UNIVERSITY OF PATRAS - SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Division: Systems and Automatic Control

Thesis

of the student of the Department of Electrical and Computer Engineering of the School of Engineering of the University of Patras

Karadimos Alexios of Loukas

Student Number: 1046820

Subject

Robotic surgical tool manipulator - Recognition, control and manipulation of laparoscopic tools

Supervisor

Associate Professor Dr. Evangelos Dermatas

Co-Supervisor

Professor Dr. Anthony Tzes

Thesis Number:

Patras, 2020

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα

Robotic surgical tool manipulator - Recognition, control and manipulation of laparoscopic tools

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Karadimos Alexios of Loukas

(A.M.: 1046820)

παρουσιάστηκε δημόσια και εξετάστηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

__/__/___

Ο Επιβλέπων | Ο Διευθυντής του Τομέα

Evangelos Dermatas
Associate Professor Dr.

Kazakos Demosthenes
Assistant Professor Dr.

Contents

1 Introduction **5**

2 Robotic arm Kinematic Analysis **5**
- 2.1 Robotic arm, DH parameters & Forward Kinematics . . . 5
- 2.2 Inverse Kinematics . . . 6
  - 2.2.1 Decoupling Technique . . . 6
  - 2.2.2 Workspace constraints & Singularity points . . . 7
  - 2.2.3 Solutions for 7DoF numerically . . . 7
  - 2.2.4 Comparison of Inverse Kinematics Techniques . . . 8

3 Grasping **8**
- 3.1 Gripper & Forward Kinematics . . . 8
- 3.2 Gripper Inverse Kinematics . . . 8
- 3.3 Force closure . . . 9
- 3.4 Firm grasping algorithm & Force control . . . 9

4 Scene and object recognition with Computer Vision **9**
- 4.1 Laparoscopic tool detection . . . 9
- 4.2 Calculation of tool position and orientation . . . 10
- 4.3 Calculation of grasping points . . . 11
- 4.4 Trocar detection & Estimation of fulcrum point . . . 11

5 Laparoscopic tool manipulation **11**
- 5.1 Tool pose . . . 11
- 5.2 Pivoting motion with respect to Fulcrum Point . . . 12
  - 5.2.1 Circular trajectory of tool tip . . . 12
  - 5.2.2 Circular arc trajectory of tool tip . . . 13
  - 5.2.3 Line segment trajectory of tool tip . . . 13
- 5.3 Task space analysis . . . 14

6 Path Planning **14**
- 6.1 Path searching . . . 14
- 6.2 Pick and place algorithm . . . 14

7 Trajectory Planning **14**
- 7.1 Trajectory planning in cartesian coordinates . . . 14
- 7.2 Trajectory planning in joint angles space . . . 15

8 Implementation with the ROS framework **16**
- 8.1 Gazebo simulation environment . . . 16
- 8.2 Visualization and Motion Planning with RViz and Moveit . . . 16
- 8.3 Experiments and Development methodology . . . 17
  - 8.3.1 Robot Planner 1 . . . 17
  - 8.3.2 Robot Planner 2 . . . 17
  - 8.3.3 Robot Planner 3 . . . 17
  - 8.3.4 Robot Planner 4 . . . 17
  - 8.3.5 Robot Planner 5 . . . 17

Nomenclature **18**

List of Figures **19**

List of programs **19**

Bibliography **19**

1 Introduction

2 Robotic arm Kinematic Analysis

2.1 Robotic arm, DH parameters & Forward Kinematics

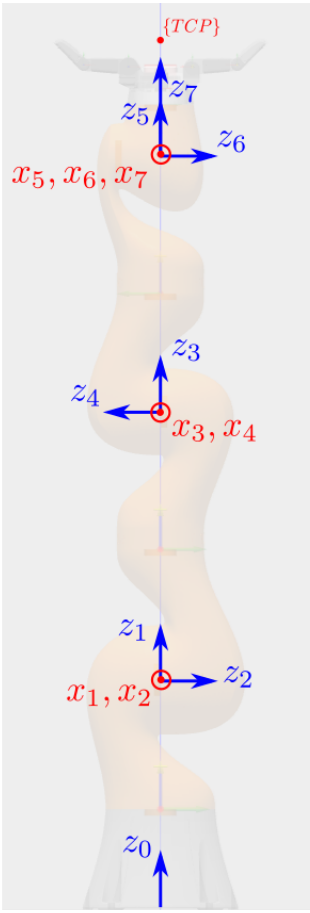

Figure 1: Joint reference frames of the KUKA iiwa14 robot

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	$\pi/2$

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

2.2 Inverse Kinematics

2.2.1 Decoupling Technique

In this section the inverse kinematics problem is solved for only the 6 out of the 7 total degrees of freedom. The third joint is not used in this analysis and it's angle is set to zero $\theta_3 = 0$. The rest of the joints form a special kind of kinematic chain that can be solved using the decoupling technique. In this technique the Inverse kinematics problem is split to 2 separate subproblems, one for the position and one for the orientation of the end-effector. This technique can be applied in this case because the axes of the 3 last joints intersect at the same point and they form an Euler wrist.

To solve for the joints' angles, the transformation matrix ${}^0T_7$ of the end-effector with respect to the robot's base is required. Usually the transformation ${}^UT_{tcp}$ is known, which is the pose of Tool's center point (TCP) with respect to the Universal Coordinate Frame $\{U\}$ from which the required ${}^0T_7$ can be calculated

$$
\begin{gathered}
{}^UT_{tcp} = {}^UT_0 \ {}^0T_7 \ {}^7T_{tcp} \\
{}^0T_7 = {}^UT_0^{-1} \ {}^UT_{tcp} \ {}^7T_{tcp}^{-1} \\
{}^0T_7 = \begin{bmatrix} R_t & \mathbf{p}_t \\ 0 & 1 \end{bmatrix}
\end{gathered}
$$

where ${}^UT_0$, ${}^7T_{tcp}$ are translation transformations by a constant distance and R_t, $\mathbf{p}_t$ are the target's orientation and position respectively.

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

$$
\theta_1 = \begin{cases} atan2\left(p_y, p_x\right) \\ \pi - atan2\left(p_y, p_x\right) \end{cases} \tag{2.2.1}
$$

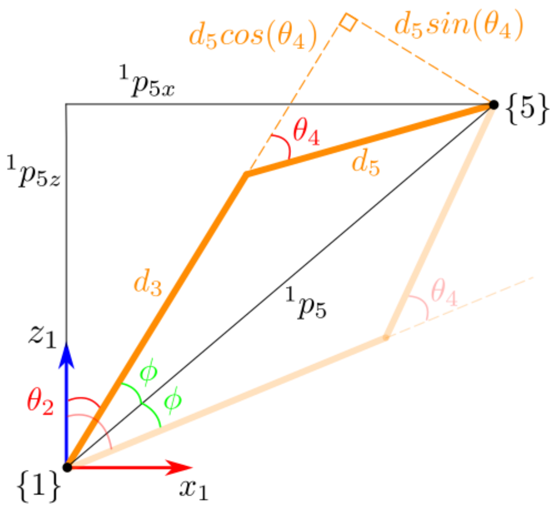

Figure 2: Calculation of angles θ_2, θ_4

$$
\varphi = acos\left(\frac{d_3^2 + \|{}^1p_5\|^2 - d_5^2}{2d_3\|{}^1p_5\|}\right)
$$

$$\theta_2 = atan2\left(\sqrt{p_x^2 + p_y^2}, {}^1p_{5z}\right) \pm \varphi \tag{2.2.2}$$

$$c_4 = \frac{\|{}^1p_5\|^2 - d_3^2 - d_5^2}{2d_3d_5}$$

$$\theta_4 = atan2\left(\pm\sqrt{1 - c_4^2}, c_4\right) \tag{2.2.3}$$

Once $\theta_1, \theta_2, \theta_3, \theta_4$ are known, the orientation matrix of the wrist can be calculated as following

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

$$\theta_6 = atan2\left(\pm\sqrt{1 - k_y^2}, k_y\right) \tag{2.2.4}$$

$$\theta_7 = atan2\left(-j_y, i_y\right)$$

$$\theta_5 = atan2\left(-k_z, k_x\right)$$

2.2.2 Workspace constraints & Singularity points

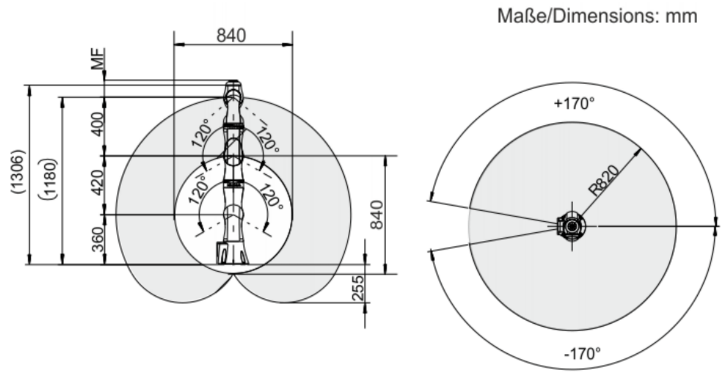

Figure 3: KUKA iiwa LBR14 workspace dimensions

Singularity points:

- When $p_x^2 + p_y^2 = 0$ then the end-effector lies on the z-axis and θ_1 is not defined
- When $sin\left(\theta_6\right) = 0$ then the angles θ_5, θ_7 are not defined

2.2.3 Solutions for 7DoF numerically

Jacobian

$$J = J(\mathbf{q}) = [J_1, J_2, \cdots, J_7] \in \mathbb{R}^{6\times 7}$$

$$J_i = \begin{bmatrix} {}^0\mathbf{z}_i \times ({}^0\mathbf{p}_8 - {}^0\mathbf{p}_i) \\ {}^0\mathbf{z}_i \end{bmatrix} \tag{2.2.5}$$

$J(\mathbf{q})$ is non rectangular and thus non-invertible. Instead of the inverse of the Jacobian the pseudoinverse is calculated which by the equation

$$J^\dagger = J^\top (JJ^\top)^{-1} \tag{2.2.6}$$

2.2.4 Comparison of Inverse Kinematics Techniques

3 Grasping

3.1 Gripper & Forward Kinematics

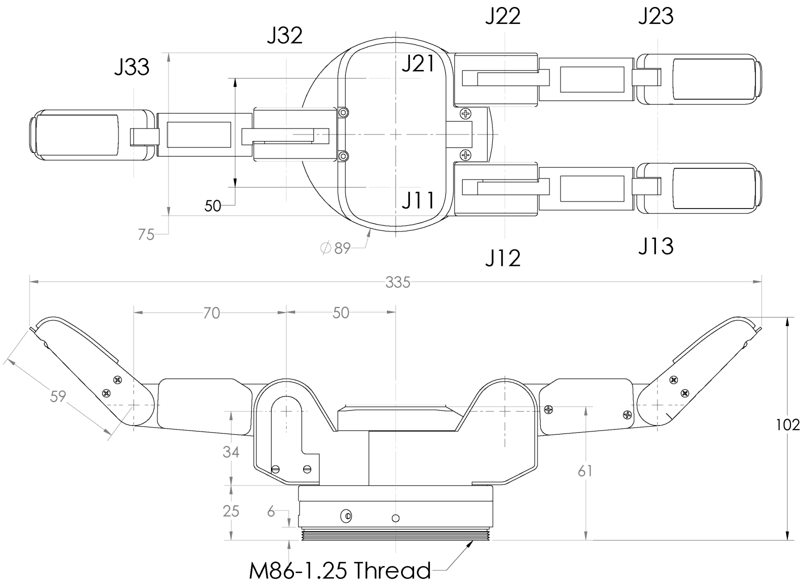

Figure 4: Barrett Hand gripper (model BH8-282) dimensions

3.2 Gripper Inverse Kinematics

The following Inverse Kinematics analysis referes to one finger of the Barrett Hand gripper, which has 3 revolute joints. Finger 3 has only 2 revolute joints for which the angle solutions are the same with the solutions of the last 2 joints of the other fingers. Let

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

be the position of the grasp point for one finger. The first angle can easily be calculated as

$$\varphi_1 = atan2\left(p_y, p_x\right) \tag{3.2.1}$$

Next, we calculate the third angle based on the law of cosines (see fig.)

$$cos\left(\pi - \varphi_3 - \frac{\pi}{4}\right) = \frac{L_2^2 + L_3^2 - p^2}{2L_2L_3}$$

$$cos\left(\varphi_3 + \frac{\pi}{4}\right) = \frac{p^2 - L_2^2 - L_3^2}{2L_2L_3}$$

$$\varphi_3 = atan2\left[\pm\sqrt{1 - \left(\frac{p^2 - L_2^2 - L_3^2}{2L_2L_3}\right)^2}, \frac{p^2 - L_2^2 - L_3^2}{2L_2L_3}\right] - \frac{\pi}{4} \tag{3.2.2}$$

In a more general case, the first argument of the $atan2$ function in the expression of φ_3 could also be negative, but in this case this second solution is rejected, because due to mechanical constraints, this angle can't be negative. After having calculated φ_3 we can calculate φ_2

$$tan\left(\psi+\varphi_2\right)=\frac{p_z}{\sqrt{p_x^2+p_y^2}}$$

$$tan\left(\psi\right)=\frac{L_3 sin\left(\varphi_3+\frac{\pi}{4}\right)}{L_2+L_3 cos\left(\varphi_3+\frac{\pi}{4}\right)}$$

$$\varphi_2=atan2\left(pz,\sqrt{p_x^2+p_y^2}\right)-atan2\left[L_3 sin\left(\varphi_3+\frac{\pi}{4}\right),L_2+L_3 cos\left(\varphi_3+\frac{\pi}{4}\right)\right] \tag{3.2.3}$$

3.3 Force closure

The planar case, the spatial case & convex hull test.

3.4 Firm grasping algorithm & Force control

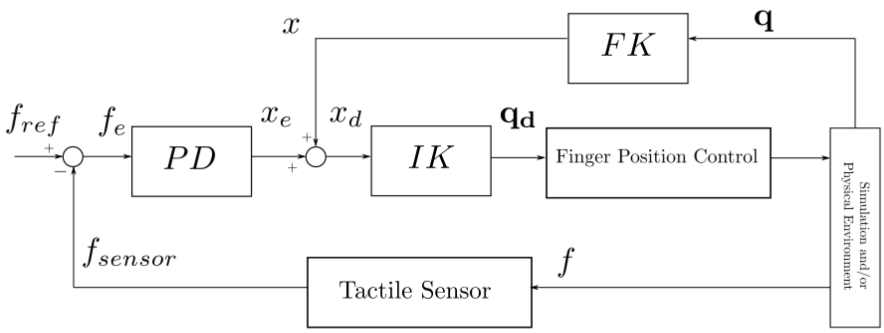

Figure 5: Force control on a Barrett Hand gripper finger

4 Scene and object recognition with Computer Vision

4.1 Laparoscopic tool detection

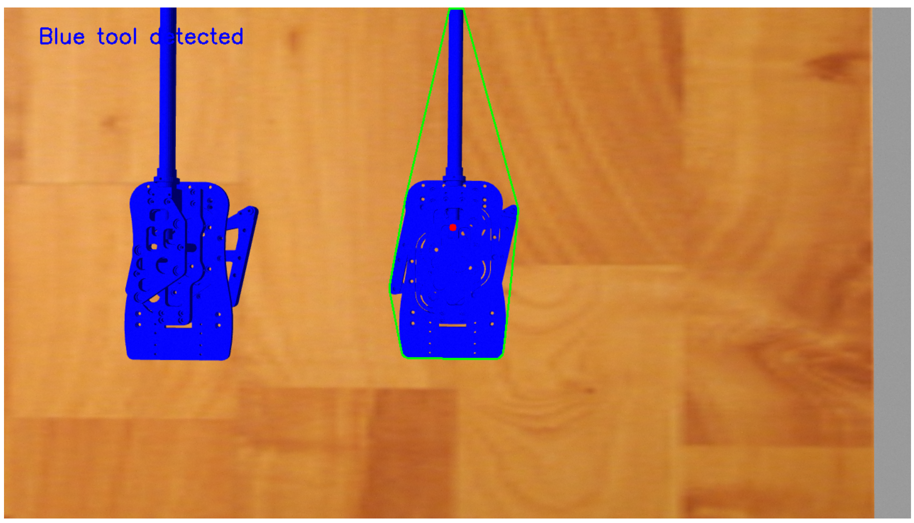

Figure 6: Simple tool detection in simulation based on color, using OpenCV. The green polygon is the convex hull, and the red point is the estimated center of mass

4.2 Calculation of tool position and orientation

In order for the gripper to grasp correctly the laparoscopic tool, it is required to calculate the tool's position and orientation in the pixel space which must then be converted with respect to the robot's workspace. From all the pixels that have been classified as part of the laparoscopic tool, one can estimate the center of mass and two perpendicular vectors attached to that point that define the orientation. The center of mass is simply the average of the (x, y) coordinates of all the tool's pixels

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

The two orientation vectors are the eigenvectors of the covariance matrix of the above pixels. Let $\mathbf{a}, \mathbf{b}$ be the orientation vectors, then $\mathbf{a}, \mathbf{b}$ are solutions of the equation

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

where C is the covariance matrix given by

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

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

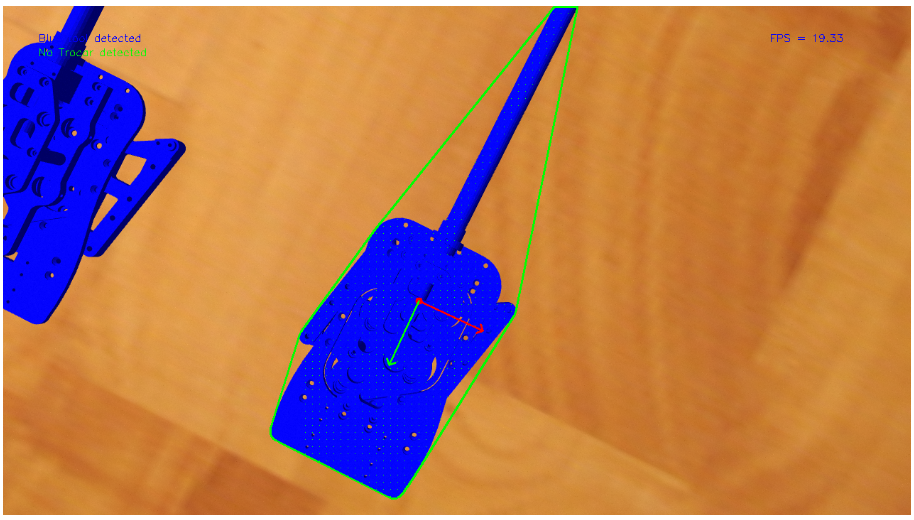

Figure 7: Estimation of tool's pose (position and orientation)

4.3 Calculation of grasping points

4.4 Trocar detection & Estimation of fulcrum point

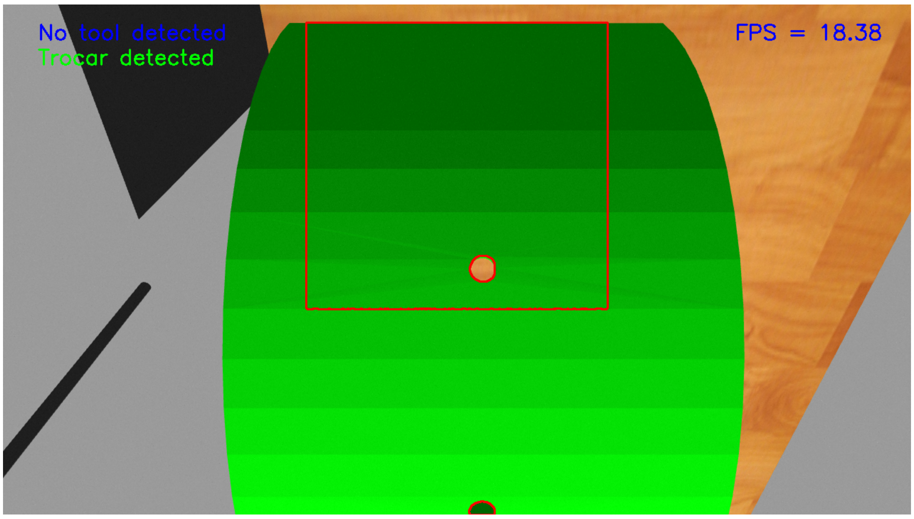

Figure 8: Simple trocar detection in simulation based on color, using OpenCV. In simulation, the trocar is simply considered to be a small cylindrical hole and it's center is the fulcrum point

5 Laparoscopic tool manipulation

5.1 Tool pose

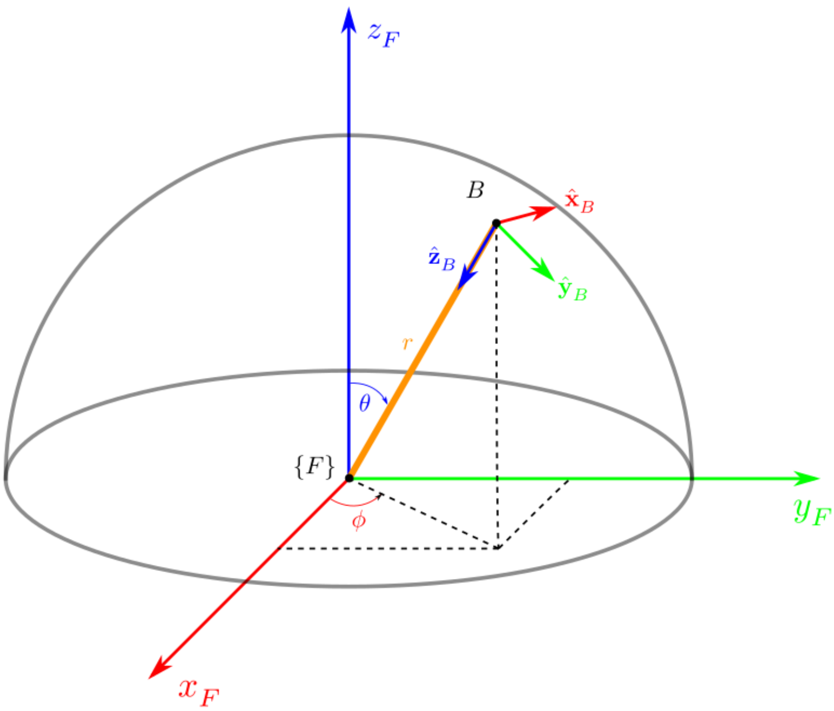

Figure 9: Tool pose at target point B calculated with respect to Fulcrum's reference frame $\{F\}$

The laparoscopic tool pose is given by the position and orientation vectors at target point B with respect to the coordinate frame $\{F\}$. The pose is given by the following transformation matrix

$$^{F}T_{B} = \begin{bmatrix} ^{F}R_{B} & ^{F}\mathbf{p}_{B} \\ \mathbf{0} & 1 \end{bmatrix} \quad where \ \ ^{F}R_{B} = \begin{bmatrix} \hat{\mathbf{x}}_B & \hat{\mathbf{y}}_B & \hat{\mathbf{z}}_B \end{bmatrix}$$

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

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

$$\hat{\mathbf{z}}_B = -\hat{\mathbf{r}} = -(sin(\theta)cos(\varphi)\hat{\mathbf{x}}_F + sin(\theta)sin(\varphi)\hat{\mathbf{y}}_F + cos(\theta)\hat{\mathbf{z}}_F) = \begin{bmatrix} -sin(\theta)cos(\varphi) \\ -sin(\theta)sin(\varphi) \\ -cos(\theta) \end{bmatrix}$$

The position of the point B is given in spherical coordinates by:

- $r = \rho$: outside penetration of laparoscopic tool
- $\theta = \beta$: altitude angle
- $\varphi = \alpha$: orientation angle

thus the position with respect to the coordinate frame $\{F\}$ is given by

$$^{F}\mathbf{p}_B = \begin{bmatrix} \rho sin(\beta)cos(\alpha) \\ \rho sin(\beta)sin(\alpha) \\ \rho cos(\beta) \end{bmatrix} = \rho\hat{\mathbf{r}}$$

The above goal point must be the same as the TCP point of the robot's end-effector. This means, that this pose must be converted with respect to the robot's reference frames.

$$\begin{aligned} ^{U}T_{TCP} &= {}^{U}T_{B} \\ ^{U}T_{0}\,{}^{0}T_{7}\,{}^{7}T_{TCP} &= {}^{U}T_{F}\,{}^{F}T_{B} \\ ^{0}T_{7} &= {}^{U}T_{0}^{-1}\,{}^{U}T_{F}\,{}^{F}T_{B}\,{}^{7}T_{TCP}^{-1} \end{aligned} \tag{5.1.1}$$

5.2 Pivoting motion with respect to Fulcrum Point

5.2.1 Circular trajectory of tool tip

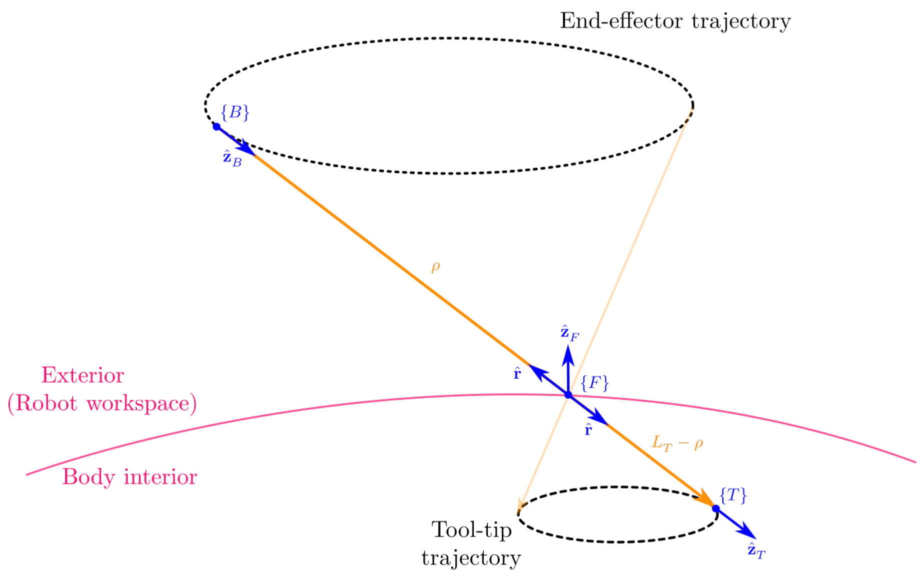

Figure 10: Circular trajectory of tool tip with respect to Fulcrum reference frame

To generate a circular trajectory for the pivot movement we must specify the center of the circle and a vector whose magnitude is the radius of the circle and it's direction gives the orientation of the plane that the circle lies at. The simplest case of a circular trajectory is the one, whose circle lies in a plane parallel to the xy plane.
We first consider the motion of the laparoscopic tool tip on a circle parallel to a z-plane, with respect to the $\{F\}$ coordinate frame.

$$(x_F - x_{F0})^2 + (y_F - y_{F0})^2 = r_0^2, \quad z_F = z_{F0}$$

It's often more convenient to express trajectories in a parametric form, which makes it easier to calculate all the waypoints of the trajectory

$$\begin{cases} x_F = r_0 cos(2\pi t) + x_{F0} \\ y_F = r_0 sin(2\pi t) + y_{F0} \\ z_F = z_{F0} \end{cases} \quad , \quad t \in [0, 1]$$

After having calculated the cartesian coordinates we can calculate the spherical coordinates as follows

$$\begin{cases} r = \sqrt{x_F^2 + y_F^2 + z_F^2} \\ \theta = atan2\left(\sqrt{x_F^2 + y_F^2}, z_F\right) \\ \varphi = atan2(y_F, x_F) \end{cases} \tag{5.2.1}$$

5.2.2 Circular arc trajectory of tool tip

To generate a circular arc trajectory for a pivot motion we must specify the same parameters as in the circular trajectory as well as the length of the arc or the total angle of the arc section.

5.2.3 Line segment trajectory of tool tip

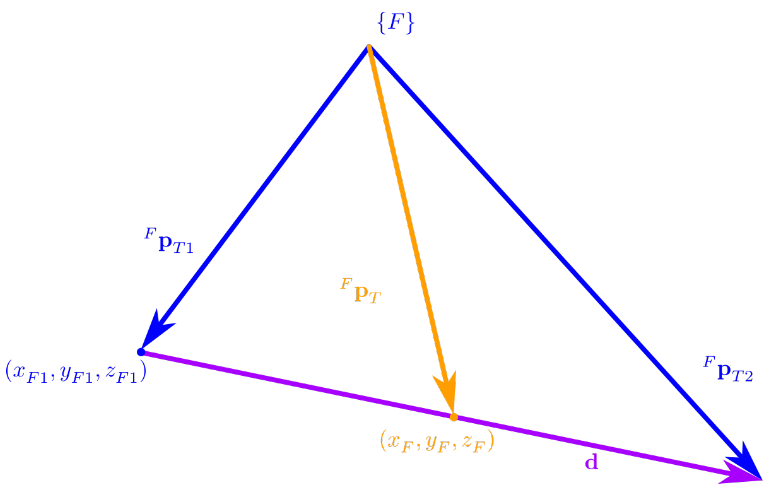

Figure 11: Line segment trajectory of tool tip with respect to Fulcrum reference frame

$$\mathbf{d} = {}^F\mathbf{p}_{T2} - {}^F\mathbf{p}_{T1} = [l, m, n]^\top$$
$${}^F\mathbf{p}_T = [x_F, y_F, z_F]^\top$$
$${}^F\mathbf{p}_T = {}^F\mathbf{p}_{T1} + t\mathbf{d}$$

$$t = \frac{x_F - x_{F1}}{l} = \frac{y_F - y_{F1}}{m} = \frac{z_F - z_{F1}}{n} \quad t \in [0, 1]$$

$$\begin{cases} x_F = tl + x_{F1} \\ y_F = tm + y_{F1} \\ z_F = tn + z_{F1} \end{cases}$$

After having calculated the cartesian coordinates we can calculate the spherical coordinates using the 5.2.1 equations.

The line segment trajectory of tool tip, as analysed in this section needs no implementation as it is already implemented in the ROS MoveIt library and can be used by calling the method **computeCartesianPath**.

5.3 Task space analysis

Dexterity analysis for tool's task space

$$\mathcal{D} = \mathcal{L}_q \mathcal{M} \tag{5.3.1}$$

where

$$\mathcal{M} = \sqrt{det(JJ^\top)} \tag{5.3.2}$$

$$\mathcal{L}_q = 1 - \exp\left\{ -\kappa \prod_{i=1}^{n_k} \frac{(q_i - q_{i,\min})(q_{i,\max} - q_i)}{(q_{i,\max} - q_{i,\min})^2} \right\} \tag{5.3.3}$$

For maximum dexterity at most points of a trajectory in a pivoting motion, the pivot sub- taskspace (i.e. the space of all configurations of feasible pivot motions) must be fully within the robot's whole reachable taskspace, otherwise only a small range of pivot movements will be feasible.

6 Path Planning

6.1 Path searching

Find path points (position and orientation) by avoiding collisions, asserting that path points is within robot's workspace and by avoiding singularity points.

6.2 Pick and place algorithm

7 Trajectory Planning

At this step

7.1 Trajectory planning in cartesian coordinates

Connect the points from path planning with line segments and add more points if needed

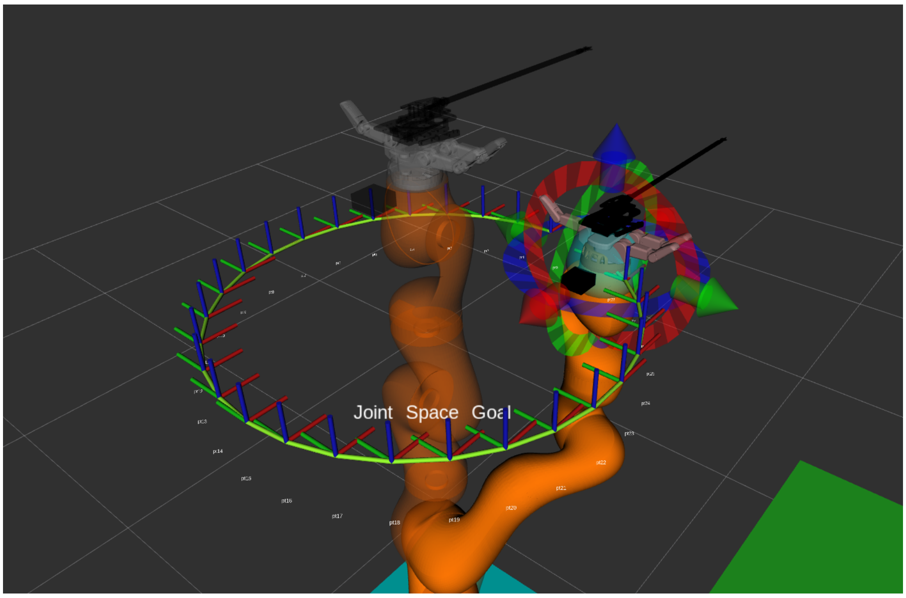

Figure 12: Circlular trajectory around the z zxis of the home position of the robot

It is very important that the designed trajectory respects the joints' angles' range. For example depending on the starting position of the circular trajectory depicted at figure 12, the robot arm may reach it's joint bounds and in order to continue the trajectory it will have to make a sudden jump to reset the angles. This could have serious side-effects for both the surgical task and thus the patient, as well as for the operating staff, who control the robot.

7.2 Trajectory planning in joint angles space

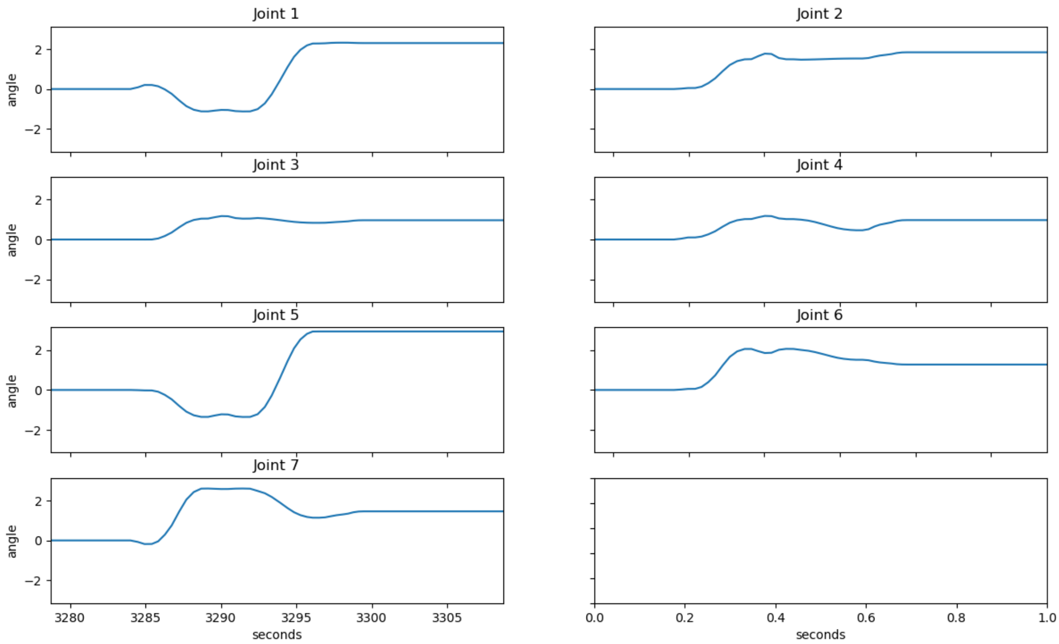

Figure 13: Trajectory diagrams in joints space.

8 Implementation with the ROS framework

8.1 Gazebo simulation environment

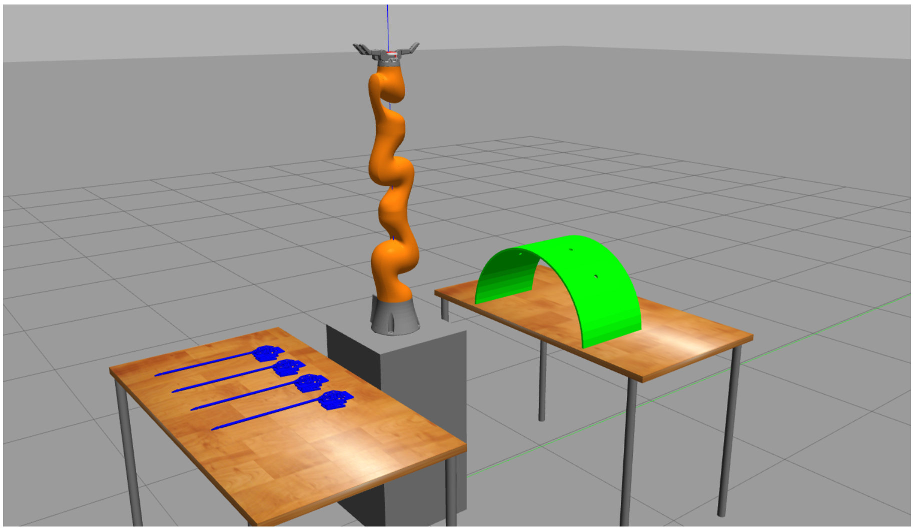

Figure 14: Simulation environment in Gazebo

The main environment setup of this thesis was designed using the Gazebo simulation environment and it consists of the following objects:

- the robot arm, KUKA®iiwa14 lbr, being at the center of the setup
- the robot base, so that the robot arm can better reach the tools and the surgical site and have more flexibility in movment
- 2 tables, one for the tools and one for the surgical site
- 4 surgical tools, using a modified version of the surgical tools used in the Raven II surgical platform
- a mounting dock, which has holes that have the same role as the trocars (small tubes from which the surgical tool is inserted). Initially a mounting dock with 4 same holes of 4mm diameter was used, but it was later replaced with a new one with holes of variable diameters to test feasibility of pivot motions. Larger diameters means more space for motion planner to search for solution and thus more probable to find a solution.

8.2 Visualization and Motion Planning with RViz and Moveit

Motion Planning parameters outside of body:

- Position tolerance: 50µm
- Orientation tolerance: 0.00005 deg
- Planning time: 10s

Motion Planning parameters inside of body:

- Position tolerance:
- Orientation tolerance:

- Planning time
- End-effector interpolation step: 1mm
- Maximum velocity scaling factor

Sometimes the motion planner finds a solution but the execution from the controller is aborted. After many iterations of the same experiment this does not happen always, which means that the feasibility of the execution of the movement by the controller depends on the initial state of the robot, i.e. if initially some joints of the robot are at their boundaries, then the next commanded trajectory maybe unfeasible.

At each time step it is important to publish a custom message containing all the information about the kinematic state of the robot. In this thesis a custom **ROS** message was created containing a tf transform with a 3D vector for the position and a quaternion for the rotation and a custom 6-by-7 matrix containing the values of the Jacobian. The MoveIt library, from which the kinematic state of the robot is obtained, returns the orientation of the end effector as a 3-by-3 rotation matrix, but in the ROS tf message it must be expressed as a quaternion. To convert the matrix to a quaternion we first calculate the euler angles and then use these values to construct the quaternion "vector". The quaternion representation of rotation is often preferred in robotic applications due to its efficiency in calculations and memory. To convert the transformation matrix to euler angles and then to quaternions the following formulas were used:

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\varphi = atan2(r_{21}, r_{11})$$

$$\theta = atan2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2})$$

$$\psi = atan2(r_{32}, r_{33})$$

where T is the transformation matrix and φ, θ, ψ are the roll, pitch and yaw (Euler) angles.

8.3 Experiments and Development methodology

8.3.1 Robot Planner 1

8.3.2 Robot Planner 2

8.3.3 Robot Planner 3

8.3.4 Robot Planner 4

8.3.5 Robot Planner 5

Nomenclature

$\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}$ Unit vectors of r, θ, φ axes respectively, in spherical coordinates

$\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ Unit vectors of x, y, z axes respectively

$\mathcal{L}_q$ Dexterity measure of the robotic arm

$\mathcal{M}$ Manipulability measure of the robotic arm

${}^{i-1}\mathbf{p}_{iO}$ Position vector from the origin of the coordinate frame $\{i\}$ to the origin of the coordinate frame $\{i-1\}$

${}^{i-1}R_i$ Rotation matrix from coordinate frame $\{i\}$ to coordinate frame $\{i-1\}$

${}^{i-1}T_i$ Transformation matrix from coordinate frame $\{i\}$ to coordinate frame $\{i-1\}$

c_i Shorthand notation for $cos\theta_i$

$J^{\dagger}$ Pseudoinverse of the Jacobian

s_i Shorthand notation for $sin\theta_i$

List of Figures

1 Joint reference frames of the KUKA iiwa14 robot 5
2 Calculation of angles θ_2, θ_4 6
3 KUKA iiwa LBR14 workspace dimensions 7
4 Barrett Hand gripper (model BH8-282) dimensions 8
5 Force control on a Barrett Hand gripper finger 9
6 Simple tool detection in simulation based on color, using OpenCV. The green polygon is the convex hull, and the red point is the estimated center of mass 9
7 Estimation of tool's pose (position and orientation) 10
8 Simple trocar detection in simulation based on color, using OpenCV. In simulation, the trocar is simply considered to be a small cylindrical hole and it's center is the fulcrum point 11
9 Tool pose at target point B calculated with respect to Fulcrum's reference frame $\{F\}$. 11
10 Circular trajectory of tool tip with respect to Fulcrum reference frame 12
11 Line segment trajectory of tool tip with respect to Fulcrum reference frame 13
12 Circlular trajectory around the z zxis of the home position of the robot 15
13 Trajectory diagrams in joints space. 15
14 Simulation environment in Gazebo 16

List of programs

Bibliography

[1] E. Bauzano et al. "Active wrists endoscope navigation in robotized laparoscopic surgery". In: *2009 IEEE International Conference on Mechatronics*. Apr. 2009, pp. 1–6. DOI: 10.1109/ICMECH.2009.4957177.

[2] M. C. Capolei et al. "Positioning the laparoscopic camera with industrial robot arm". In: *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. Apr. 2017, pp. 138–143. DOI: 10.1109/ICCAR.2017.7942675.

[3] Sachin Chitta et al. "ros_control: A generic and simple control framework for ROS". In: *The Journal of Open Source Software* (2017). DOI: 10.21105/joss.00456. URL: http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf.

[4] Carlos Faria et al. "Position-based kinematics for 7-DoF serial manipulators with global configuration control, joint limit and singularity avoidance". In: *Mechanism and Machine Theory* 121 (2018), pp. 317–334. ISSN: 0094-114X. DOI: https://doi.org/10.1016/j.mechmachtheory.2017.10.025. URL: http://www.sciencedirect.com/science/article/pii/S0094114X17306559.

[5] Tully Foote. "tf: The transform library". In: *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*. Open-Source Software workshop. Apr. 2013, pp. 1–6. DOI: 10.1109/TePRA.2013.6556373.

[6] K. Fujii et al. "Gaze contingent cartesian control of a robotic arm for laparoscopic surgery". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nov. 2013, pp. 3582–3589. DOI: 10.1109/IROS.2013.6696867.

[7] M. Fujii et al. "Relationship between workspace reduction due to collisions and distance between endoscope and target organ in pediatric endoscopic surgery". In: *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. Aug. 2014, pp. 46–51. DOI: 10.1109/BIOROB.2014.6913750.

[8] J. Funda et al. "Control and evaluation of a 7-axis surgical robot for laparoscopy". In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. Vol. 2. 1995, 1477–1484 vol.2.

[9] G. Gras et al. “Implicit gaze-assisted adaptive motion scaling for highly articulated instrument manipulation”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. May 2017, pp. 4233–4239. DOI: 10.1109/ICRA.2017.7989488.

[10] G. B. Hanna, S. Shimi, and A. Cuschieri. “Optimal port locations for endoscopic intracorporeal knotting”. In: *Surgical Endoscop* 11.4 (Apr. 1997), pp. 397–401. ISSN: 1432-2218. DOI: 10.1007/s004649900374. URL: https://doi.org/10.1007/s004649900374.

[11] M. R. Hasan et al. “Modelling and Control of the Barrett Hand for Grasping”. In: *2013 UKSim 15th International Conference on Computer Modelling and Simulation*. Apr. 2013, pp. 230–235. DOI: 10.1109/UKSim.2013.142.

[12] Felix C. Huang et al. “Learning kinematic mappings in laparoscopic surgery”. In: *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference* 2010 (2010). PMC3280950[pmcid], pp. 2097–2102. ISSN: 1557-170X. DOI: 10.1109/IEMBS.2010.5626188. URL: https://pubmed.ncbi.nlm.nih.gov/21095685.

[13] J. Hutzl and H. Wörn. “Spatial probability distribution for port planning in minimal invasive robotic surgery (MIRS)”. In: *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*. Feb. 2015, pp. 204–210. DOI: 10.1109/ICARA.2015.7081148.

[14] Reza N. Jazar. *Theory of Applied Robotics, Kinematics, Dynamics, and Control (2nd Edition)*. Springer, Boston, MA, 2010. ISBN: 978-1-4419-1750-8. DOI: 10.1007/978-1-4419-1750-8.

[15] I. Kuhlemann et al. “Robust inverse kinematics by configuration control for redundant manipulators with seven DoF”. In: *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*. Apr. 2016, pp. 49–55. DOI: 10.1109/ICCAR.2016.7486697.

[16] Kevin M Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. English (US). Cambridge Univeristy Press, 2017. ISBN: 978-1107156302.

[17] Victor F. Muñoz et al. “Pivoting motion control for a laparoscopic assistant robot and human clinical trials”. In: *Advanced Robotics* 19 (2005), pp. 694–712.

[18] A. A. Navarro et al. “Automatic positioning of surgical instruments in minimally invasive robotic surgery through vision-based motion analysis”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2007, pp. 202–207. DOI: 10.1109/IROS.2007.4399150.

[19] Jason M. O’Kane. *A Gentle Introduction to ROS*. Available at http://www.cse.sc.edu/~jokane/agitr/. Independently published, Oct. 2013. ISBN: 978-1492143239.

[20] Jaydeep Palep. “Robotic assisted minimally invasive surgery”. In: *Journal of minimal access surgery* 5 (Feb. 2009), pp. 1–7. DOI: 10.4103/0972-9941.51313.

[21] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA Workshop on Open Source Software*. 2009.

[22] I. A. Şucan and S. Chitta. *Moveit!* URL: https://moveit.ros.org/ (visited on 04/28/2020).

[23] Haiwang Xu et al. “Laparoscopic Robot Design and Kinematic Validation”. In: Jan. 2007, pp. 1426–1431. DOI: 10.1109/ROBIO.2006.340138.

[24] Tao Yang et al. “Mechanism of a Learning Robot Manipulator for Laparoscopic Surgical Training”. In: *Intelligent Autonomous Systems 12*. Ed. by Sukhan Lee et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 17–26. ISBN: 978-3-642-33932-5.

[25] L. Yu et al. “Research on the Trajectory Planning of the Minimally Invasive Surgical Robot 6-DOF Manipulator”. In: *2010 International Conference on Biomedical Engineering and Computer Science*. Apr. 2010, pp. 1–4. DOI: 10.1109/ICBECS.2010.5462451.