

# 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 joint angles 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

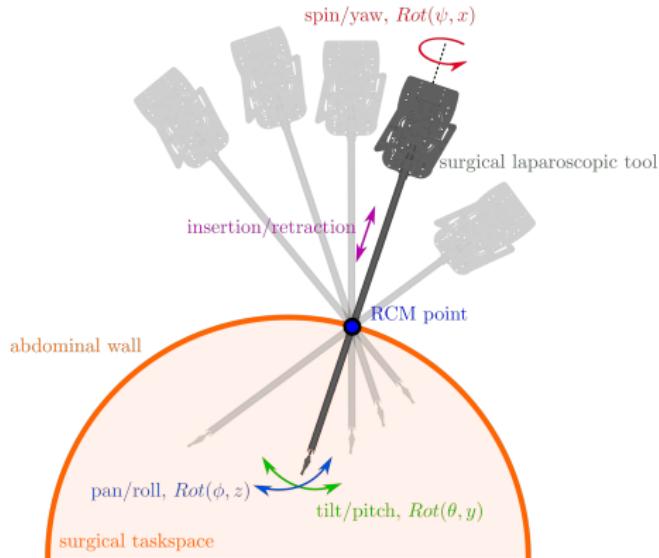
- 1 Introduction
- 2 Robotic arm Kinematic Analysis
- 3 Grasping
- 4 Scene and object recognition with Computer Vision
- 5 Path Planning
- 6 Trajectory Planning - Laparoscopic tool manipulation
  - Trajectory planning in cartesian coordinates
  - Trajectory planning in joint angles space
- 7 System Control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

# Forward Kinematics

# Inverse Kinematics - Decoupling Technique

# 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

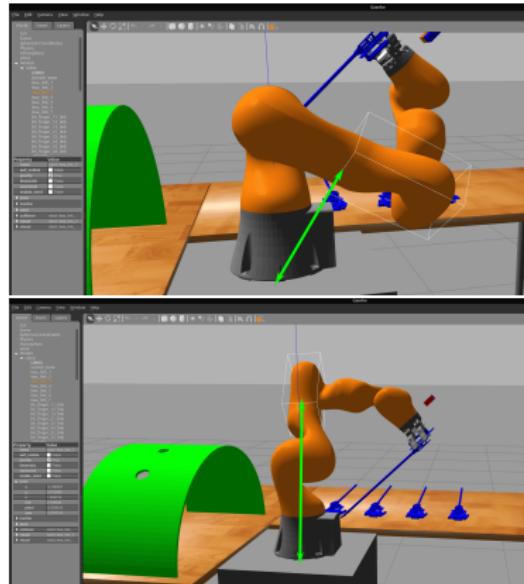


Figure: Top: elbow-down solution, bottom: elbow-up solution

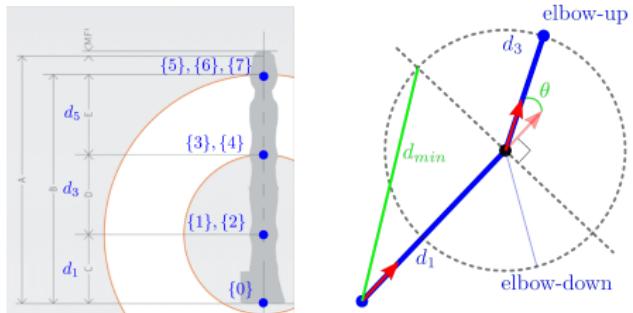


Figure: Elbow-up constraint description with relative distance or angle between links with lengths  $d_1$  and  $d_3$

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

# 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 joint angles space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

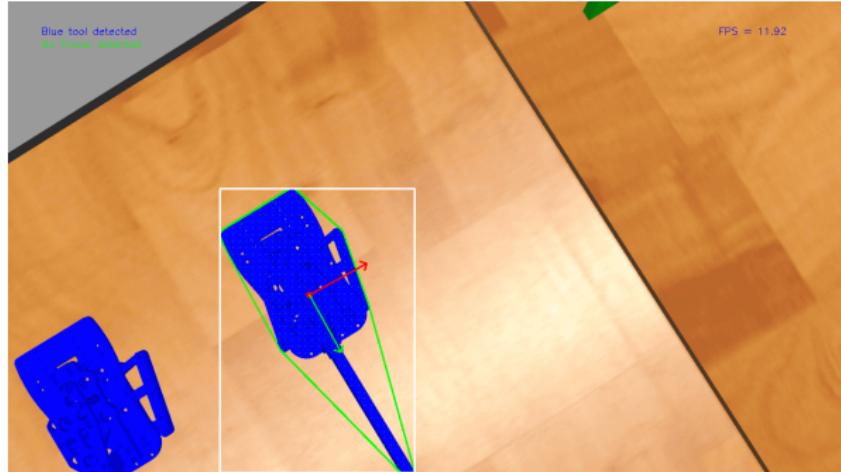
# Gripper & Forward Kinematics

# Gripper Inverse Kinematics

# 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 joint angles 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

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

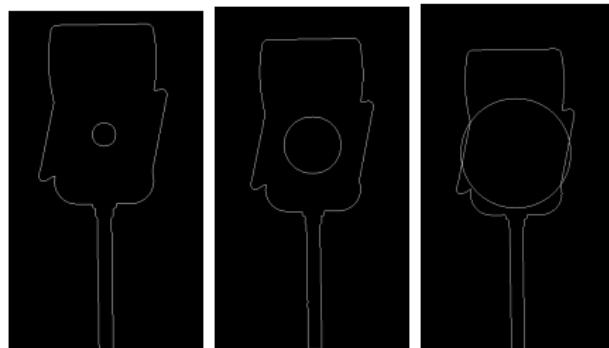
$$\mathbf{C} = \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix}$$

**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  $I_2(x, y)$

- ①  $\mathbb{G} = \arg \max_{(x,y)} I_1(x, y) \odot I_2(x, y)$
- ② match points with stereo 3d points
- ③ check feasibility with gripper kinematics

# 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 joint angles space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

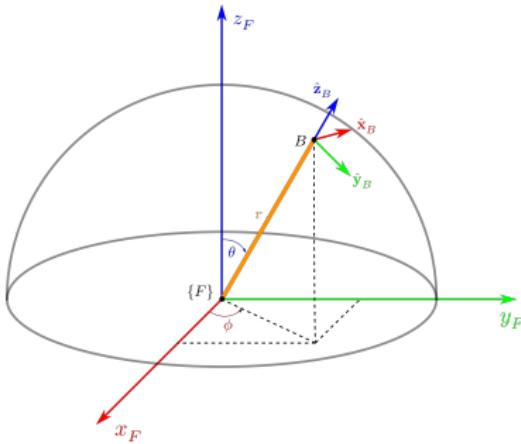
# Path Planning - Sampling methods

# 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 joint angles space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

# Tool pose



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

Calculate orientation vectors using spherical coordinate unit vectors

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

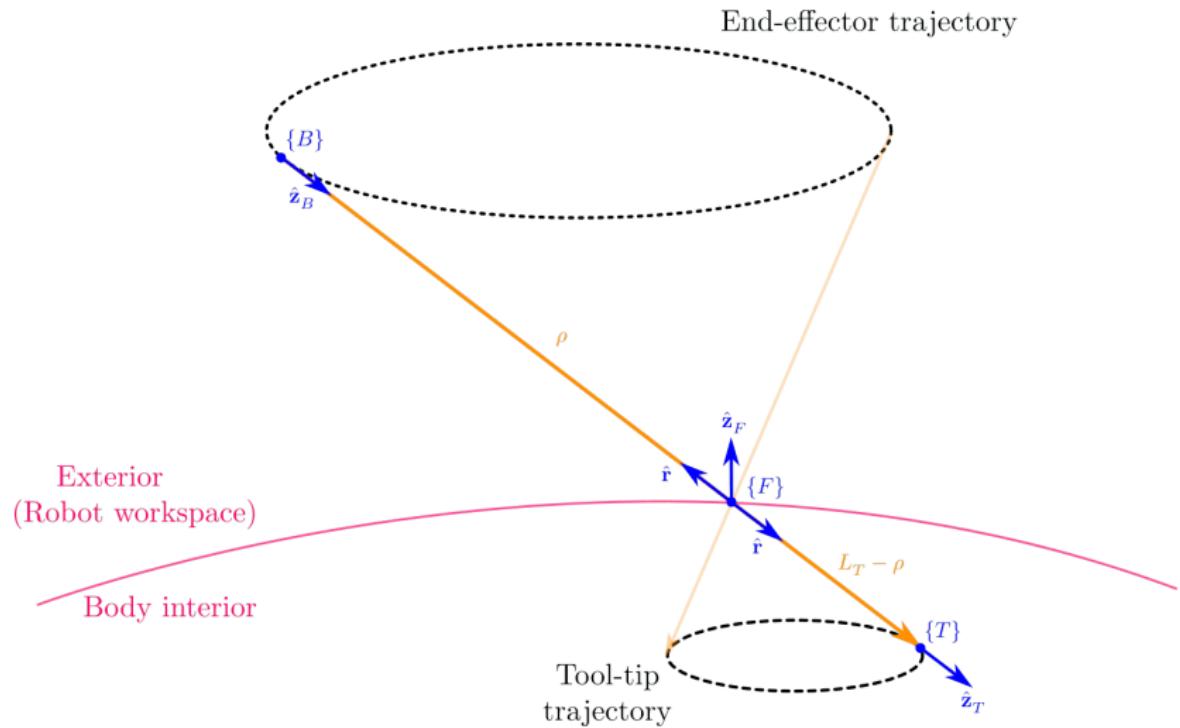
$$\hat{\mathbf{y}}_B = \hat{\theta} = \begin{bmatrix} \cos(\theta) \cos(\varphi) \\ \cos(\theta) \sin(\varphi) \\ -\sin(\theta) \end{bmatrix}$$

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

where

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

## The Fulcrum Effect



# Line segment trajectory of tool tip

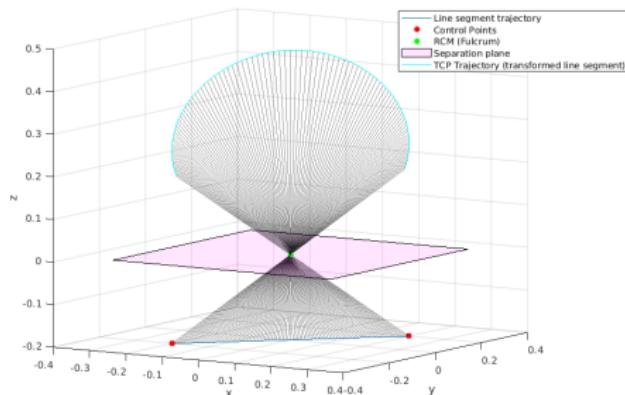
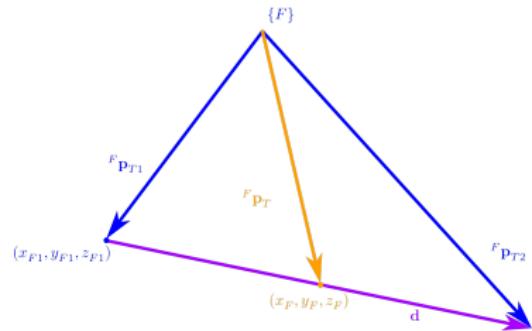
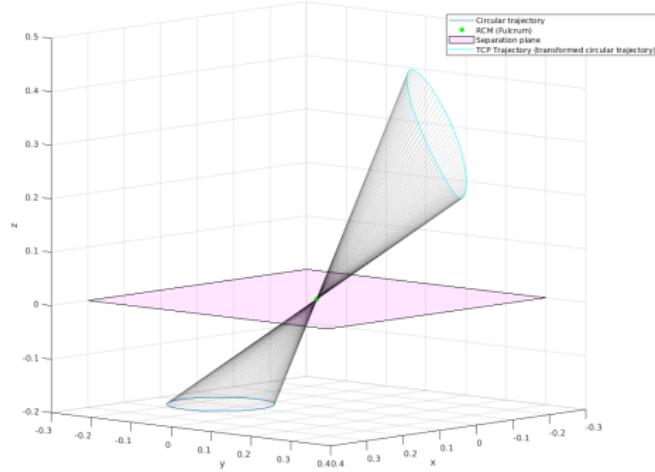


Figure: A Line segment trajectory and it's transformation due to the Fulcrum Effect



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

# Circular trajectory of tool tip



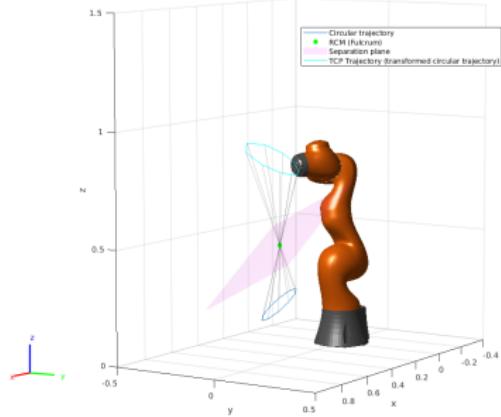
**Figure:** Circular trajectory of tool tip with respect to Fulcrum reference frame and it's transformation via the Fulcrum Effect

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

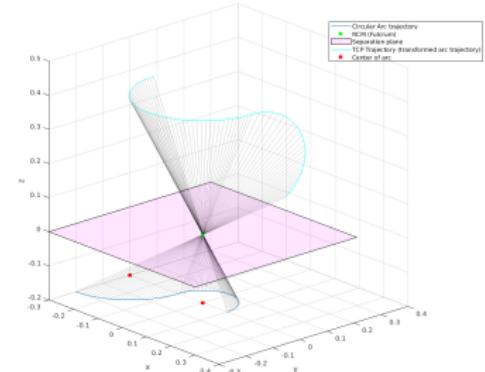
$$s \in [0, 1]$$

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

# 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

# Cubic Spline trajectory of tool tip

# B-Spline trajectory of tool tip

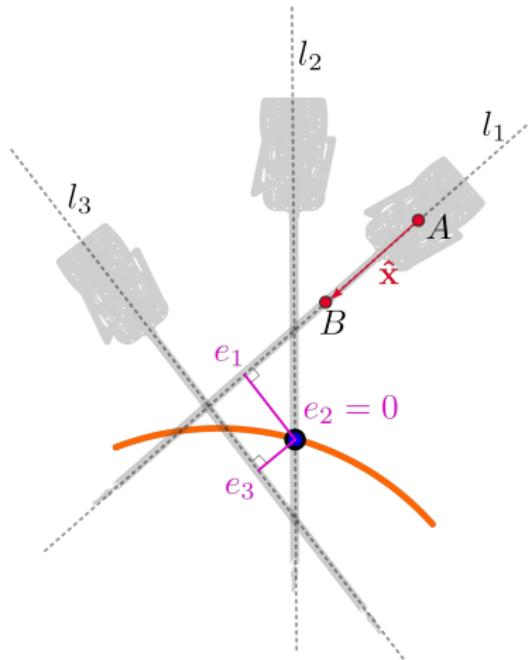
# Polynomials of 5th order

# Planning with velocity profiles

# 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 joint angles 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

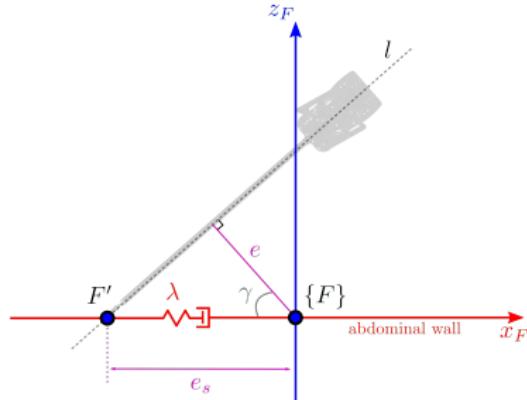


Figure: Force interaction model of the laparoscopic tool and the abdominal wall around the fulcrum point (RCM point)

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

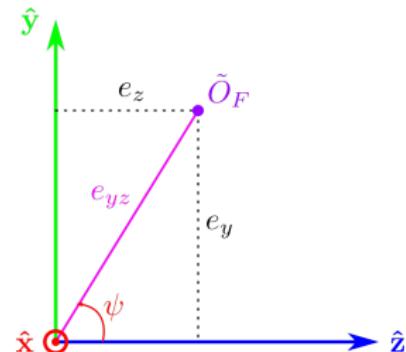
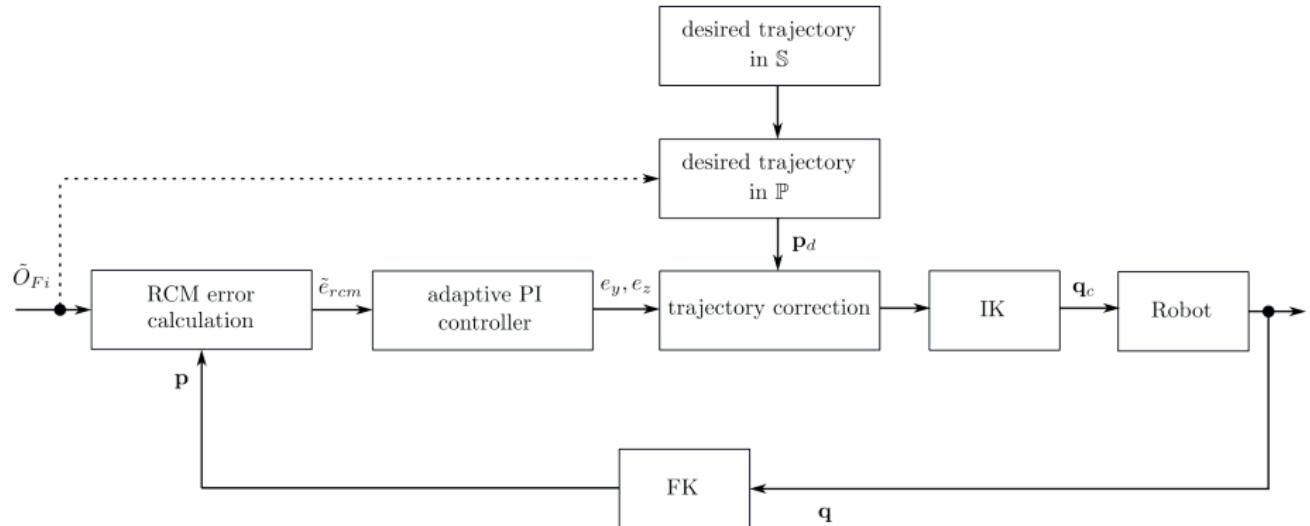


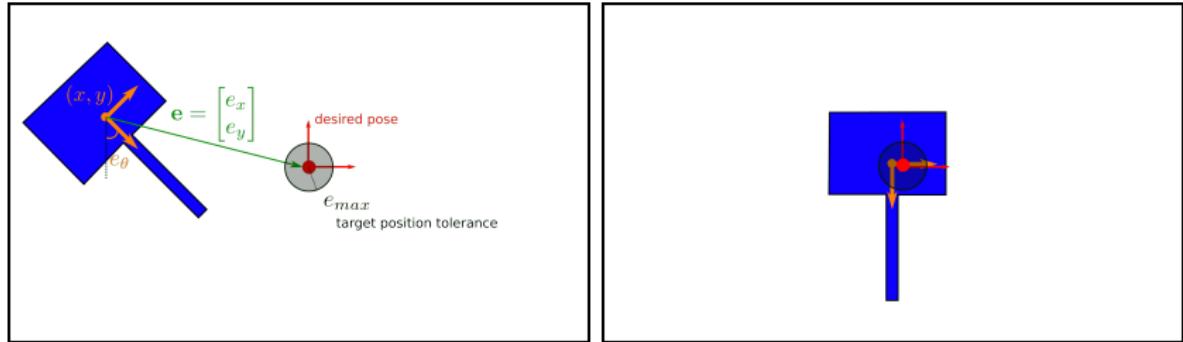
Figure: RCM error calculation in  $yz$  plane. The RCM error or  $yz$ -error is the distance between the line of the  $\hat{x}$  vector (here seen as a point) and the estimated position of the origin of the fulcrum reference frame  $\tilde{O}_F$

# 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  $x, y, \theta$  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

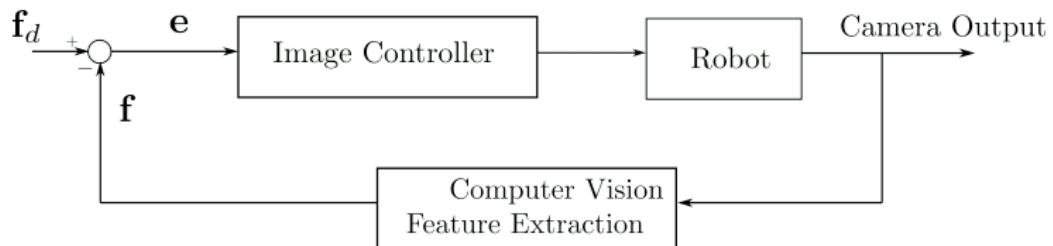


Figure: Image based visual servoing closed loop control

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

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

# Firm grasping algorithm & Force control

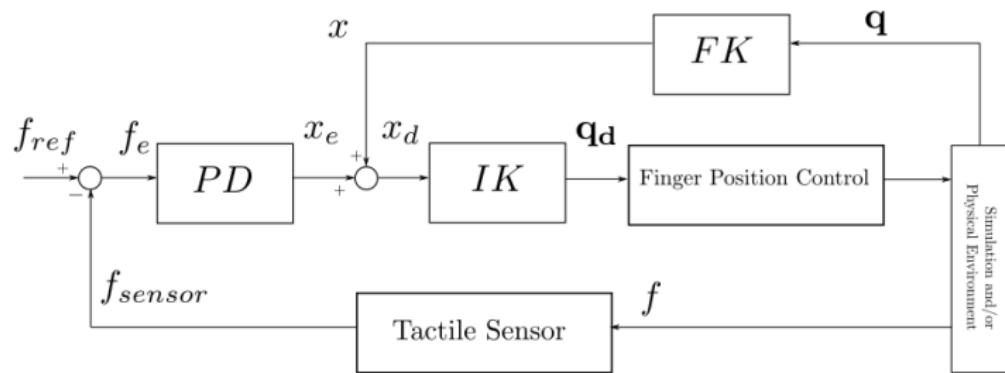


Figure: Force control on a Barrett Hand gripper finger

# 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 joint angles 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
- ...

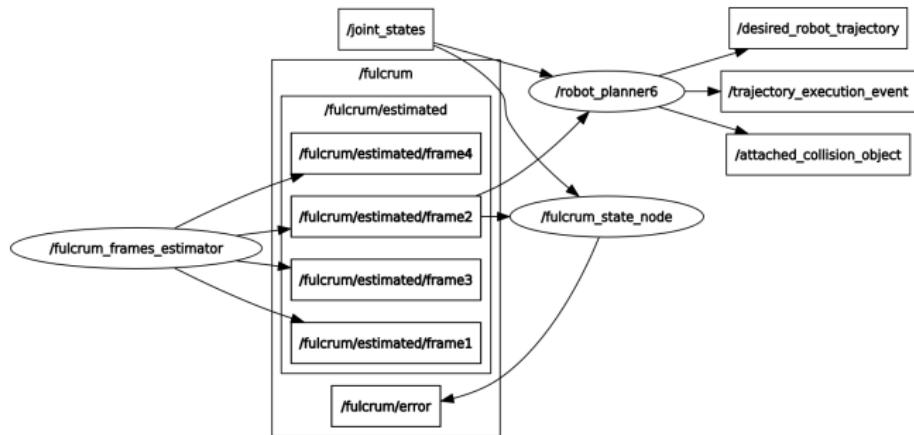


Figure: Subset of ROS nodes and topics used for the robot-planner6 experiment

# Gazebo simulation environment

# Visualization with RViz

# Motion Planning with Moveit

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

- **Position tolerance:** 50 - 500 $\mu\text{m}$ , 5 $\mu\text{m}$
- **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

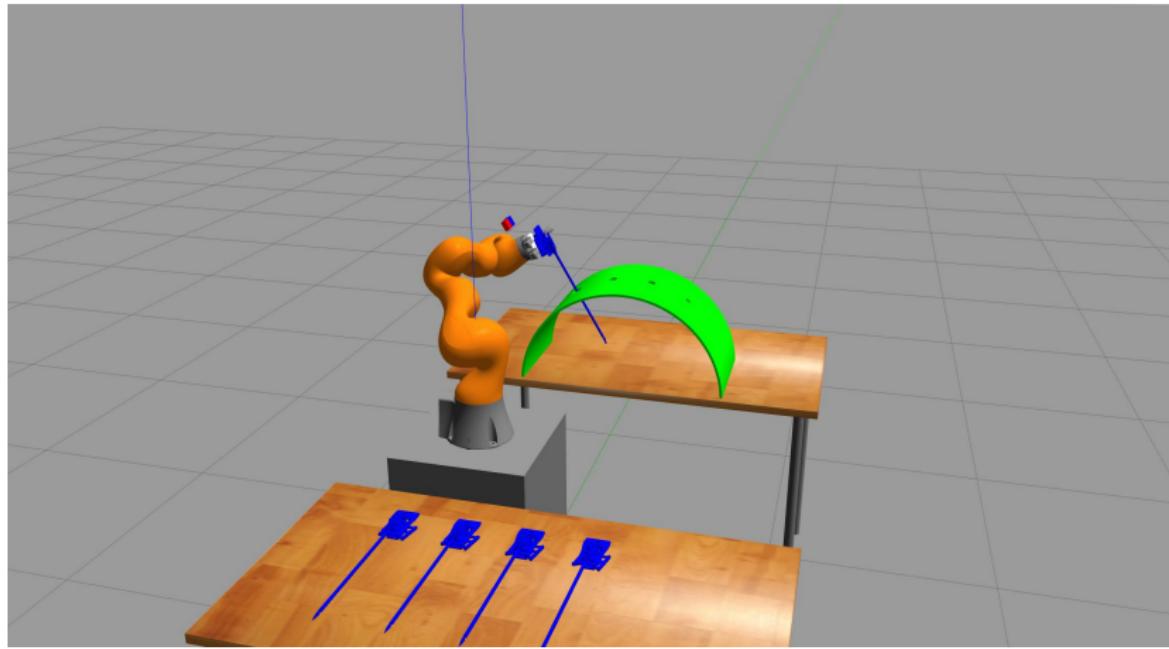
- **tf2**: keep track of multiple coordinate frames, apply transformations
- **geometry\_msgs**
- **Eigen**: linear algebra
- **OpenCV2**: computer vision
- **numpy**
- **actionlib**
- State machines with **Smach**
- ros-industrial/kuka\_experimental
- **barrett\_hand**
- **gazebo-pkgs**
- **moveit-pkgs**
- ...

# 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 joint angles 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
<b>Average</b>	0.163102	1	0.5014398	0.9
<b>Standard deviation</b>	0.110001	-	1.110582	-

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

# Robot Planner 2: Simulation layout and reachability experiments

**Goal:** Find layout with better reachability

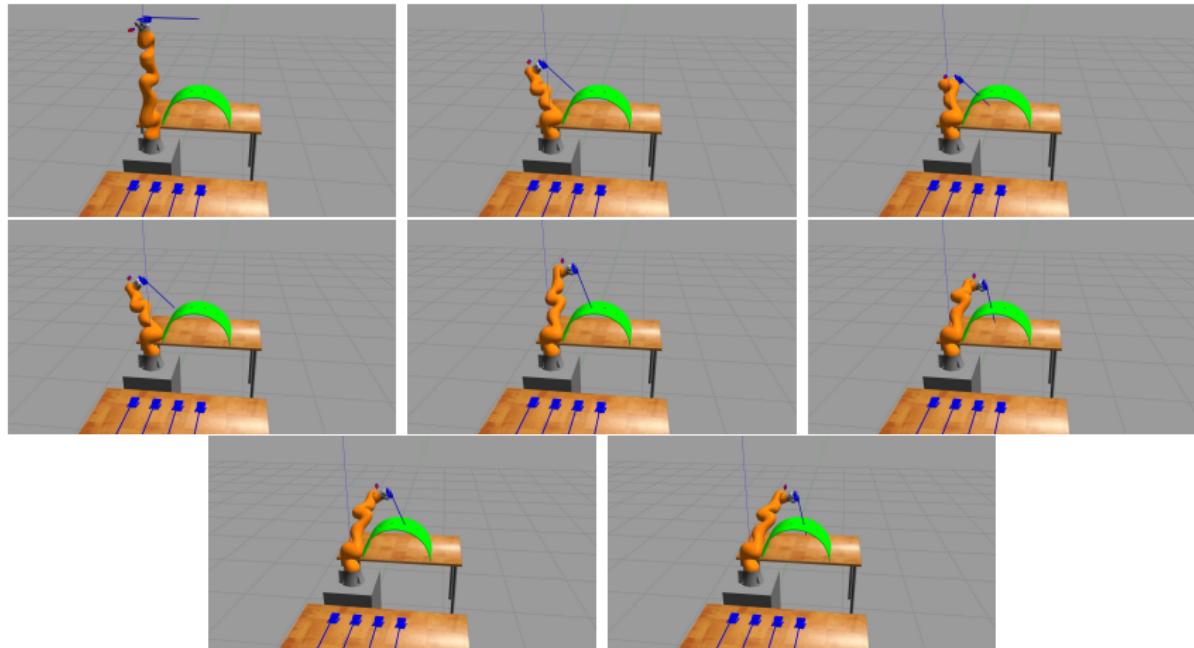


Figure: Experiment 2a: first layout

# Robot Planner 2: Simulation layout and reachability experiments

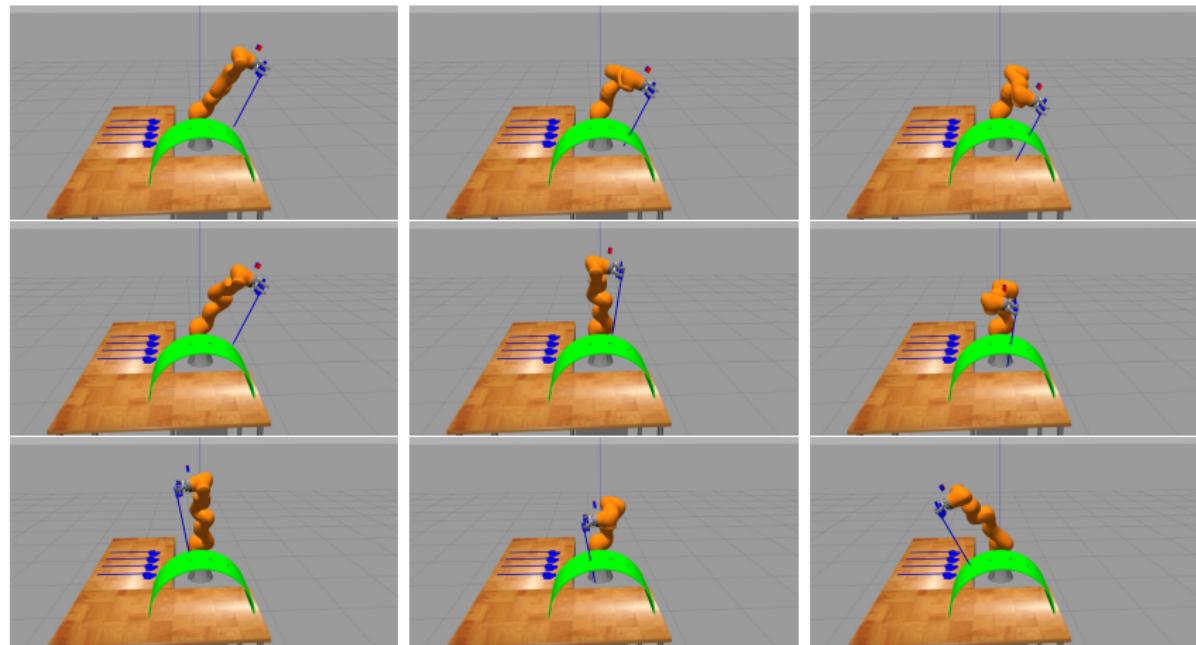


Figure: Experiment 2b: second layout

# Robot Planner 3b: Line segment trajectories in task space

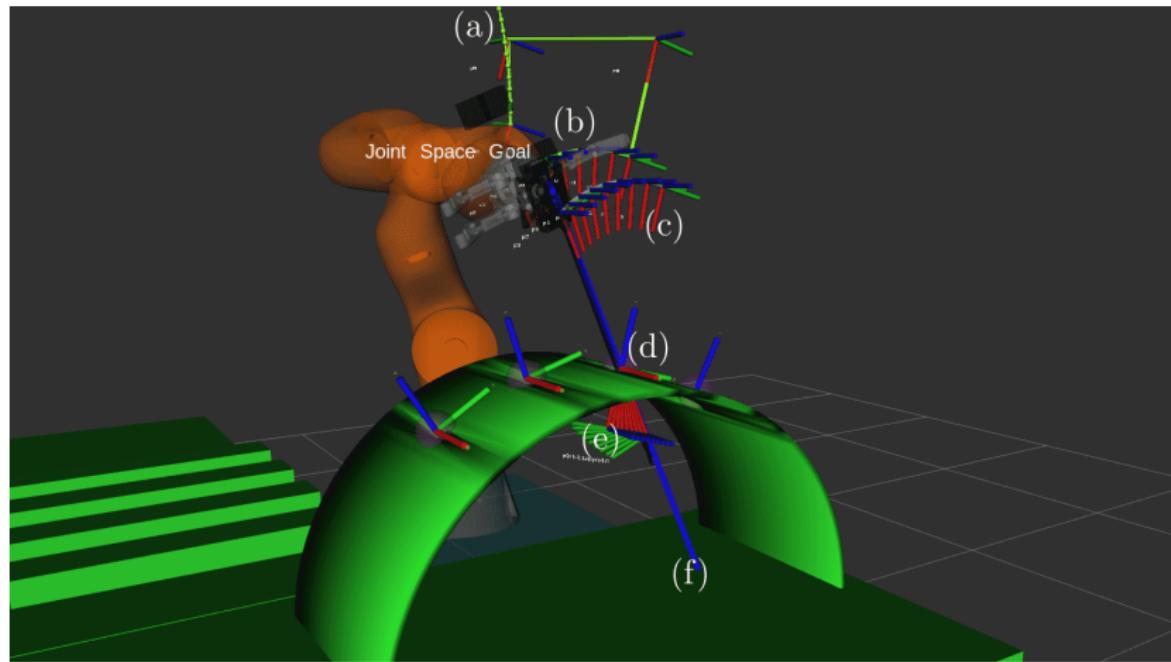
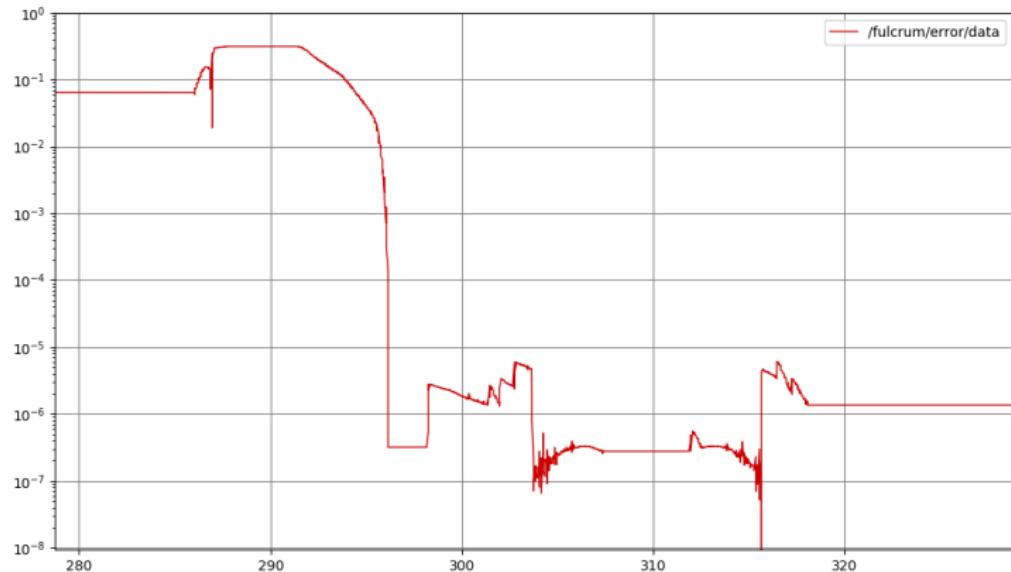


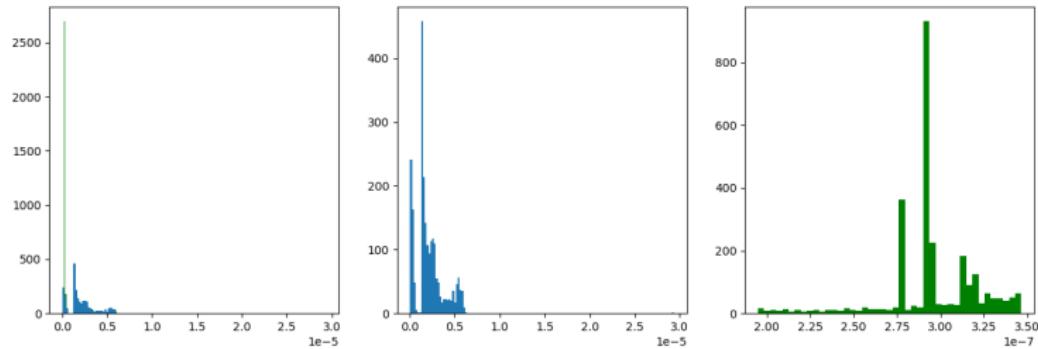
Figure: Experiment 3b

# Robot Planner 3b: Line segment trajectories in task space



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

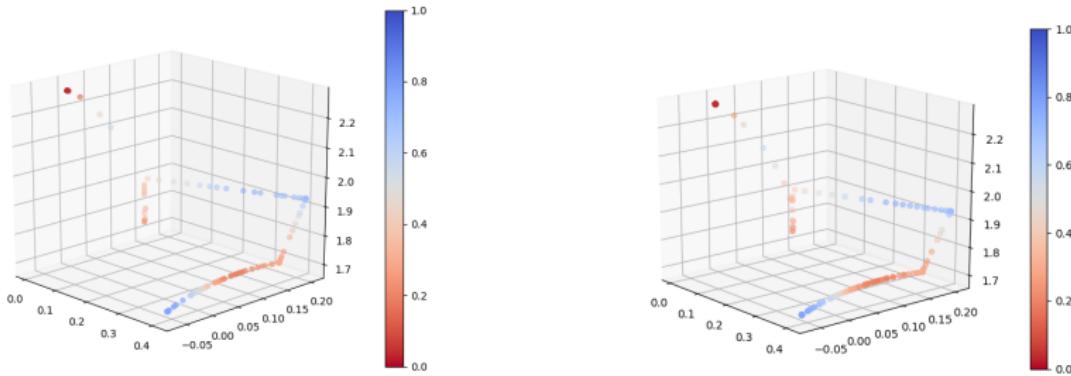
# Robot Planner 3b: Line segment trajectories in task space



**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
<b>while pivoting</b>	$2.112649 \cdot 10^{-6}$	$1.609277 \cdot 10^{-6}$	2309
<b>while inserted and still</b>	$2.948652 \cdot 10^{-7}$	$2.948652 \cdot 10^{-7}$	2696

# Line segment trajectories in task space

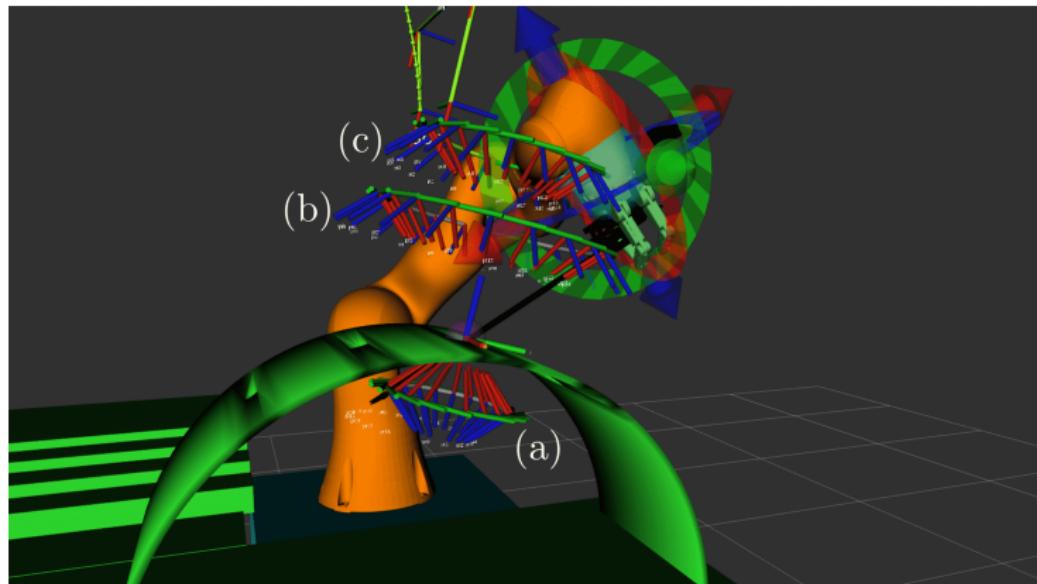


**Figure:** Experiment 3b: Manipulability plots of the whole trajectory the robot executed during 2 iterations of the same experiment

# Robot Planner 3b: Line segment trajectories in task space

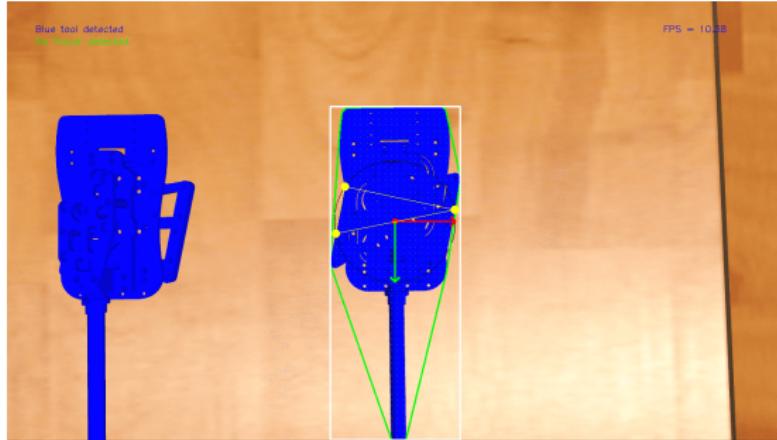
Robot Planner 3b	Approach and line segment pivot trajectories with RRTConnect			
<b>elbow-up preparatory path</b>				
10 Experiments	Elbow-up Start pose planning time (sec)	Execution status	Elbow-up preparation path planning time (sec)	Execution status
<b>Average</b>	0.174222	1	0.117040	1
<b>Standard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

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



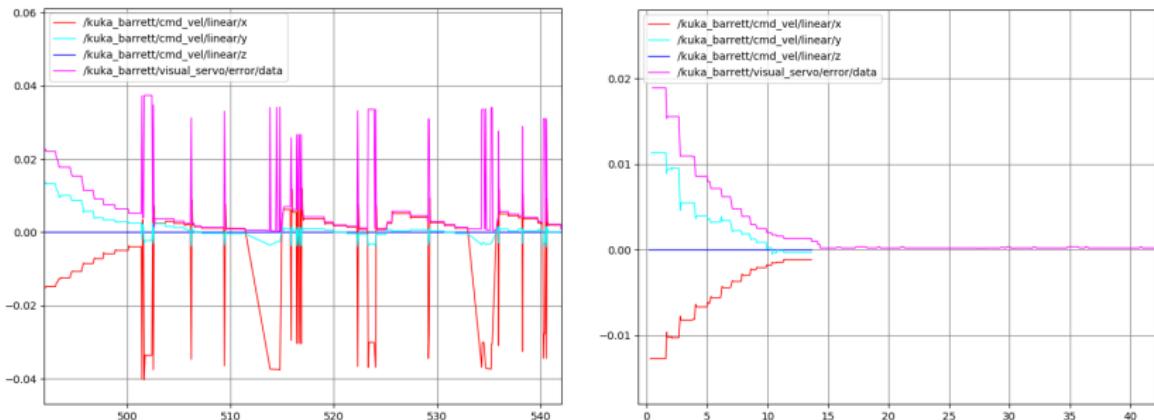
**Figure:** (a) The original circular trajectory designed inside the surgical taskspace, (b) the transformed trajectory that the base of the surgical tool will follow, (c) the actual transformed trajectory that the robot's end-effector will follow

# Robot Planner 5: Visual servoing



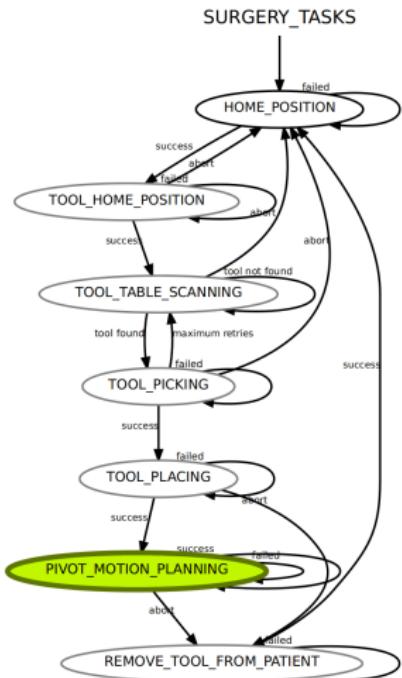
**Figure:** Image based visual servoing and calculation of grasp points. The yellow points are the grasp points and the thin black circumscribed circle is the growing circle that was used to calculate them.

# Robot Planner 5: Visual servoing



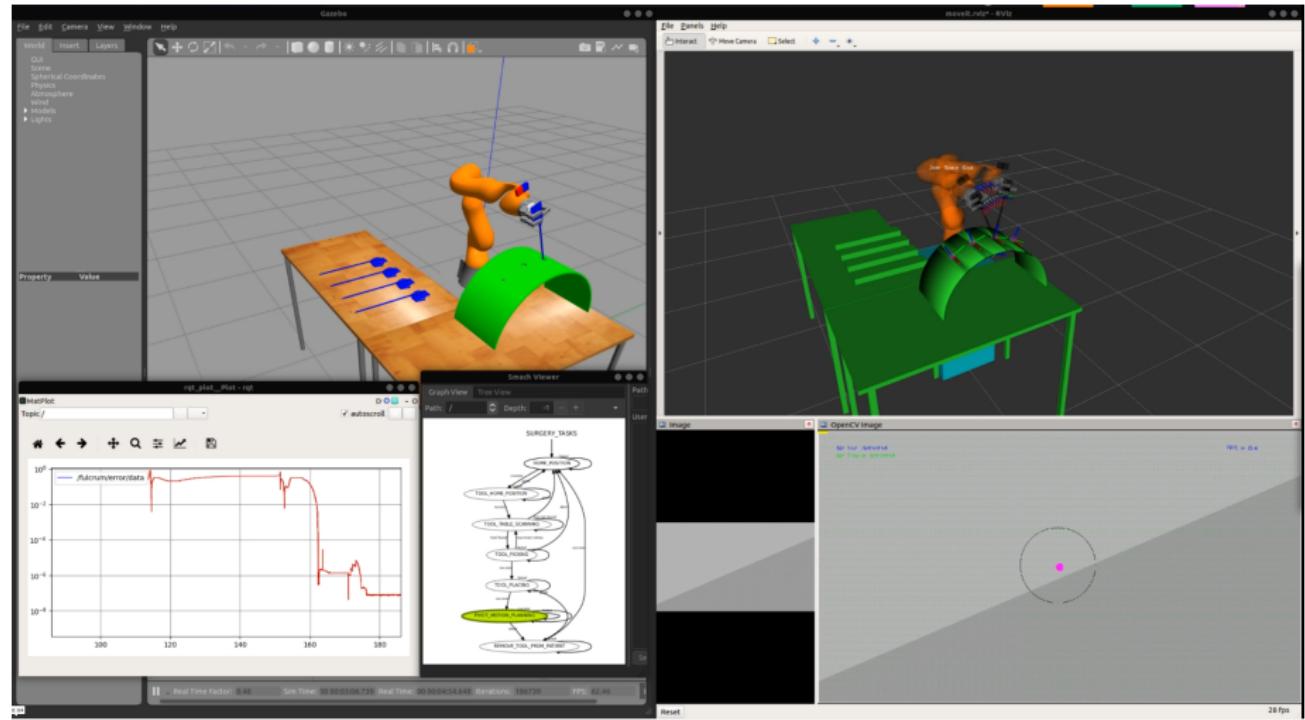
**Figure:** Visual servo controller error diagrams. On the left image in the error graphs appear some spikes. These spikes occur from the sudden temporary detection of a nearby surgical tool. On the right image, these spikes are filtered out, and only the error graphs of the visual servoing of one tool 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>

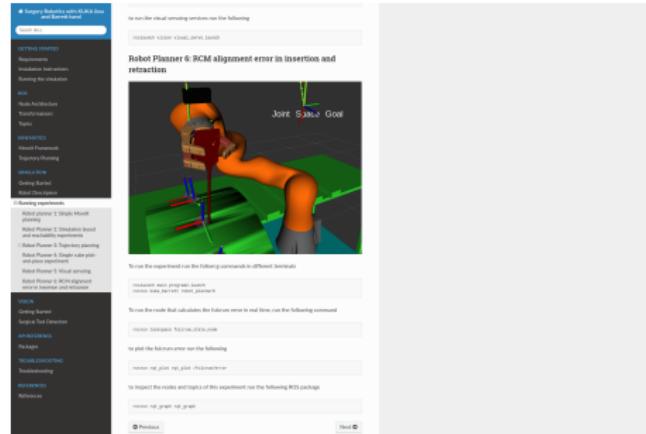
# Outline

- 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 joint angles space
- 7 System Control
- 8 ROS framework
- 9 Simulation Studies
- 10 Conclusions and Future Work

# Conclusions

# Future Work

# Code & Documentation



- Git repository:  
[https://github.com/karadalex/surgery\\_robotics\\_kuka\\_barrett](https://github.com/karadalex/surgery_robotics_kuka_barrett)
- Documentation:  
[https://karadalex.github.io/surgery\\_robotics\\_kuka\\_barrett/](https://karadalex.github.io/surgery_robotics_kuka_barrett/)

# Questions

**Questions?**

Thank you,

up1046820@upnet.gr  
karadalex@gmail.com