

Perception, control and path planning of robotic laparoscopic surgical system

Alexios Karadimos

Supervisor: Associate Professor, **Evangelos Dermatas**
Co-Supervisor: Professor, **Anthony Tzes**

Department of Electrical and Computer Engineering

Patras, February 2022



Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

Surgical Robotics Procedure

Advantages & Disadvantages of Surgical robotics

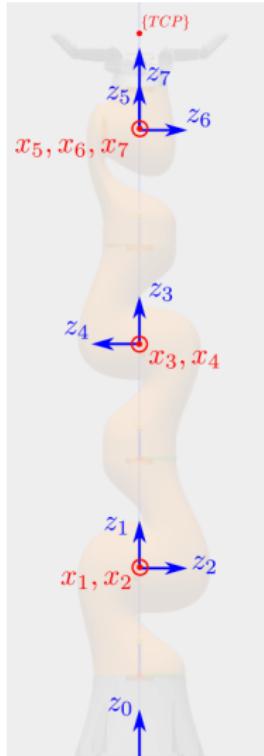
Bibliography overview

Thesis goals

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

Forward Kinematics



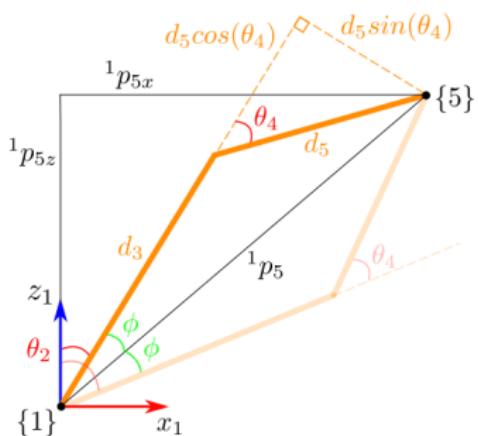
i	θ_i (rad)	L_{i-1} (m)	d_i (m)	α_{i-1} (rad)
1	θ_1	0	0.36	0
2	θ_2	0	0	$-\pi/2$
3	θ_3	0	0.36	$\pi/2$
4	θ_4	0	0	$\pi/2$
5	θ_5	0	0.4	$-\pi/2$
6	θ_6	0	0	$-\pi/2$
7	θ_7	0	0	$\pi/2$

Table: D-H parameters for KUKA iiwa14

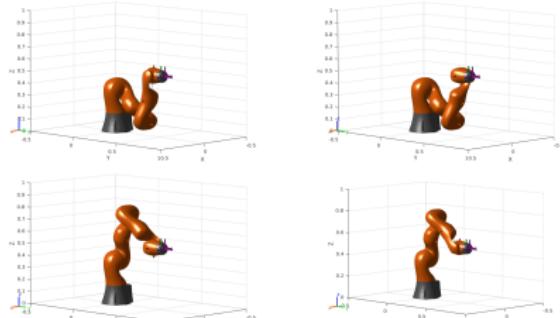
$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & L_{i-1} \\ s\theta_i c a_{i-1} & c\theta_i c a_{i-1} & -s a_{i-1} & -s a_{i-1} d_i \\ s\theta_i s a_{i-1} & c\theta_i s a_{i-1} & c a_{i-1} & c a_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics - Decoupling Technique

Solve separately for position and orientation (assume $\theta_3 = 0$)



$$R_{target} = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}$$



$${}^0\mathbf{p}_5 = {}^0T_4 {}^4\mathbf{p}_5 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

Figure: The first 4 out of 8 solutions of the IK-problem

Singularity points

RCM constraint

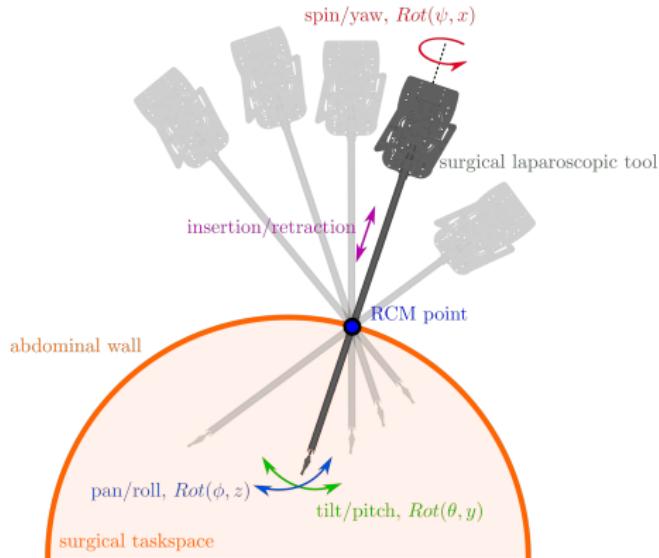


Figure: Illustration of pivoting motion of surgical laparoscopic tool around RCM point (also known as fulcrum or trocar point). Due to the RCM constraint, the tool has only 4 degrees of freedom.

Elbow-up constraint

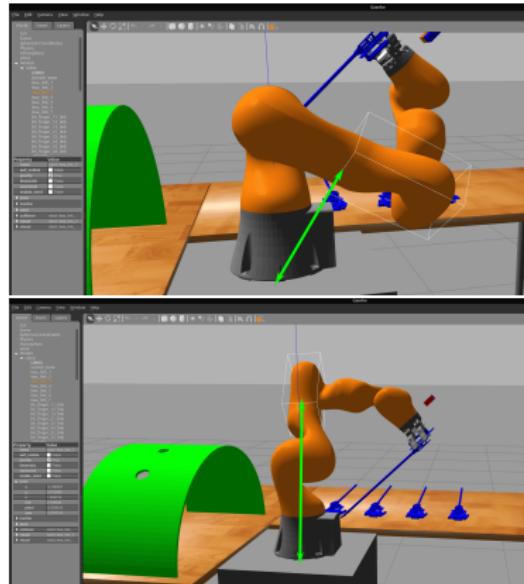


Figure: Top: elbow-down solution, bottom: elbow-up solution

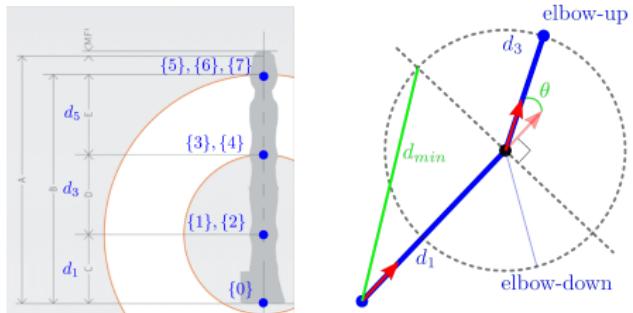


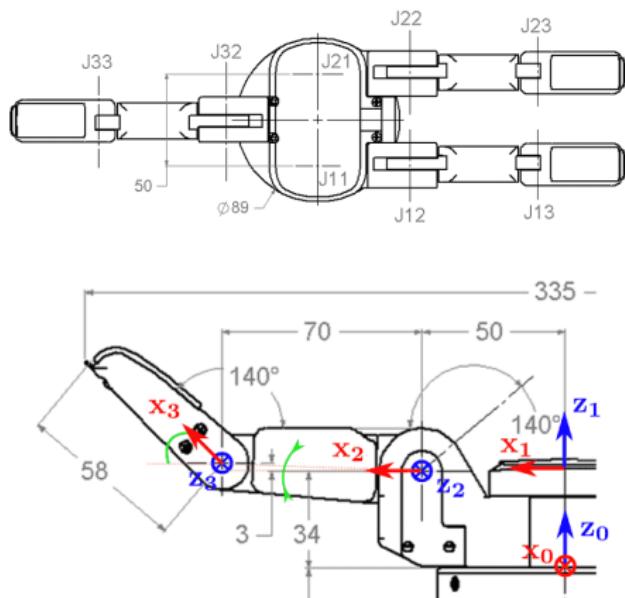
Figure: Elbow-up constraint description with relative distance or angle between links with lengths d_1 and d_3

$$d_{\min} \leq d \leq d_{\max}, \text{ where}$$
$$d_{\min} = \sqrt{d_1^2 + d_3^2} = 553\text{mm} \text{ and}$$
$$d_{\max} = d_1 + d_3 = 780\text{mm}.$$

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

Gripper & Forward Kinematics



i	θ_i (rad)	L_{i-1} (m)	d_i (m)	α_{i-1} (rad)
1 (J11)	$\theta_{11} - \pi/2$	0.025	0.0034	0
2 (J12)	$\theta_{12} + 0.04$	0.05	0	$\pi/2$
3 (J13)	$\theta_{13} + 0.69$	0.07	0	0
i	θ_i (rad)	L_{i-1} (m)	d_i (m)	α_{i-1} (rad)
1 (J21)	$\theta_{21} - \pi/2$	0.025	0.0034	0
2 (J22)	$\theta_{22} + 0.04$	0.05	0	$\pi/2$
3 (J23)	$\theta_{23} + 0.69$	0.07	0	0
i	θ_i (rad)	L_{i-1} (m)	d_i (m)	α_{i-1} (rad)
1	$\pi/2$	0	0.0034	0
2 (J32)	$\theta_{32} + 0.04$	0.05	0	$\pi/2$
3 (J33)	$\theta_{33} + 0.69$	0.07	0	0

Gripper Inverse Kinematics

Standard solutions of a RRR kinematic chain, using the law of cosines

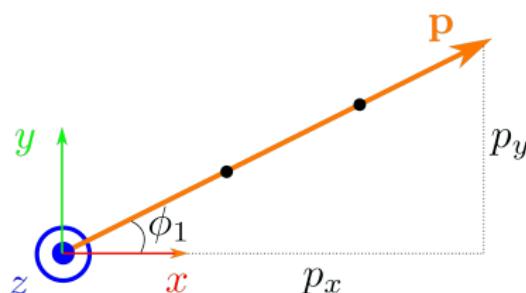


Figure: top view

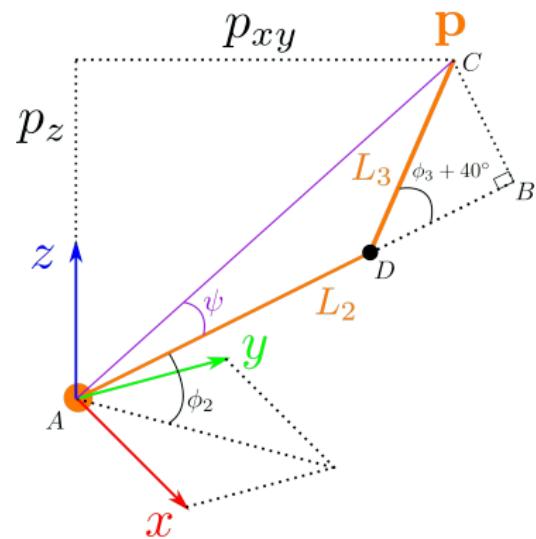
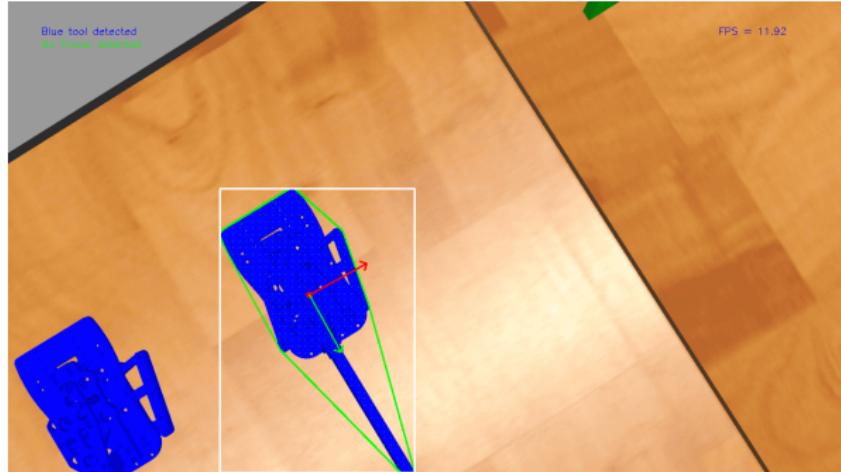


Figure: side view

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

Tool detection & pose estimation



a, b: orientation vectors,
solutions of

$$\mathbf{Cv} = \lambda \mathbf{v}$$

$$\mathbf{C} = \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix}$$

Figure: Tool's ROI, convex hull, center of mass and orientation vectors

$$(\bar{x}, \bar{y}) = \left(\frac{1}{N} \sum_{i=1}^N x_i, \frac{1}{N} \sum_{i=1}^N y_i \right)$$

$$\sigma(x, y) = \frac{1}{n-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

Calculation of grasping points

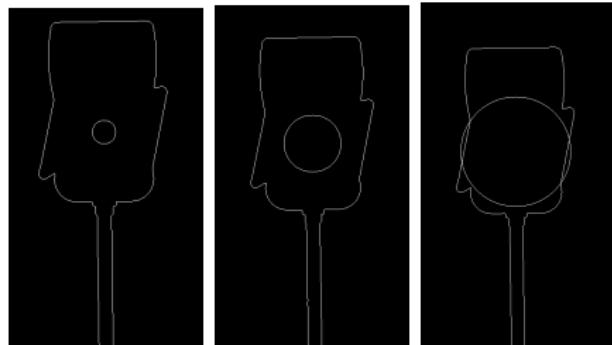


Figure: Finding candidate grasping points from the intersections of a growing circle mask $I_1(x, y)$ and the contour of the detected surgical tool $I_2(x, y)$

- ➊ $\mathbb{G} = \arg \max_{(x,y)} I_1(x, y) \odot I_2(x, y)$
- ➋ match points with stereo 3d points
- ➌ check feasibility with gripper kinematics & force closure
- ➍ repeat (grow circle) until all conditions are met

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

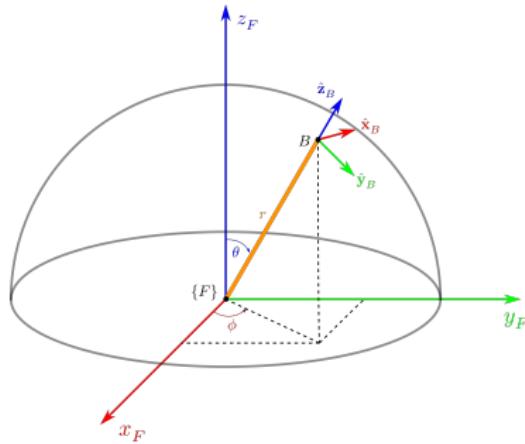
Path Planning - Sampling methods

Task space analysis

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

Tool pose



$${}^F T_B = \begin{bmatrix} {}^F R_B & {}^F \mathbf{p}_B \\ \mathbf{0} & 1 \end{bmatrix}$$

where

$${}^F R_B = [\hat{\mathbf{x}}_B \quad \hat{\mathbf{y}}_B \quad \hat{\mathbf{z}}_B]$$

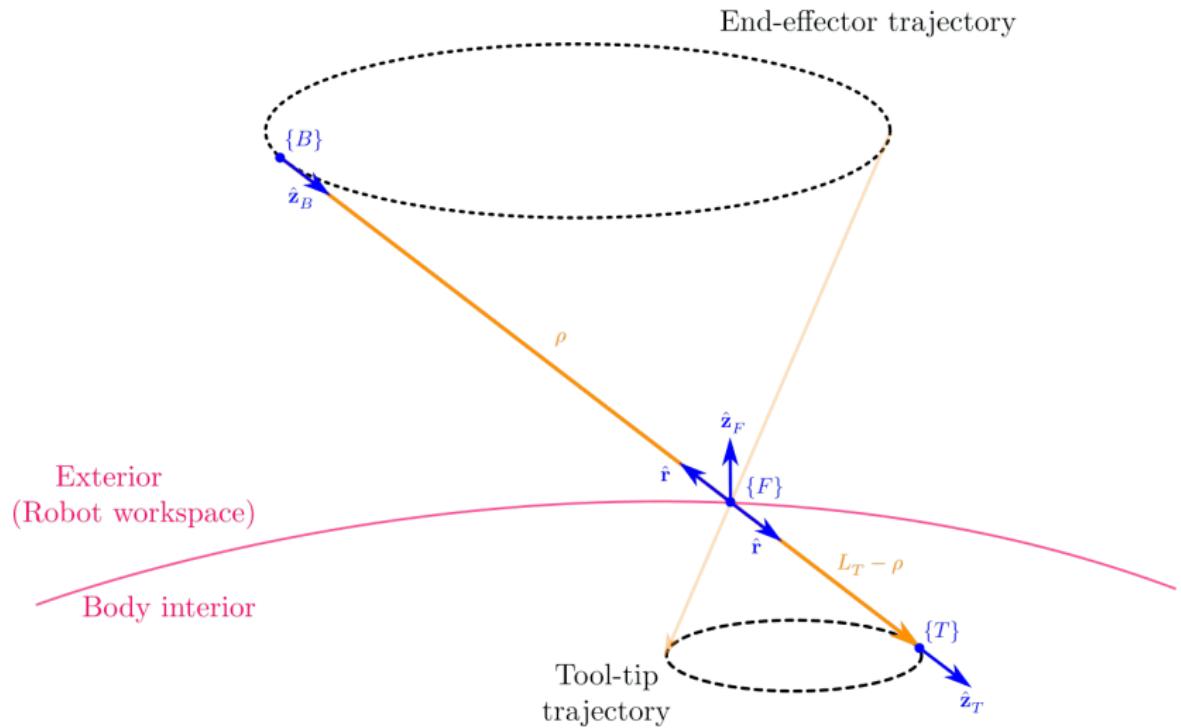
Calculate orientation vectors using spherical coordinate unit vectors

$$\hat{\mathbf{x}}_B = \hat{\phi} = \begin{bmatrix} -\sin(\varphi) \\ \cos(\varphi) \\ 0 \end{bmatrix}$$

$$\hat{\mathbf{y}}_B = \hat{\theta} = \begin{bmatrix} \cos(\theta) \cos(\varphi) \\ \cos(\theta) \sin(\varphi) \\ -\sin(\theta) \end{bmatrix}$$

$$\hat{\mathbf{z}}_B = \hat{\mathbf{r}} = \begin{bmatrix} \sin(\theta) \cos(\varphi) \\ \sin(\theta) \sin(\varphi) \\ \cos(\theta) \end{bmatrix}$$

The Fulcrum Effect



Line segment trajectory of tool tip

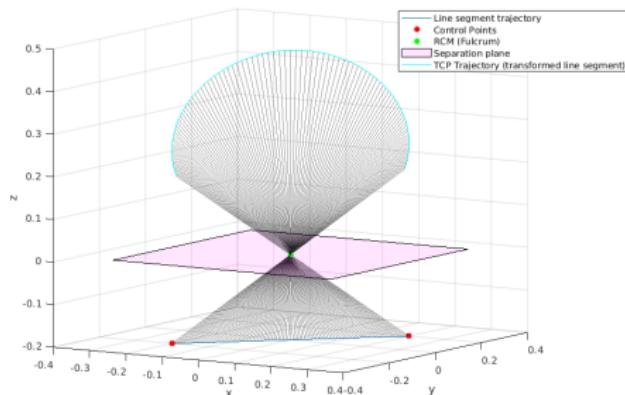
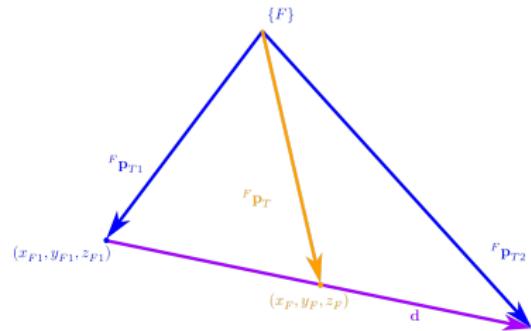


Figure: A Line segment trajectory and it's transformation due to the Fulcrum Effect



$$\begin{cases} x_F = (1-s)x_{F1} + sx_{F2} \\ y_F = (1-s)y_{F1} + sy_{F2} \\ z_F = (1-s)z_{F1} + sz_{F2} \end{cases}$$

Circular trajectory of tool tip

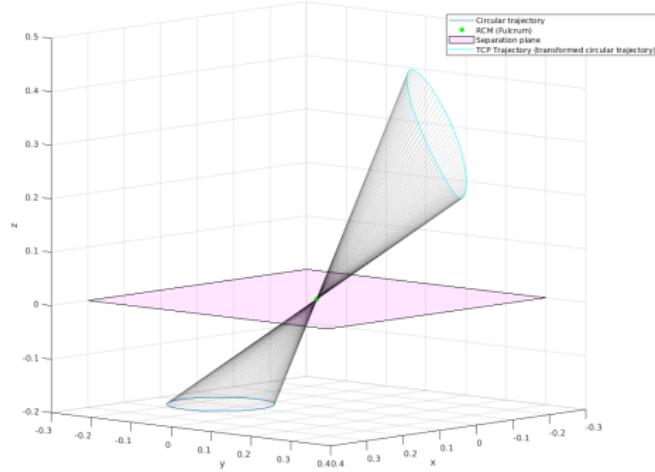


Figure: Circular trajectory of tool tip with respect to Fulcrum reference frame and it's transformation via the Fulcrum Effect

$$\begin{cases} x_F = r_0 \cos(2\pi s) + x_{F0} \\ y_F = r_0 \sin(2\pi s) + y_{F0} \\ z_F = z_{F0} \end{cases},$$

$$s \in [0, 1]$$

$$\begin{cases} r = \sqrt{x_F^2 + y_F^2 + z_F^2} \\ \theta = \text{atan2}\left(\sqrt{x_F^2 + y_F^2}, z_F\right) \\ \varphi = \text{atan2}(y_F, x_F) \end{cases}$$

Circular trajectory of tool tip

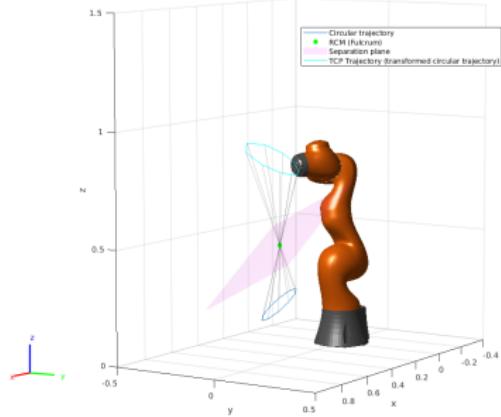


Figure: Circular trajectory that lies on an a plane of arbitrary orientation with respect to the fulcrum point

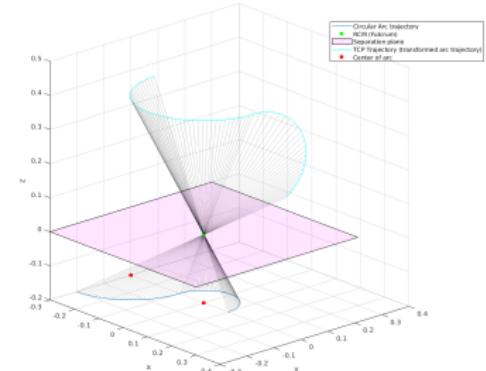
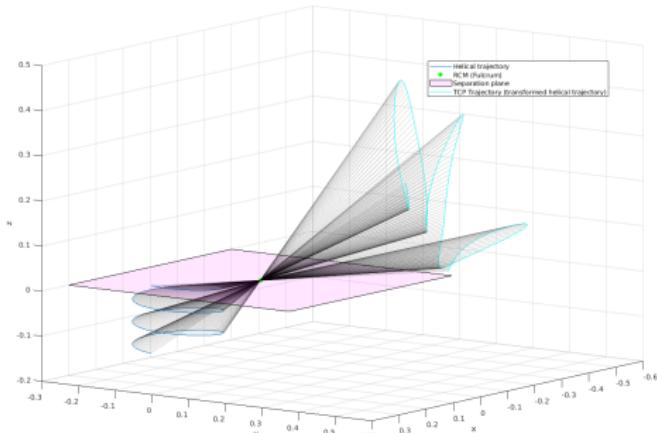


Figure: Circular arc trajectory of tool tip with respect to Fulcrum reference frame and it's transformation via the Fulcrum Effect. In this trajectory 2 circular arcs are used

Helical trajectory of tool tip



$$\begin{cases} x_F = r_0 \cos(2\pi s) + x_{F0} \\ y_F = r_0 \sin(2\pi s) + y_{F0} \\ z_F = \pm \beta s \end{cases}$$

$$s \in [0, \tau]$$

τ : cycles

β/r_0 : slope (aka pitch)

Figure: Helical trajectory of tool tip with respect to Fulcrum reference frame and it's transformation via the Fulcrum Effect

Cubic Spline trajectory of tool tip

$$x_i(s) = a_i(s - s_i)^3 + b_i(s - s_i)^2 + \\ + c_i(s - s_i) + d_i$$

$$s_i \leq s \leq s_{i+1}$$

boundary conditions $\Rightarrow a_i, b_i, c_i, d_i \in \mathbb{R}$

$$x_i(s_i) = x_i$$

$$x_i(s_{i+1}) = x_{i+1}$$

$$\dot{x}_i(s_i) = \dot{x}_i$$

$$\dot{x}_i(s_{i+1}) = \dot{x}_{i+1}$$

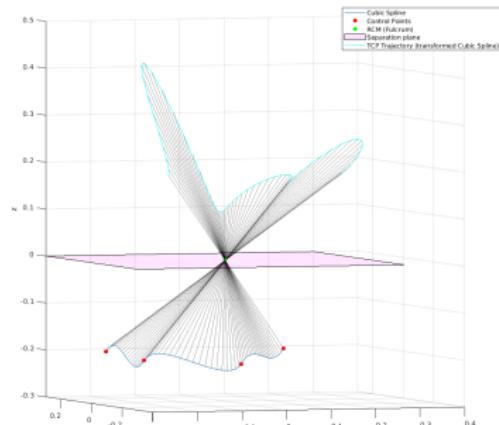


Figure: A Cubic Spline trajectory and it's transformation via the Fulcrum Effect

B-Spline trajectory of tool tip

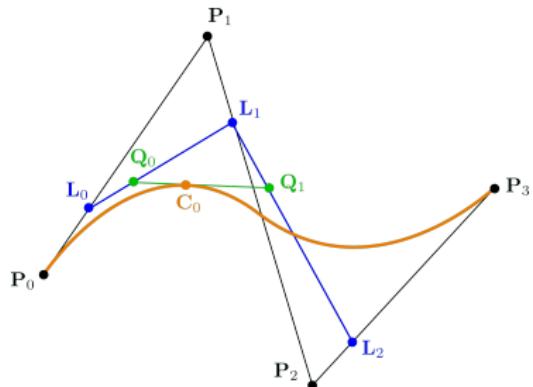


Figure: Cubic Bézier curve
calculated using cubic
interpolation of 4 control points

$$\mathbf{L}_0(s) = (1 - s)\mathbf{P}_0 + s\mathbf{P}_1$$

$$\mathbf{Q}_0(s) = (1 - s)\mathbf{L}_0(s) + s\mathbf{L}_1(s)$$

$$\mathbf{C}_0(s) = (1 - s)\mathbf{Q}_0(s) + s\mathbf{Q}_1(s)$$

$$\begin{aligned}\mathbf{C}_0(s) = & (1 - s)^3 \mathbf{P}_0 + 3(1 - s)^2 s \mathbf{P}_1 + \\ & + 3(1 - s)s^2 \mathbf{P}_2 + s^3 \mathbf{P}_3\end{aligned}$$

B-Spline trajectory of tool tip

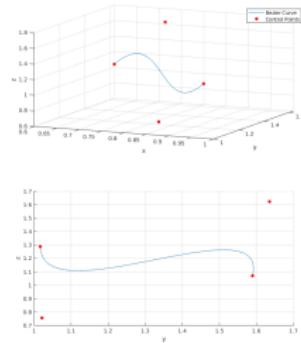


Figure: A cubic Bézier curve calculated and plotted in MATLAB

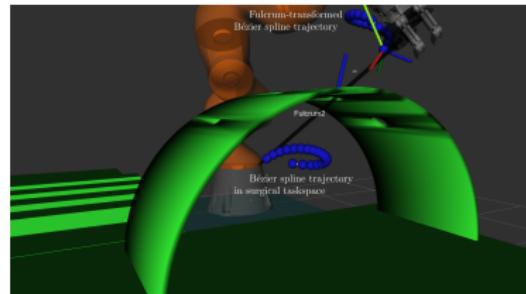
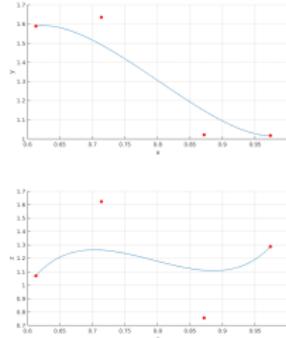


Figure: Experiment 3d: Create the B-spline trajectory inside the surgical site (below the green mounting dock) and transform it via the fulcrum transformation to a trajectory for the robot's TCP.

Trajectory planning in joints' space

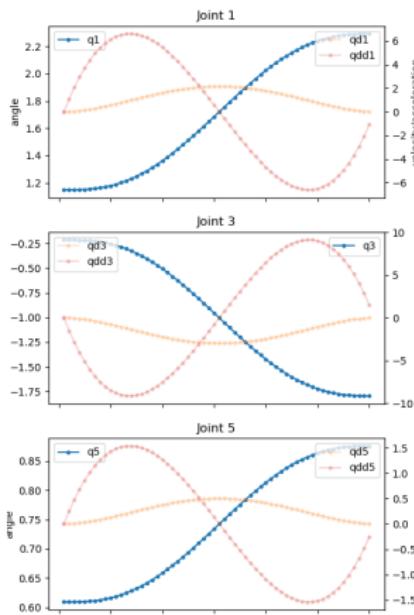


Figure: Quintic polynomial trajectories

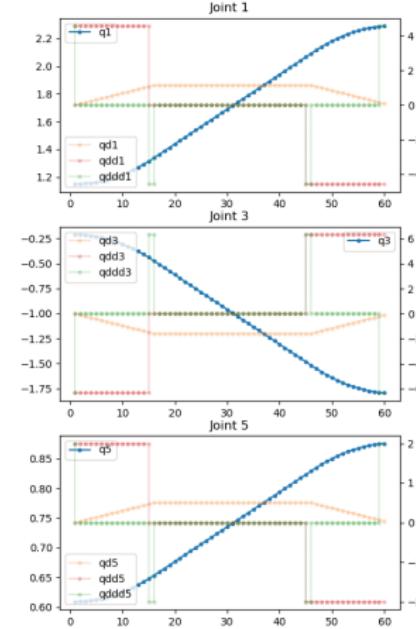


Figure: Trajectories with trapezoid velocity profile

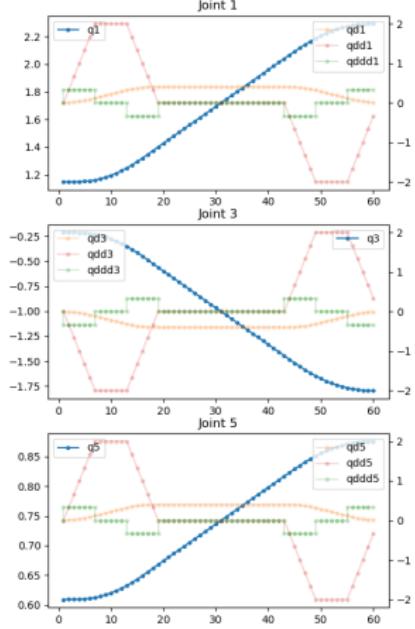
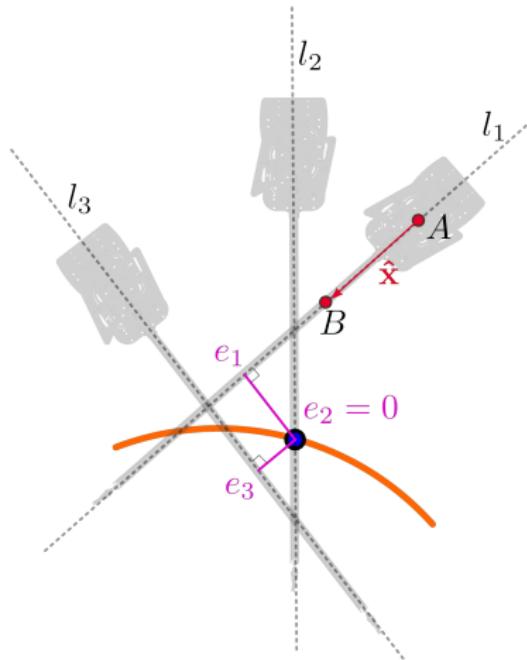


Figure: Trajectories with s-curve velocity profile

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

RCM Tracking



$${}^U T_{T0} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\overrightarrow{O_F A} = \mathbf{p}, \quad \text{and} \quad \overrightarrow{O_F B} = \mathbf{p} + \hat{\mathbf{x}}$$

$$e_{rcm} = d(l, O_F)$$

$$d(l, O_F) = \frac{\|\overrightarrow{O_F A} \times \hat{\mathbf{x}}\|}{\|\hat{\mathbf{x}}\|}$$

Figure: Geometric calculation of the RCM alignment error e using the distance between the line l and the RCM point.

RCM Tracking

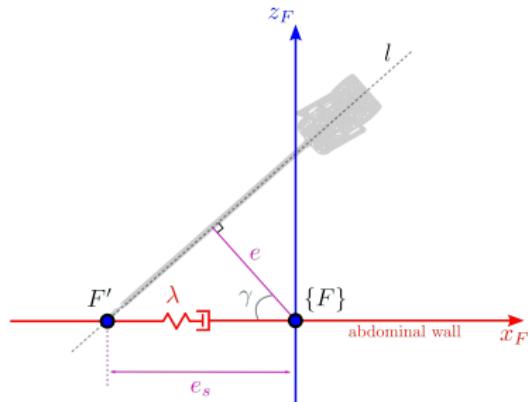


Figure: Force interaction model of the laparoscopic tool and the abdominal wall around the fulcrum point (RCM point)

$$\|\mathbf{f}_s\| = \frac{\lambda}{\cos\gamma} e$$

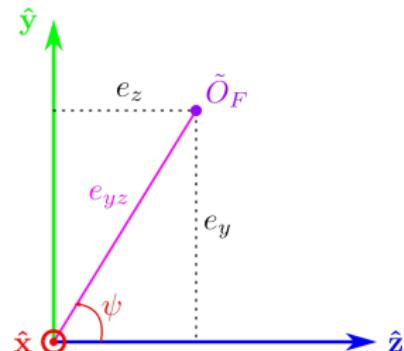


Figure: RCM error calculation in yz plane. The RCM error or yz -error is the distance between the line of the \hat{x} vector (here seen as a point) and the estimated position of the origin of the fulcrum reference frame \tilde{O}_F .

RCM Tracking

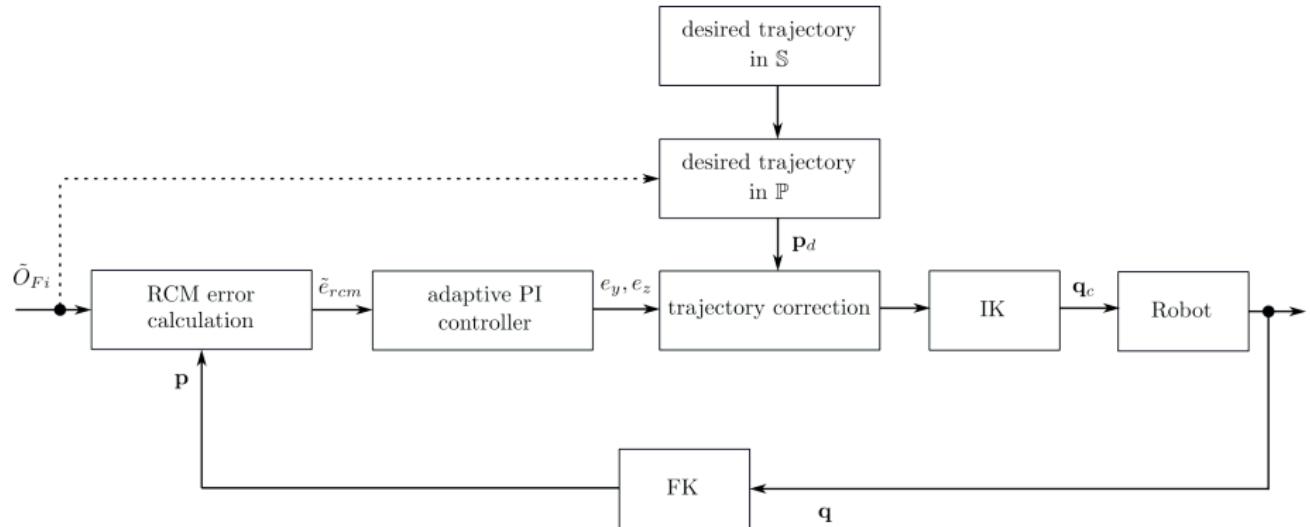


Figure: RCM tracking proposed control system. The RCM error is used as input in the trajectory generator to correct the trajectory command in order to fix the RCM misalignment

Image based visual servoing

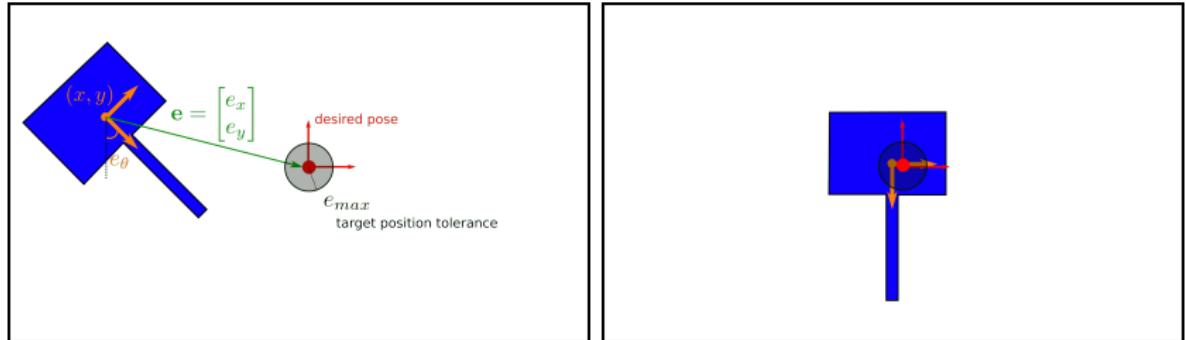


Figure: Image based visual servoing. The robot arm is controlled using the information gained from the video frames. The frames are 2-Dimensional and thus the detected objects can have only 3 degrees of freedom which means we can mainly control 3 independent variables, here the x, y, θ variables. The left image is the initial frame and the right image is the frame where the object is at the target pose.

$$\mathbf{e}[k] = [e_x, e_y, e_\theta]^\top$$

Image based visual servoing

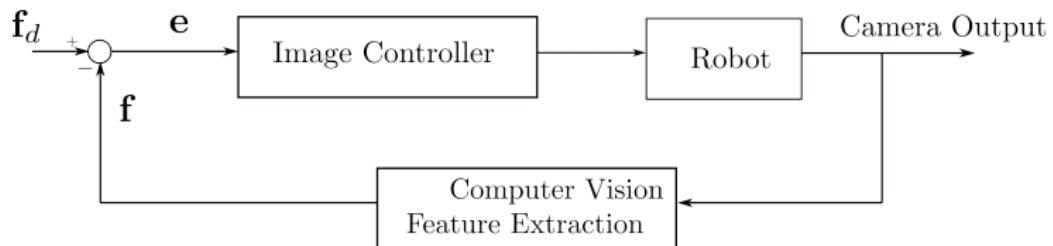


Figure: Image based visual servoing closed loop control

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \mathbf{u}[k]$$

$$\mathbf{u}[k] = K_p \mathbf{e}[k] + K_i \sum_{i=0}^{k-1} \mathbf{e}[i] + K_d (\mathbf{e}[k] - \mathbf{e}[k-1])$$

Firm grasping algorithm & Force control

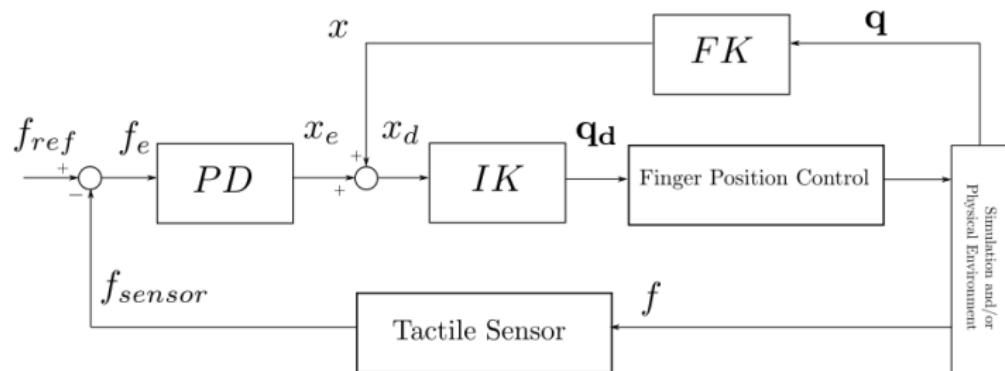


Figure: Force control on a Barrett Hand gripper finger

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

ROS framework

Key components:

- Nodes
- Topics & Messages
- Parameters
- Launch files
- Packages
- ROS filesystem, network, tools & community
- ...

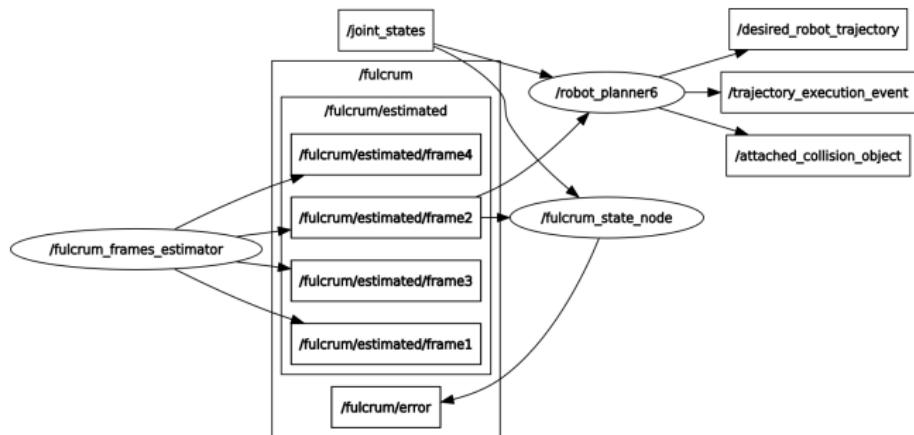


Figure: Subset of ROS nodes and topics used for the robot-planner6 experiment

Gazebo & RViz

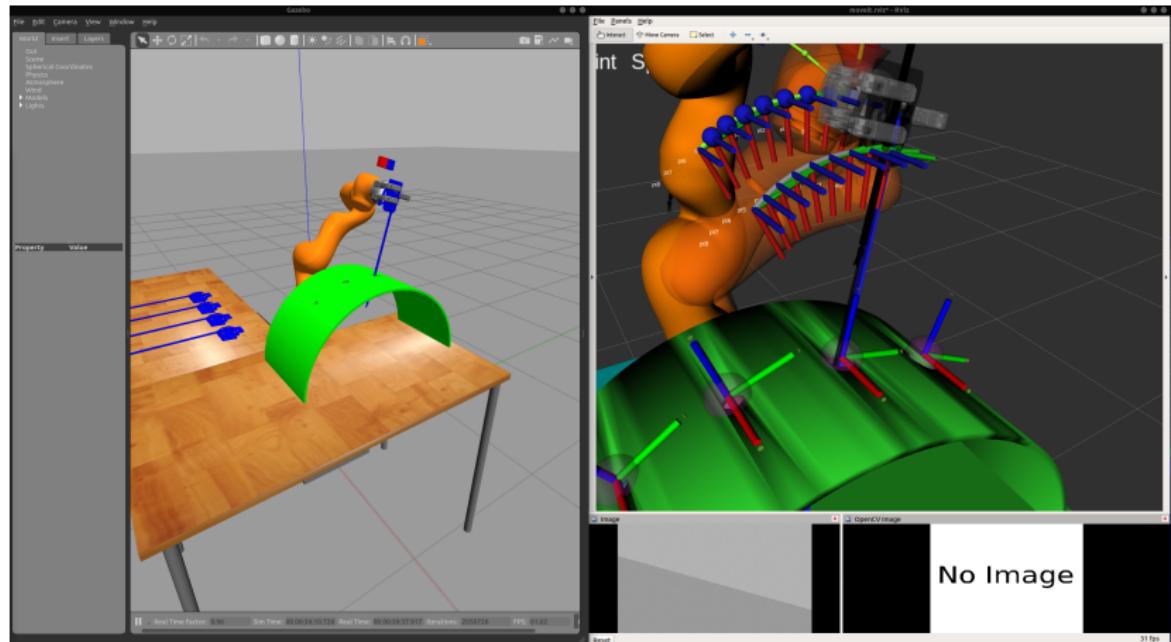


Figure: Simulation with Gazebo (physics engine, sensors, graphics etc.) & visualization with RViz (visualize robot state, known data such as topics or sensor measurements, markers, geometric objects etc.)

Motion Planning with Moveit

Moveit motion planning parameter values outside and inside of surgical site:

- **Position tolerance:** 50 - 500 μm , 5 μm
- **Orientation tolerance:** 0.00005deg, 0.000005deg
- **Maximum planning time:** 5-10s, 5s
- **Replanning allowed:** true
- **End-effector interpolation step:** 1mm
- **Base frame:** world (universal)
- **Jump threshold**
- **Planner algorithm:** RRTConnect
- **Maximum planning attempts:** 6
- **Fraction**

Tools, Packages and Libraries

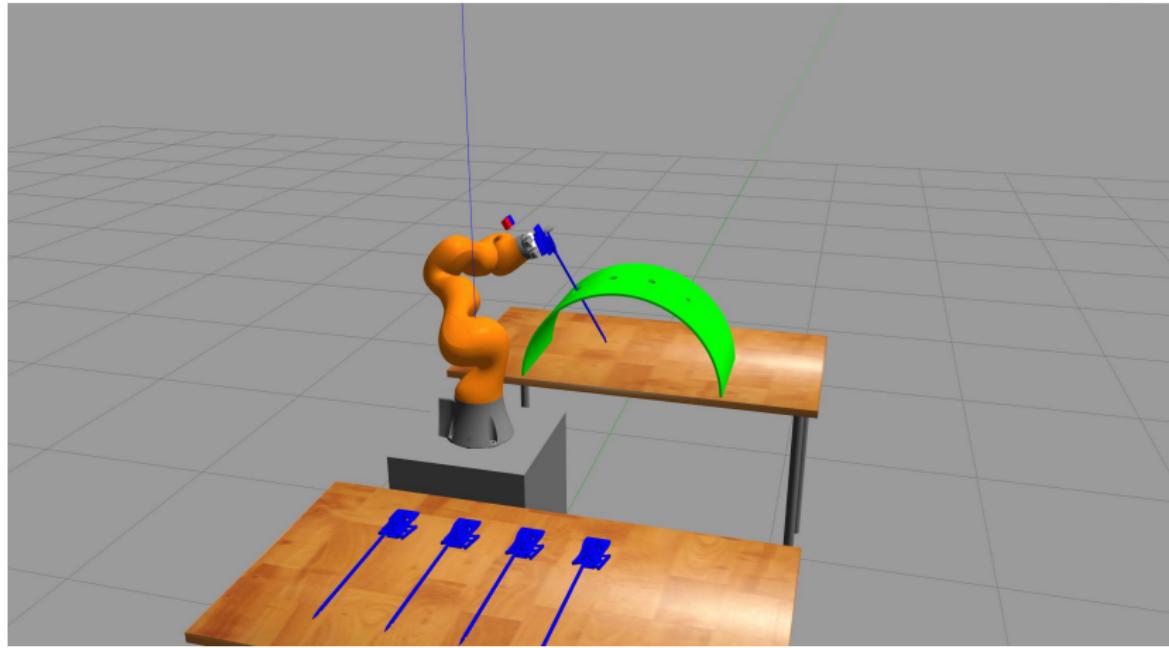
- **tf2**: keep track of multiple coordinate frames, apply transformations
- **geometry_msgs**
- **Eigen**: linear algebra
- **OpenCV2**: computer vision
- **numpy**
- **actionlib**
- State machines with **Smach**
- ros-industrial/kuka_experimental
- **barrett_hand**
- **gazebo-pkgs**
- **moveit-pkgs**
- ...

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

Robot Planner 1: Simple MoveIt planning

Goal: Benchmark MoveIt Path planning algorithms



Robot Planner 1: Simple MoveIt planning

RRTConnect				
Insertion & Pivot trajectories				
10 Experiments	Insertion planning time	Execution status	Pivot planning time	Execution status
Average	0.163102	1	0.5014398	0.9
Standard deviation	0.110001	-	1.110582	-

RRT*				
Insertion & Pivot trajectories				
10 Experiments	Insertion planning time	Execution status	Pivot planning time	Execution status
Average	5.083449	0.9	5.090813	0.6
Standard deviation	0.066253	-	0.041338	-

Robot Planner 2: Simulation layout and reachability experiments

Goal: Find layout with better reachability

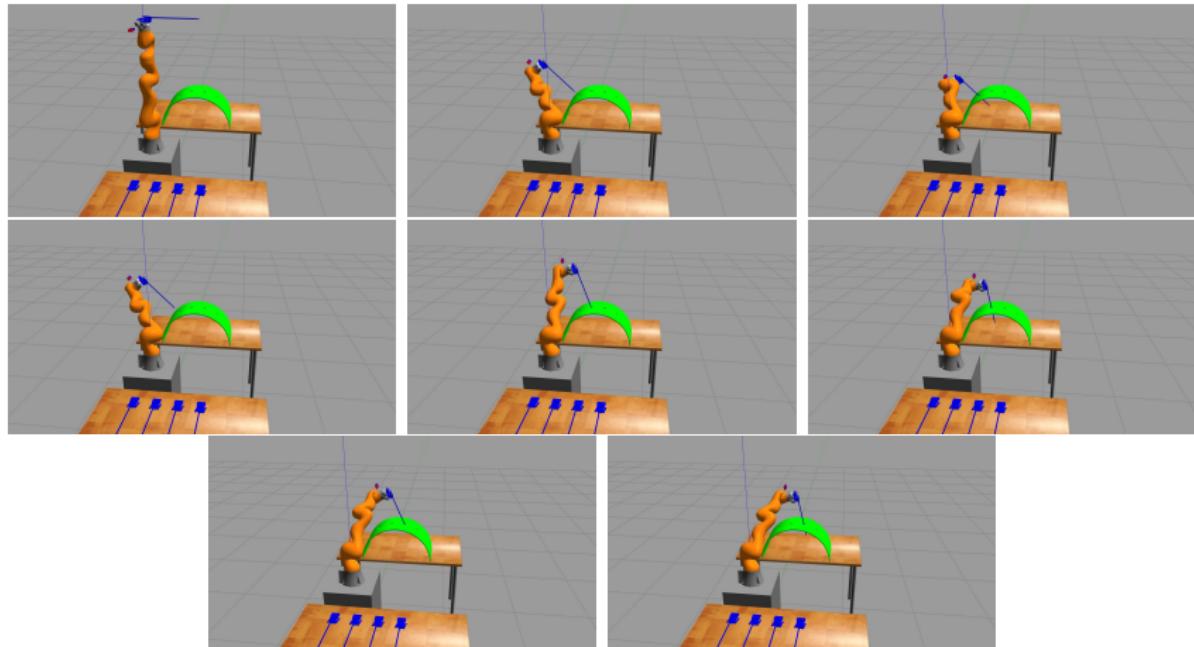


Figure: Experiment 2a: first layout

Robot Planner 2: Simulation layout and reachability experiments

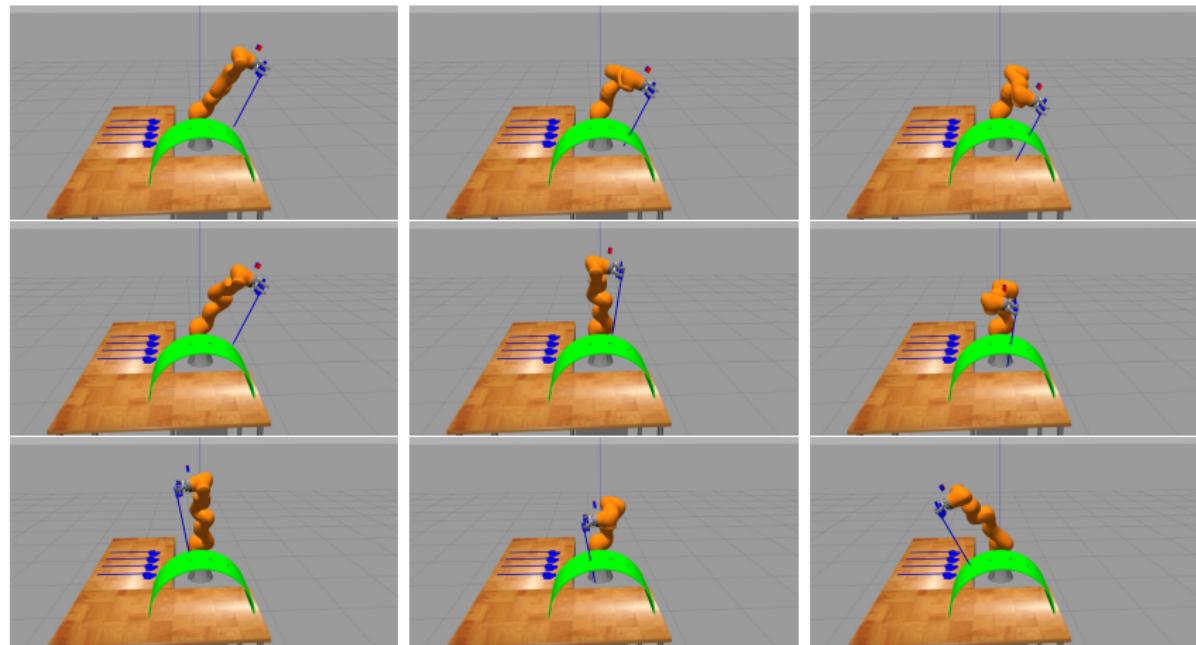


Figure: Experiment 2b: second layout

Robot Planner 3b: Line segment trajectories in task space

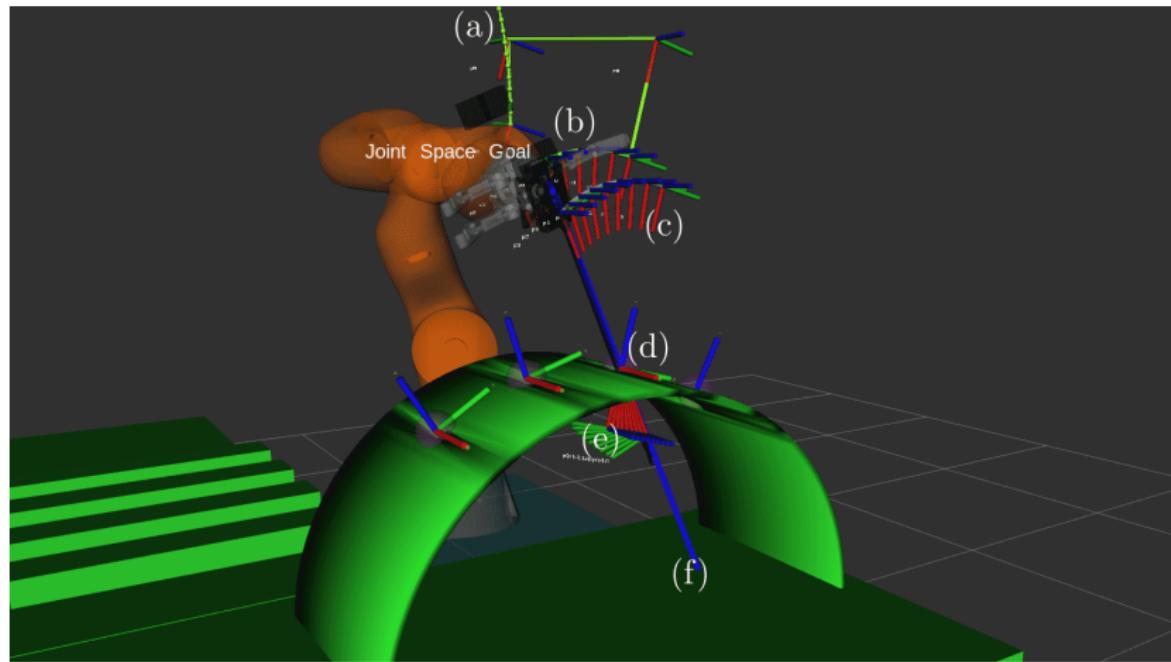


Figure: Experiment 3b

Robot Planner 3b: Line segment trajectories in task space

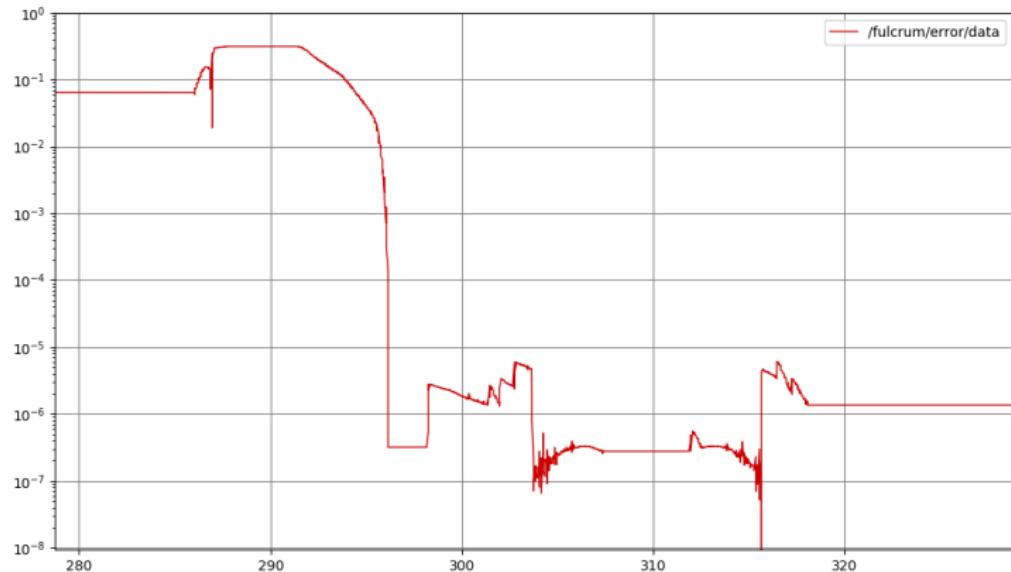


Figure: RCM error diagram from home position to line and reverse-line segment trajectories.

Robot Planner 3b: Line segment trajectories in task space

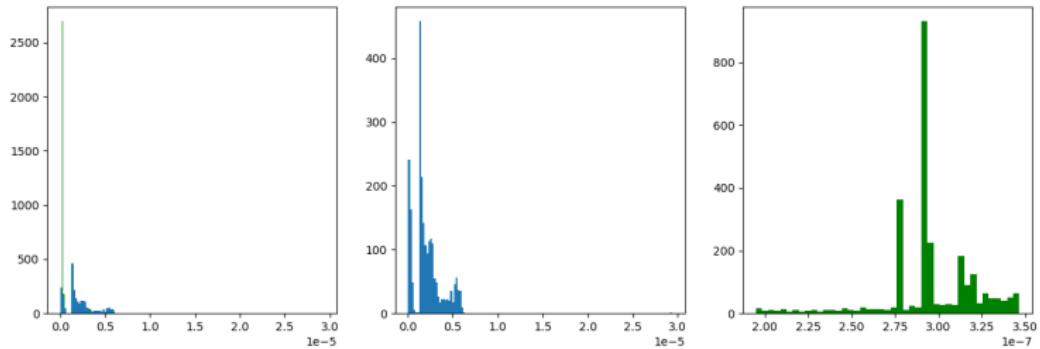


Figure: RCM error distributions, measurements from 10 iterations of the same experiment. From left to right: distribution of all measurements, distribution of measurements while the robot was pivoting, distribution of measurements while the robot was inserted but still.

	Average [m] (accuracy)	Standard Deviation [m] (repeatability)	sample size
while pivoting	$2.112649 \cdot 10^{-6}$	$1.609277 \cdot 10^{-6}$	2309
while inserted and still	$2.948652 \cdot 10^{-7}$	$2.948652 \cdot 10^{-7}$	2696

Line segment trajectories in task space

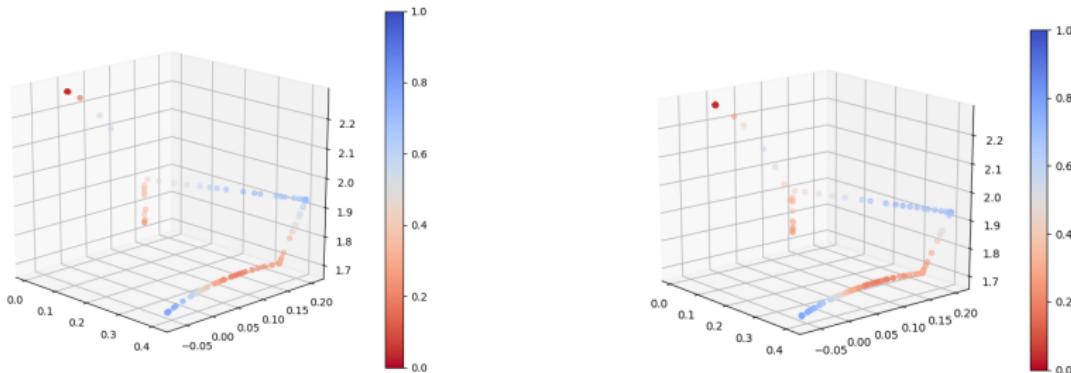


Figure: Experiment 3b: Manipulability plots of the whole trajectory the robot executed during 2 iterations of the same experiment

$$\mathcal{M} = \sqrt{\det(JJ^\top)}$$

Robot Planner 3b: Line segment trajectories in task space

Robot Planner 3b	Approach and line segment pivot trajectories with RRTConnect			
elbow-up preparatory path				
10 Experiments	Elbow-up Start pose planning time (sec)	Execution status	Elbow-up preparation path planning time (sec)	Execution status
Average	0.174222	1	0.117040	1
Standard deviation	0.049002	-	0.084238	-
Approach & Insertion				
10 Experiments	Approach fulcrum 2 path planning time (sec)	Execution status	Insertion path planning time (sec)	Execution status
Average	0.116498	1	0.249556	1
Standard deviation	0.088078	-	0.078941	-
Line segment pivot trajectories				
10 Experiments	Line segment path planning time (sec)	Execution status	Reverse line segment path planning time (sec)	Execution status
Average	1.809001	1	5.356607	0.7
Standard deviation	2.421448	-	0.086818	-

Robot Planner 3a: Circular and Circular arc trajectories in task space

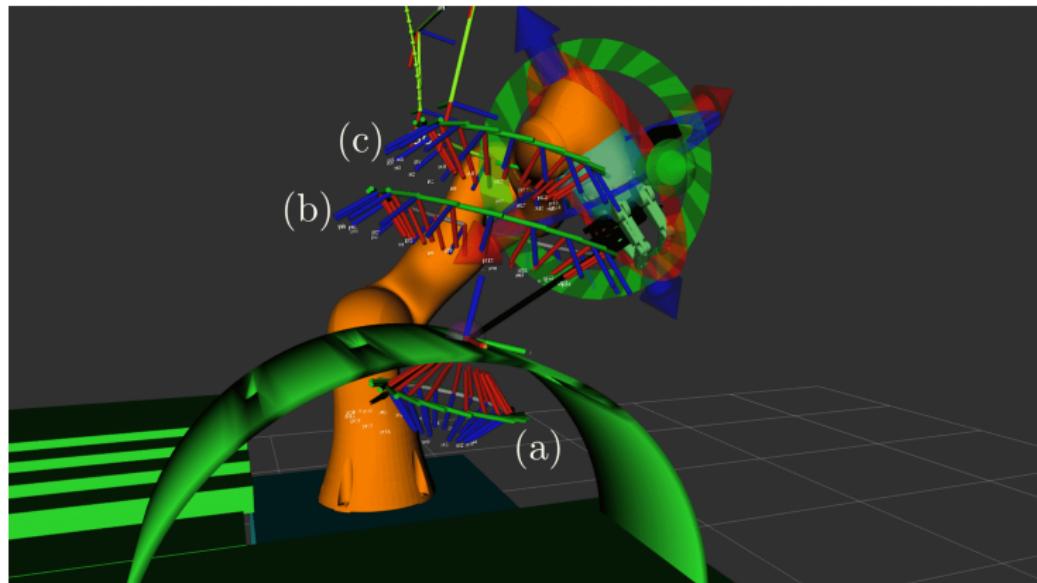


Figure: (a) The original circular trajectory designed inside the surgical taskspace, (b) the transformed trajectory that the base of the surgical tool will follow, (c) the actual transformed trajectory that the robot's end-effector will follow

Robot Planner 5: Visual servoing

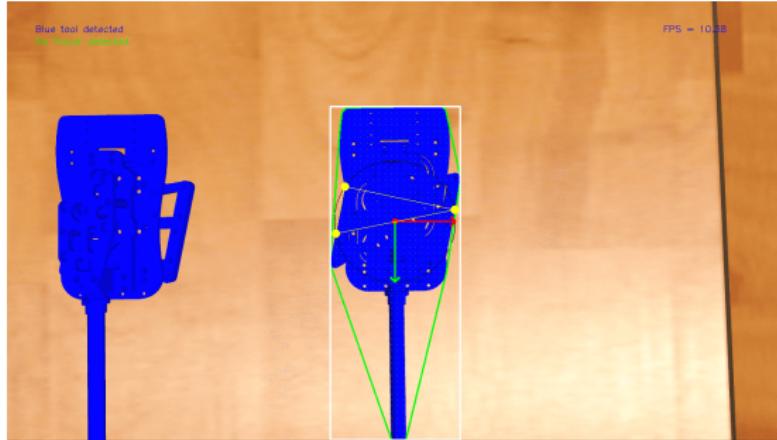


Figure: Image based visual servoing and calculation of grasp points. The yellow points are the grasp points and the thin black circumscribed circle is the growing circle that was used to calculate them.

Robot Planner 5: Visual servoing

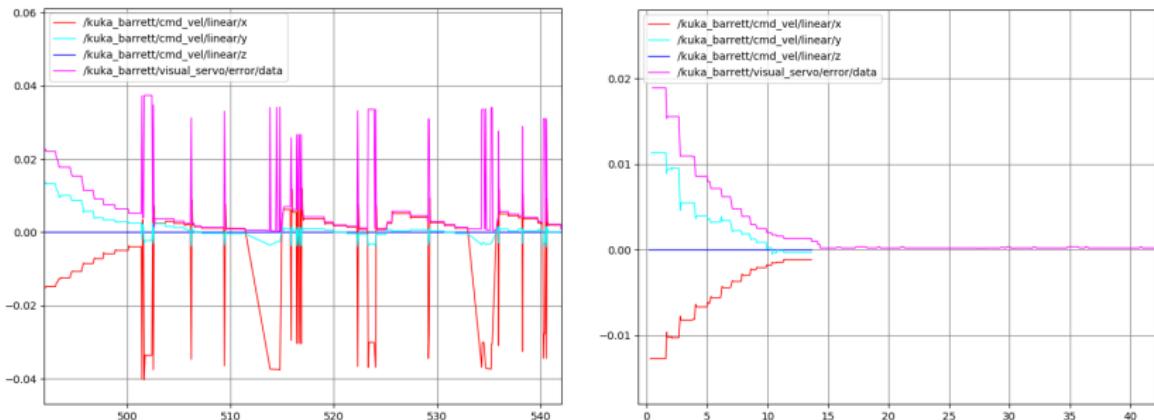
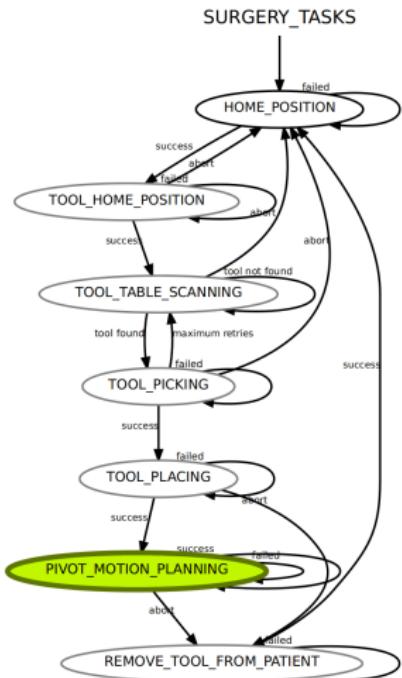


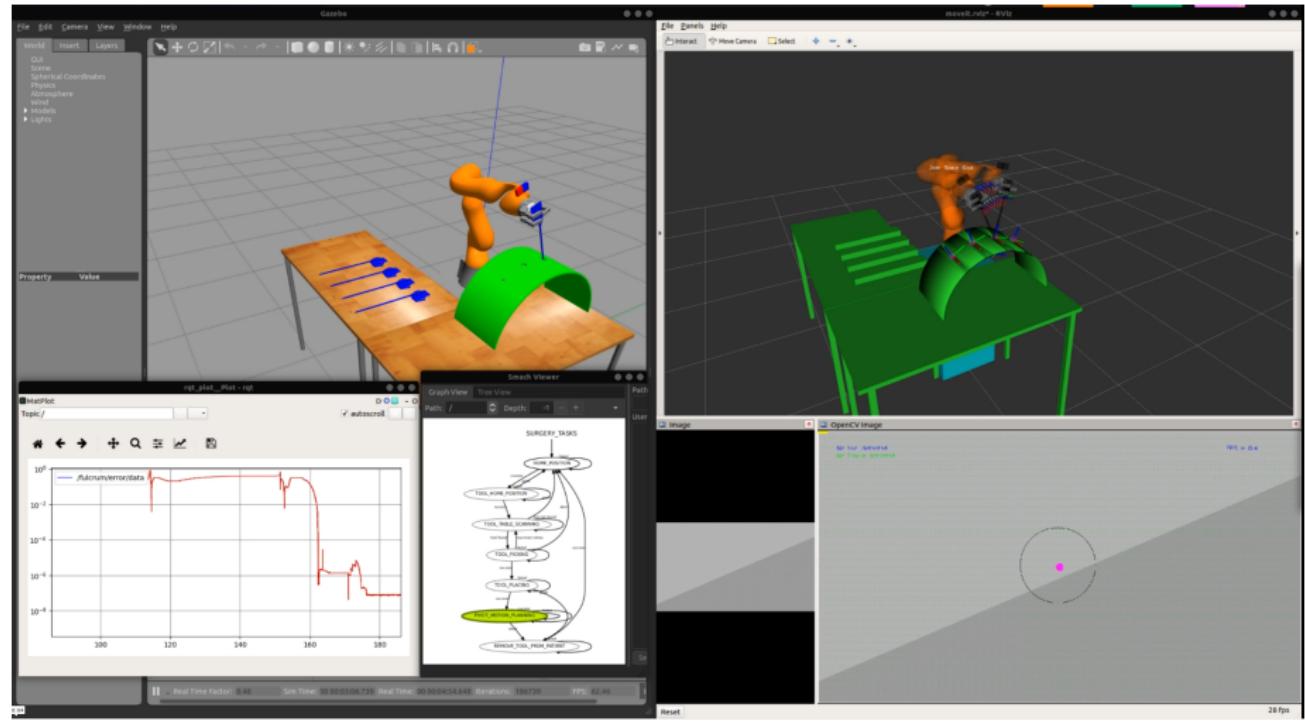
Figure: Visual servo controller error diagrams. On the left image in the error graphs appear some spikes. These spikes occur from the sudden temporary detection of a nearby surgical tool. On the right image, these spikes are filtered out, and only the error graphs of the visual servoing of one tool are shown. The controller parameters are $K_p = 0.9, K_d = 0.2$

Robot Planner 7: State machine - End-to-end simulation



Run all the stages of this thesis together (integration testing) using a state machine.

Demo



<https://youtu.be/lfV1vdHf7bk>

Outline

- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
 - Trajectory planning in cartesian coordinates
 - Trajectory planning in joints' space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

Conclusions

Surgical robotics critical requirements:

- Positional accuracy
- Orientation accuracy
- Time optimization
- RCM constraint
- Interpolation accuracy
- Collision avoidance
- Suitable path planning algorithms
- Repeatability
- Good environment layout and robot base position
- Good reachability of trajectories
- Singularity avoidance
- Jump threshold
- Fulcrum position and orientation
- Firm surgical tool grasping

Future Work

- Simulation and interaction with deformable bodies
- Advanced visualization and Haptics feedback
- Using the trajectories from chapter 6 as building blocks for more complicated trajectories like suturing
- Multiple robot arm collaboration: designing pivot trajectories for 2 or more robot arms
- Advanced grasping of gripper and/or laparoscopic tool's tip

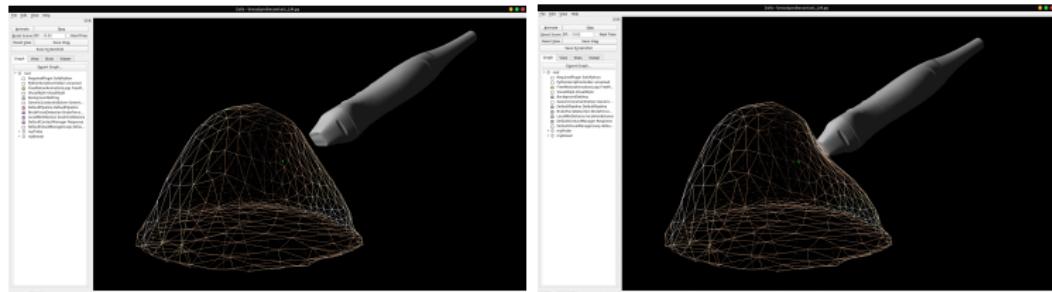
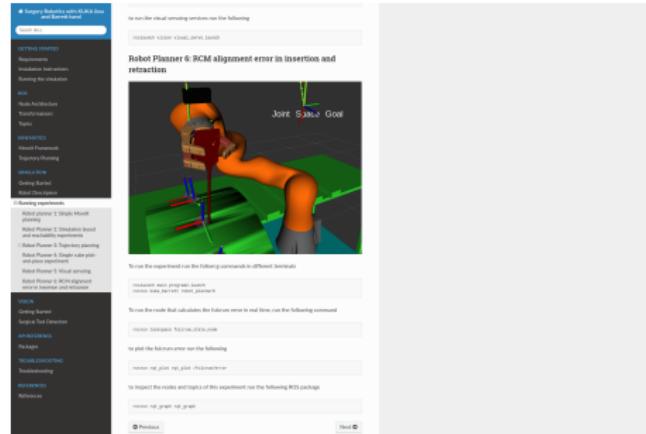


Figure: Soft tissue interaction, deformable objects simulation with the SOFA Framework

Code & Documentation



- Git repository:
https://github.com/karadalex/surgery_robotics_kuka_barrett
- Documentation:
https://karadalex.github.io/surgery_robotics_kuka_barrett/

Questions

Questions?

Thank you,

up1046820@upnet.gr
karadalex@gmail.com