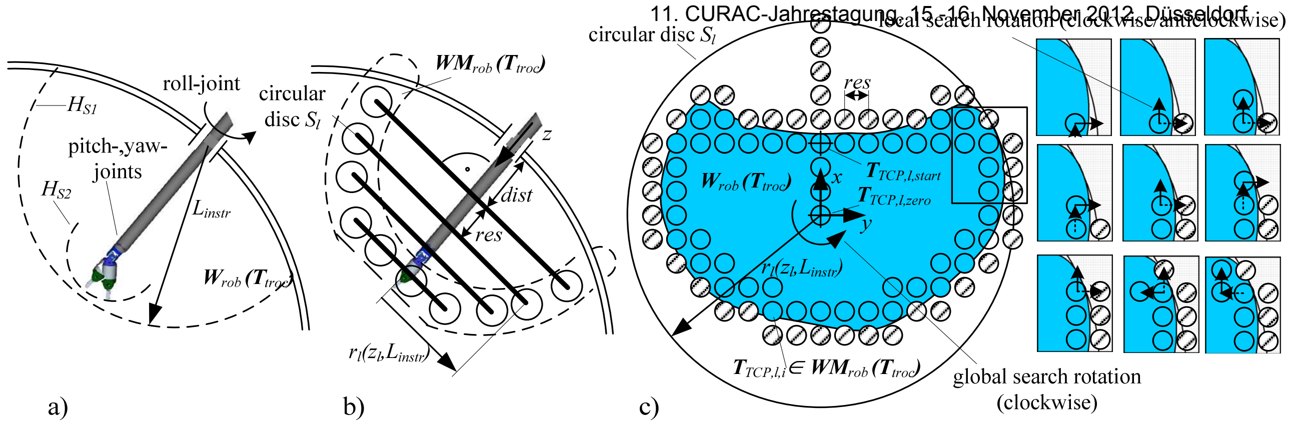


**Figure 3: a), b) Geometric definition of the reduced MIRS-specific $WM_{rob}(T_{troc})$. c) Algorithmic calculation of the reduced MIRS-specific $WM_{rob}(T_{troc})$: All TCP poses $\sum T_{TCP,l,i}$ which lie on the border of $W_{rob}(T_{troc})$ within $S_l$ are stored in $WM_{rob}(T_{troc})$. To identify the border, the search direction always rotates 90° clockwise if $T_{TCP,l,i}$ is reachable and anticlockwise otherwise.**

## Algorithmic calculation of the reduced MIRS-specific workspace map $WM_{rob}(T_{troc})$

To create $WM_{rob}(T_{troc})$ we propose an algorithm that discretizes $WM_{rob}(T_{troc})$ into circular discs $S$ (figure 3b),c)). The discs are defined as orthogonal to $z$ and with a distance of *res* to each other, their radius measures $r(z, L_{instr})$. The steps described in the following are done for all discs $S_l$ (figure 3c)).

In step1, the outer border of $W_{rob}(T_{troc})$ within disc $S_l$ has to be found. Thereby the algorithm starts from a zero position $T_{TCP,l,zero}$, searching in the positive $x$-direction with steps of *res* until it reaches $r$. The last change from reachable to not reachable is identified as the outer border of $W_{rob}(T_{troc})$ and marked as $T_{TCP,l,start}$. If during step1 two borders are found, $W_{rob}(T_{troc})$ includes a reachability hole within $S_l$. In this case the algorithm deletes all previously found TCP poses from $WM_{rob}(T_{troc})$ and continues with the next disc $S_{l+1}$. This is done to gain a conservative estimation of the workspace by only representing an outline of a fully reachable volume in $WM_{rob}(T_{troc})$ . To increase the chance of finding reachability holes, step1 can be repeated with different search directions. In step 2 all reachable TCP poses $\sum T_{TCP,l,i}$ along the border of $W_{rob}(T_{troc})$ within $S_l$ are identified. Thereby the global search direction is anticlockwise, which means that the border of $W_{rob}(T_{troc})$ is always assumed on the right side. Starting with the TCP pose $T_{TCP,l,i}$ which is left of $T_{TCP,l,start}$ the algorithm will always turn its local search direction 90° clockwise if the current $T_{TCP,l,i}$ is reachable. If the algorithm comes to a $T_{TCP,l,i}$ which is not reachable, it will step back to the last reachable TCP pose $T_{TCP,l,i-1}$ and turn its local search direction 90° anticlockwise.

The difference in calculation complexity between this method and a brute force approach can be compared as followed. For this method the function to gain the number of necessary inverse kinematics calculations $f_{invkin}$ is defined as $f1_{invkin}(n_1, n_2, n_3) = (n_1+n_2)\cdot n_3$. Thereby $n_1$ and $n_2$ are the amounts of TCP poses for which the inverse kinematics have to be calculated within the described steps 1 and 2 and $n_3$ is the amount of the discs $\sum S_l$ . For $n_1$ and $n_3$ we can define $n_1=n_3=L_{instr}/res$ (as a simplification we set $r = L_{instr}$), but for $n_2$ only an approximation for the maximal number of calculations $n_{2,max,approx}$ can be defined. As can be seen in figure 3c), the algorithm has to calculate the inverse kinematics for approximately every $T_{TCP,l,i}$ on the inside and on the outside of the border of $W_{rob}(T_{troc})$ within $S_l$. The maximal borderlength is approximated with $2\cdot L_{instr}\cdot\pi$. Because $n_1$, $n_2$ and $n_3$ all depend on $L_{instr}$ and *res*, the equation yields:

$$(eq.4)\qquad f1_{invkin}\left(L_{instr}, res\right) = \left(\frac{L_{instr}}{res} + 2\cdot\frac{2\cdot L_{instr}\cdot\pi}{res}\right)\cdot\frac{L_{instr}}{res} = \left(1+4\cdot\pi\right)\cdot\left(\frac{L_{instr}}{res}\right)^2$$

If in contrast the map $WM_{rob}(T_{troc})$ is created with a brute force approach as described by [3], the function $f_{invkin}$ to gain the number of necessary inverse kinematics calculations can be defined as: $f2_{invkin}(n,m) = n^3\cdot m^3$

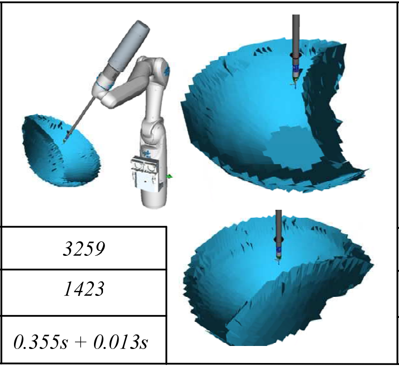
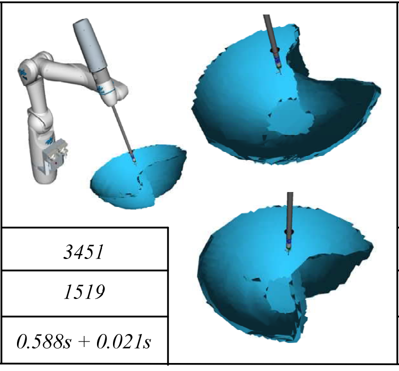
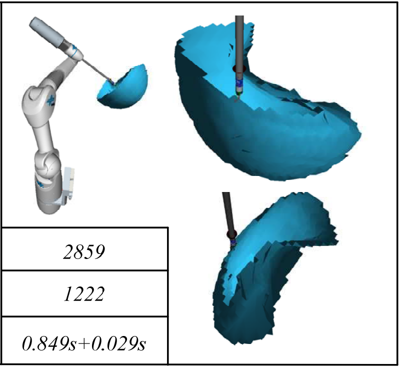
Thereby $n$ is the number of translational steps per axis and $m$ is the number of rotational steps per axis. They both define the number of TCP poses $\sum T_{TCP,i}$ which have to be checked with respect to their reachability. For the creation of a map $WM_{rob}(T_{troc})$ for MIRS, the number of translational steps $n$ is defined by $L_{instr}$ and *res* as approx. $n=L_{instr}/res$ (figure 1). Therefore the growth rate $O$ of $f1_{invkin}(L_{instr}, res)$ is smaller than the growth rate of $f2_{invkin}(L_{instr}, res, m)$:

$$(eq.5)\qquad f1_{invkin}\left(L_{instr}, res\right)\in O\left(\left(\frac{1}{res}\right)^2\right);\qquad f2_{invkin}\left(L_{instr}, res, m\right)\in O\left(\left(\frac{1}{res}\right)^3\cdot m^3\right)$$

## 3 Results and Discussion

We created workspace maps $WM_{rob}(\boldsymbol{T}_{troc})$ for three different trocar points with the parameters $res = 0.01\ m$, $L_{instr} = 0.2\ m$. Table 1 shows meshes of the workspace maps, the amount of inverse kinematics calculations $n_{invkin}$, the amount of points identified as reachable $n_{WM}$, and the mean overall time $t_{calc}$ of one calculation (Prozessor: Intel(R) Xeon(R) CPU W3530, 2.80GHz, 6GB main memory). The mean was determined over 100 repetitions.

**Table 1: The meshes $WM_{rob}(T_{troc})$ of three different trocar points, with the number of inverse kinematics calculations, the number of TCP-poses identified as reachable and the mean calculation time.**

| mesh of $WM_{rob}$ |  |  |  |
|---|---|---|---|
| $n_{invkin}$ | *3259* | *3451* | *2859* |
| $n_{WM}$ | *1423* | *1519* | *1222* |
| $\overline{t}_{calc}+\sigma_{calc}$ | $0.355s + 0.013s$ | $0.588s + 0.021s$ | $0.849s+0.029s$ |

The number of inverse kinematics calculations stayed under the maximum of 5426, calculated with (eq. 4). The time $t_{calc}$ was always smaller than $1\ s$. As a calculation of one inverse kinematics solution of our system takes about $20\ \mu s$ [1], only around 10 % of $t_{calc}$ is caused by the inverse kinematics calculations. The remainder of $t_{calc}$ is used by the algorithm for other operations. In the future we hope to optimize this remainder of $t_{calc}$ to gain faster calculation times. The overall time to create and mesh $WM_{rob}(\boldsymbol{T}_{troc})$ was measured between $1\text{-}2\ s$, which allows to create the map online in the operation room. The mesh to display $WM_{rob}(\boldsymbol{T}_{troc})$ is done with an algorithm that deforms the mesh of a half sphere and which is implemented in open scene graph. As can be seen in table 1, the meshing quality is not optimal and will be improved by a nearest neighbour algorithm. The use of displaying $WM_{rob}(\boldsymbol{T}_{troc})$ during trocar placement, is shown in figure 4. Here, the surgeon uses the robot in the hands-on-mode [4] to measure the position of the desired trocar point. The calculated $WM_{rob}(\boldsymbol{T}_{troc})$ can be displayed in a virtual copy of the scene to evaluate e.g. the overlap of $WM_{rob}$ with a registrated organ. If the overlap is not satisfying for the intended task, the procedure can be repeated until a suitable trocar point is found. Thereby the risk of coming across workspace borders which are caused by wrongly chosen trocar points might be reduced. We will evaluate the use of the described method for trocar placement with surgeons within the next six months. Moreover we will use the map for setup optimization methods and augmentation of the endoscopic picture.

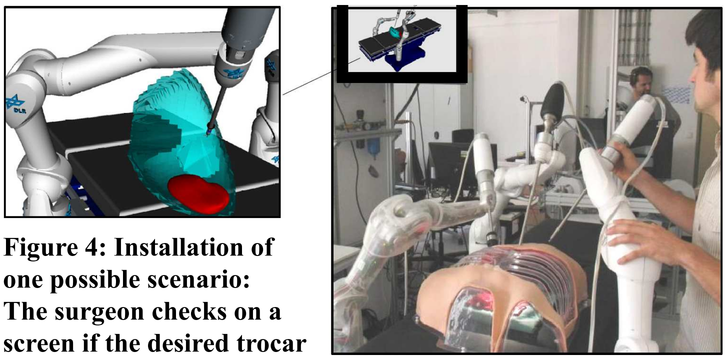


**Figure 4: Installation of one possible scenario: The surgeon checks on a screen if the desired trocar is suitable to reach e.g. a registrated organ. To measure position of the trocar he uses the robot in the hands-on-mode.**

## 4 References

[1] Konietschke, R.: "*Planning of Workplaces with Multiple Kinematically Redundant Robots*", Dissertation thesis, Technische Universität München, 2008

[2] Stolzenburg, J.U. et. al..: „*3.3 Setup of da Vinci Robot for Pelvic Surgery*", Laparoscopic and Robot-Assisted Surgery in Urology: Atlas of Standard Procedures, Springer Verlag, 2010

[3] Zacharias, F.: "*Knowledge Representations for Planning Manipulation Tasks*", Springer Verlag, 2012

[4] Hagn, U et. al.: „*DLR MiroSurge – Towards Versatility in Surgical Robotics*", Proceedings of CURAC, 2008

[5] Thielmann, S. et. al.: „*MICA--A new generation of versatile instruments in robotic surgery*", IEEE International Conference on Intelligent Robots and Systems (IROS), 2010