External Hybrid Vision-Force control for da Vinci robot using Matlab and V-Rep

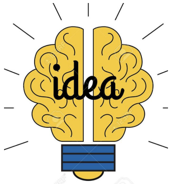
Medical Robotics

SAPIENZA
UNIVERSITÀ DI ROMA

Michele Ciciolla Flavio Lorenzi

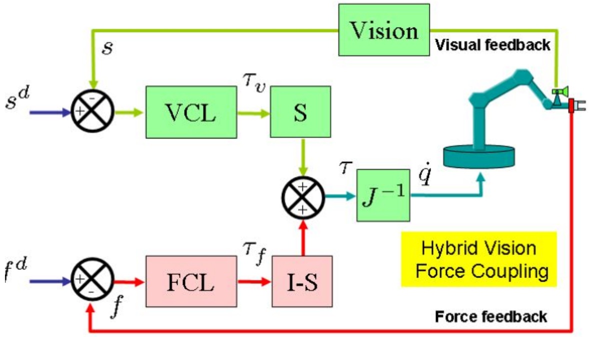
Introduction -1

Previous literature: Martinet and Prats



Coupling the vision control with the force can give us many advantages

The classic hybrid vision-force system works fine *only* if the tasks are well known; so we are limited by this standard configuration



So a **new approach** that goes beyond the limits and problems of these control schemes is presented…

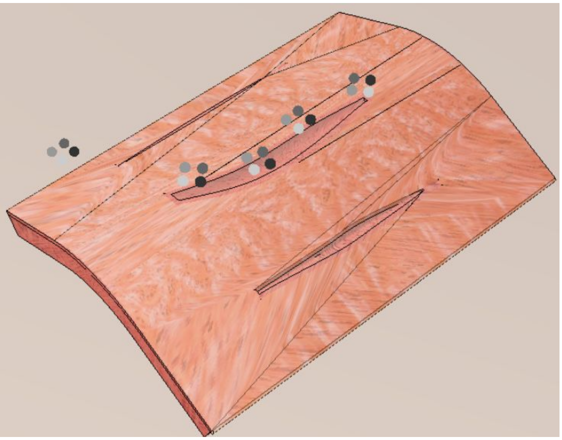
Introduction -2

External Hybrid Vision/Force System on Da Vinci Robot

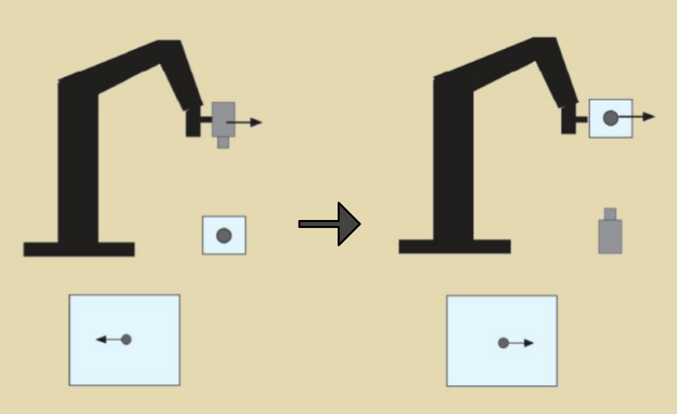
From:

- the general idea of this specific coupling mechanism
- pre-implemented direct kinematics on **eye-in-hand** system

We present a re-implementation of this task in **eye-to-hand** control with **inverse kinematic** to efficiently solve the suturing operation task



Phanom skin on the simulator



From **eye-in-hand** to the **eye-to-hand**

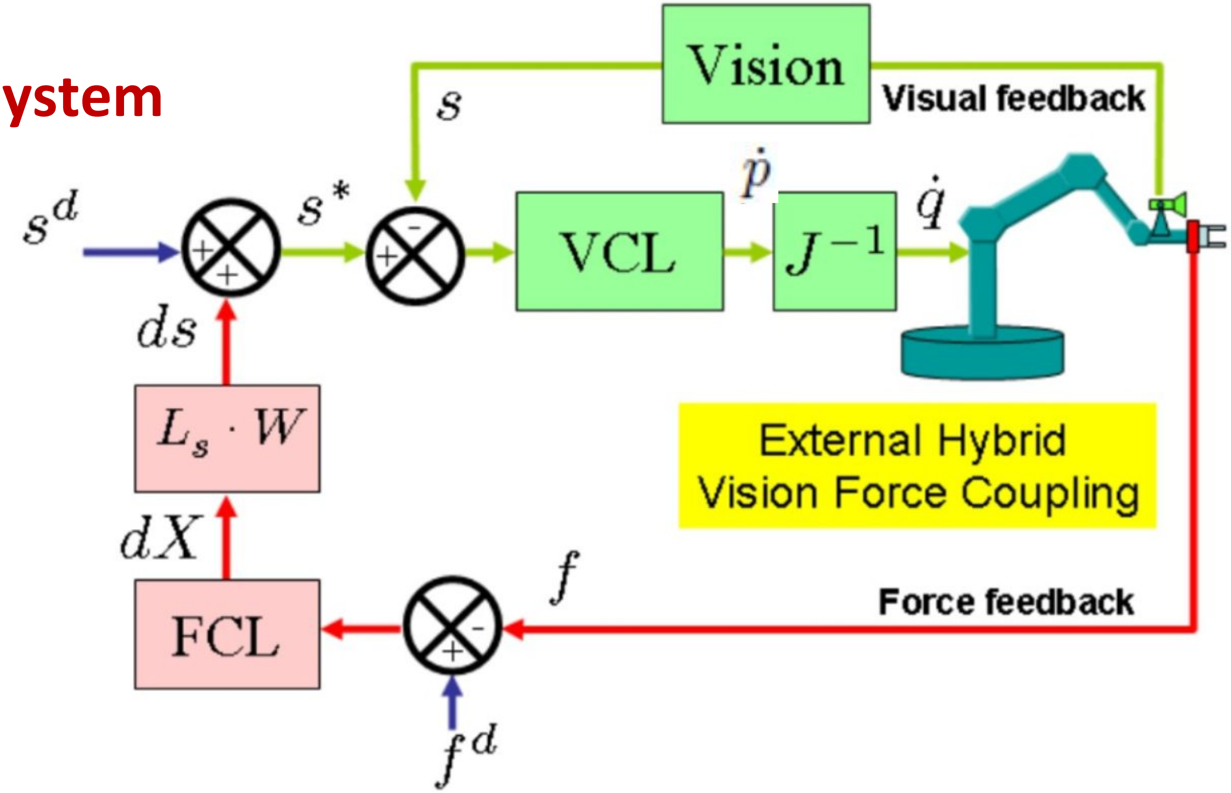
The control scheme

This paper introduces a Control Scheme composed by two main branches:

Vision system

and

Force system



Control scheme: implementation

Coupling is done in sensor space: the reference trajectory generated by visual control is modified by the force control loop

$$\dot{p} = \hat{L}^{\dagger} K(s^* - s)$$

Where $s^* = s^d - \hat{L}WC(f^d - f)$

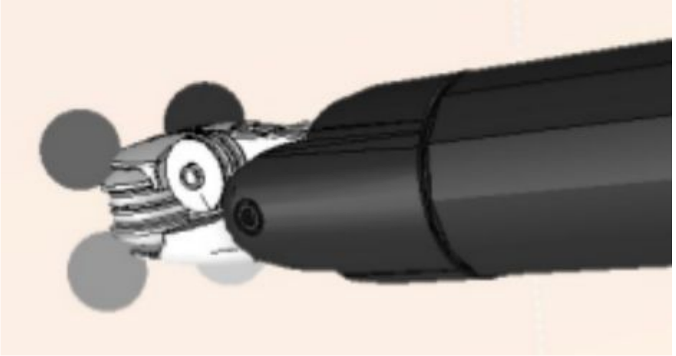
Note: It's quite common to integrating velocity in spatial coordinates when K is composed by small values to find position *p* and update joints

Visual Servoing -1

- Main important aspect in robotics to guarantee motion control
- Data acquisition with very large information content in the acquired images
- Fast information received and real time

Aim:
Feature extraction task and computing the error

$$e = s^d - s \in R^{8\times 1}$$



Visual Servoing -2

The interaction matrix L plays a key role here…

$$\boxed{\dot{U} = L \cdot \dot{p}}$$

… allowing relation between velocities

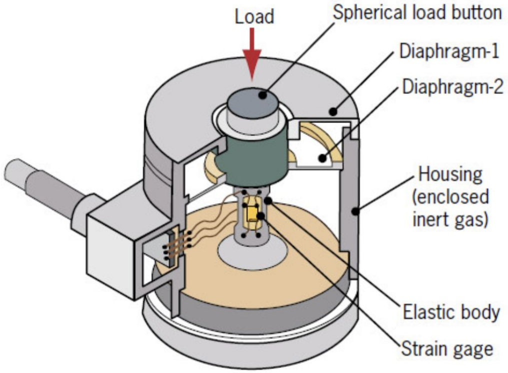
$$L = \begin{pmatrix} \frac{f}{Z_1} & 0 & \frac{u_1}{Z_1} & \frac{u_1 v_1}{Z_1} & -\frac{f^2+u_1^2}{f} & v_1 \\ 0 & \frac{f}{Z_1} & \frac{v_1}{Z_1} & \frac{f^2+v_1^2}{f} & -\frac{u_1 v_1}{Z_1} & -u_1 \\ \frac{f}{Z_2} & 0 & \frac{u_2}{Z_2} & \frac{u_2 v_2}{Z_2} & -\frac{f^2+u_2^2}{f} & v_2 \\ 0 & \frac{f}{Z_2} & \frac{v_2}{Z_2} & \frac{f^2+v_2^2}{f} & -\frac{u_2 v_2}{Z_2} & -u_2 \\ & & & \vdots & & \\ \frac{f}{Z_n} & 0 & \frac{u_n}{Z_n} & \frac{u_n v_n}{Z_n} & -\frac{f^2+u_n^2}{f} & v_n \\ 0 & \frac{f}{Z_n} & \frac{v_{1n}}{Z_n} & \frac{f^2+v_n^2}{f} & -\frac{u_n v_n}{Z_n} & -u_n \end{pmatrix}$$

Image processing should present some problems:

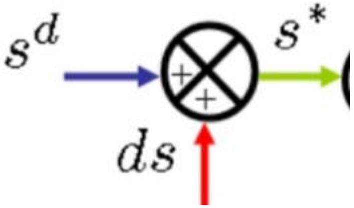
- Difficult to extract essential data;
- Nonlinear perspective transformations
- High sensitivity to ambient conditions (lightening)
- Noise…

The force sensor

Introduction of a sensor that can solve problems when only the camera cannot

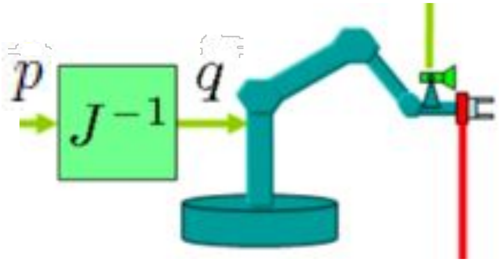


- Receiving in input the force when there is a contact
- Computing the error *(fd – f)*
- Receiving as output a force-based image correction *ds*



- We obtain the desired feature *s** for the direct chain

Inverse kinematics



$p = L^{\dagger}K(s^* - s)$

vision

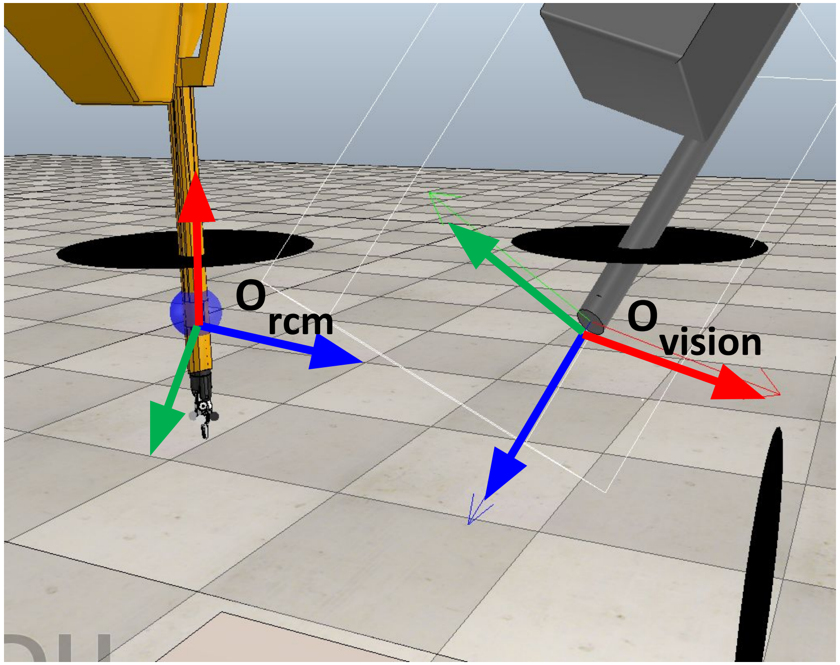
getPoseInRCM

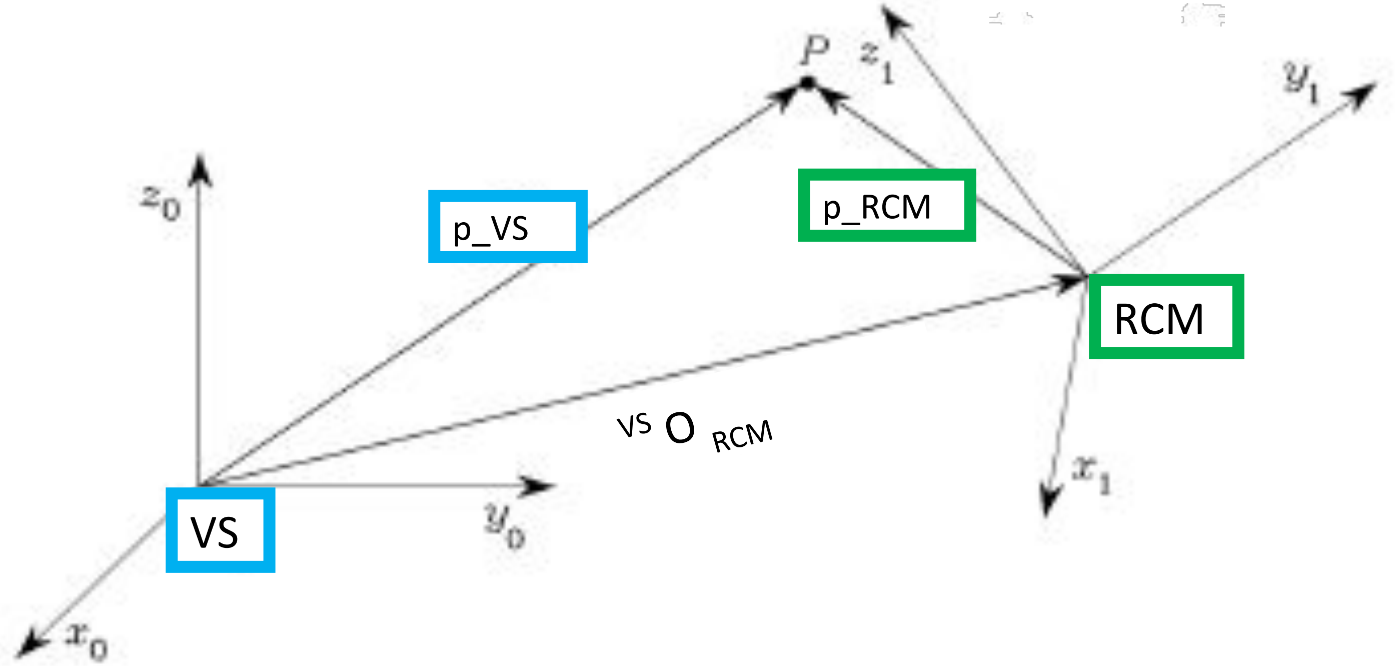
p

rcm

inverseKinematics

q





$$\text{p\_VS} = {}^{VS}O_{RCM} + {}^{VS}R_{RCM}\ \underset{\text{?}}{\text{P\_RCM}} \rightarrow \boxed{\text{P\_RCM} = -\,{}^{VS}O_{RCM}\ ({}^{VS}R_{RCM})^T + ({}^{VS}R_{RCM})^T\ \text{p\_VS}}$$

known **known** **known**

$$\boxed{({}^{VS}R_{RCM})^T\ {}^{VS}R_{EE} = {}^{RCM}R_{EE}}$$

p_RCM □ **getPoseInRCM** ($^{VS}O_{RCM}$, p_VS)

p → q

Inverse kinematics

```
err_RCM ← next_pose_RCM - current_pose_RCM
Q_k = getJoints()

Q_k+1 □ inverseKinematics ( Q_k , err_RCM )

sendToJoints(Q_k+1 )
```

$$q^{k+1} = q^k + J_r^{-1}(q^k)[r_d - f_r(q^k)]$$

$$q^{k+1} = q^k + \alpha J_r^{\dagger}(q^k)[r_d - f_r(q^k)]$$

Newton method

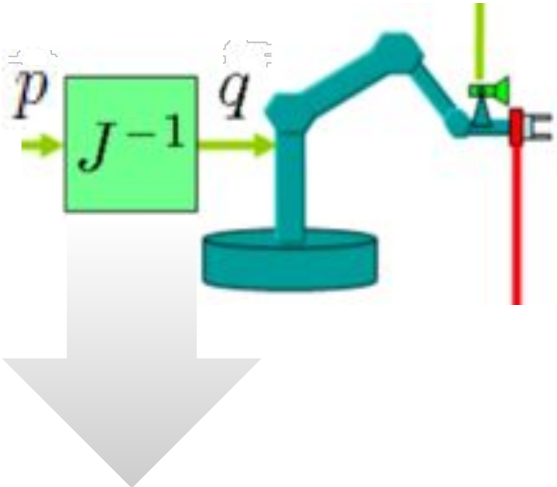
- Computationally fast
- Quadratic convergence rate

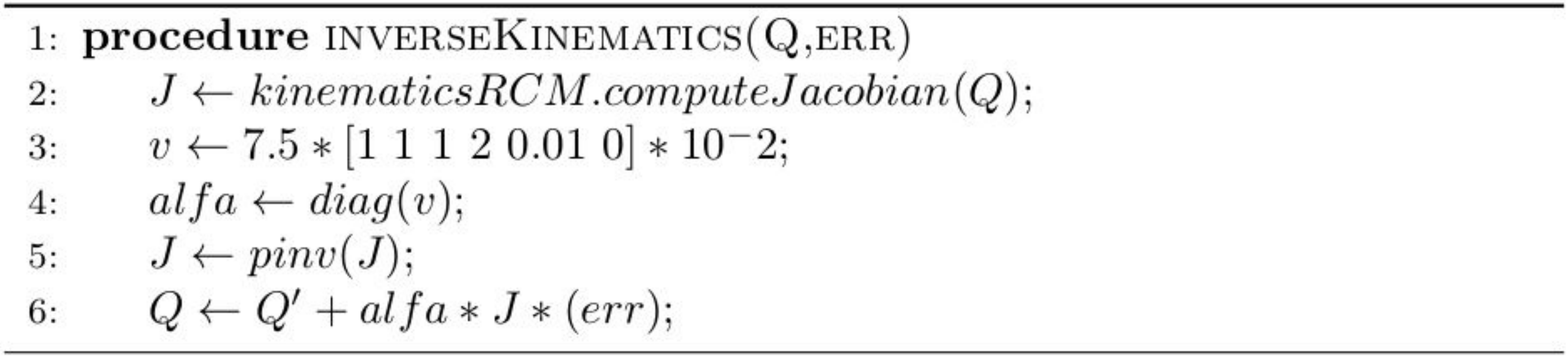
+ Costant vector alpha

A V-REP Simulator for the da Vinci Research Kit Robotic Platform

G. A. Fontanelli[1], M. Selvaggio[1], M. Ferro[2], F. Ficuciello[1], M. Vendittelli[3] and B. Siciliano[1]

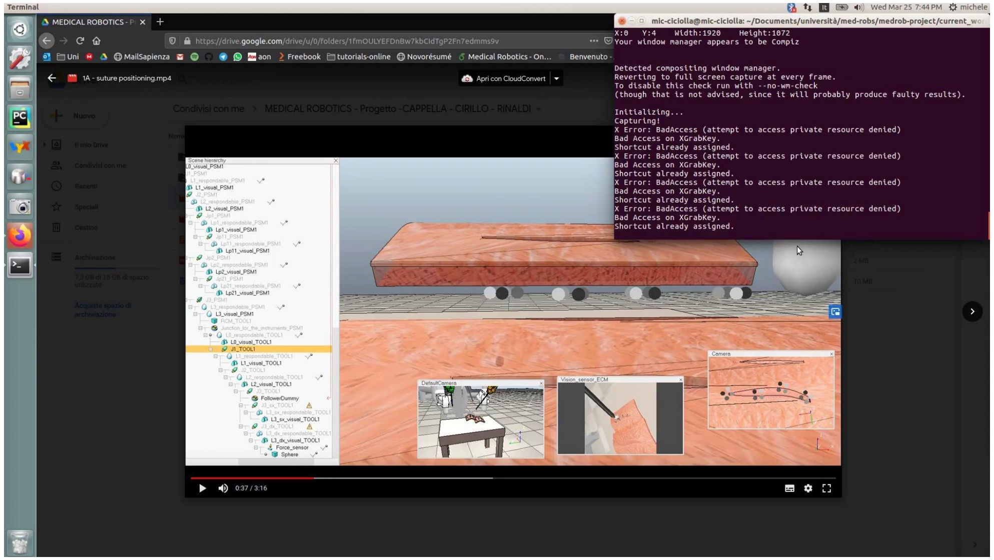
Inverse kinematics recap



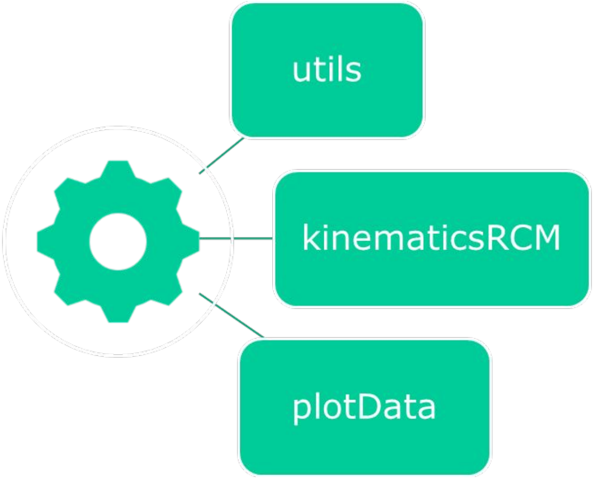


```
procedure INVERSEKINEMATICS(Q,ERR)
    $J \leftarrow kinematicsRCM.computeJacobian(Q);$
    $v \leftarrow 7.5 * [1\ 1\ 1\ 2\ 0.01\ 0] * 10^{-}2;$
    $alfa \leftarrow diag(v);$
    $J \leftarrow pinv(J);$
    $Q \leftarrow Q' + alfa * J * (err);$
```

Tuning and code reorganization


Terminal
MEDICAL ROBOTICS
1A - suture positioning.mp4
Apri con CloudConvert
Condivisi con me > MEDICAL ROBOTICS - Progetto -CAPPELLA - CIRILLO - RINALDI
Wed Mar 25 7:44 PM
michele
mic-ciciolla@mic-ciciolla: ~/Documents/università/med-robs/medrob-project/current_wo
0:37 / 3:16

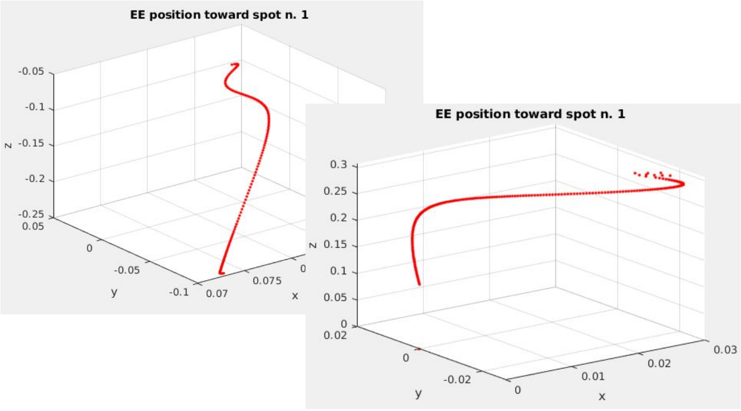
```
X:0   Y:4   Width:1920   Height:1072
Your window manager appears to be Compiz


Detected compositing window manager.
Reverting to full screen capture at every frame.
To disable this check run with --no-wm-check
(though that is not advised, since it will probably produce faulty results).

Initializing...
Capturing!
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.
X Error: BadAccess (attempt to access private resource denied)
Bad Access on XGrabKey.
Shortcut already assigned.
```


utils

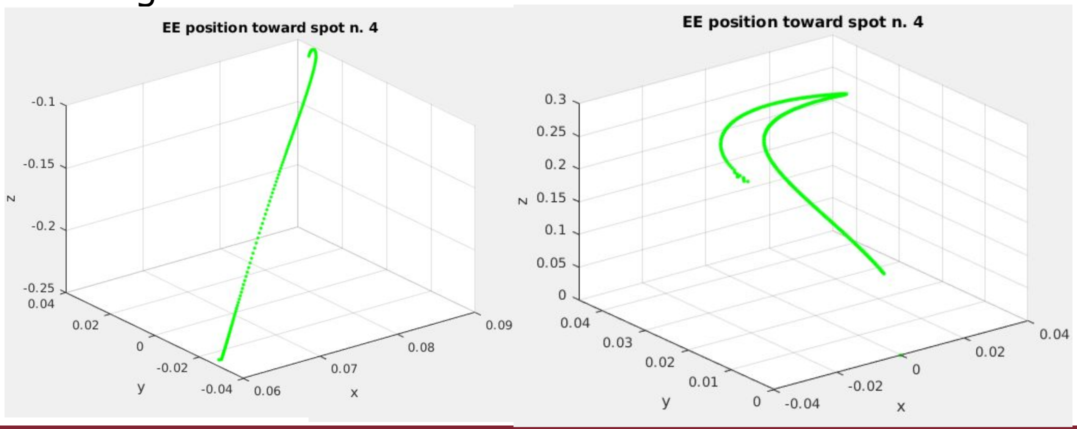
kinematicsRCM

plotData

tuning



tuning



Software section: Matlab and VREP

Why Matlab?

Easy connection with VREP and many API functions available;

Ability to use technical language and mathematical operations easily;

Better computing performance and ease of deployment;

Why VREP?

The tree structure allows the creation of nodes and dependences;

We can also call external functions implemented in Lua inside the simulator nodes to guarantee connection.

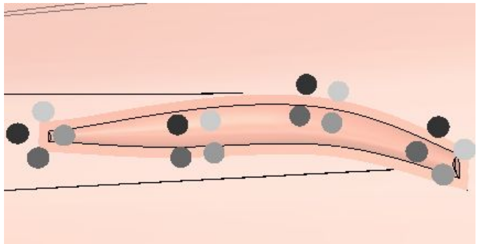
Matlab init functions

Init connection with matlab and handles

[~, handle_FS] = vrep.simxGetObjectHandle(ID, 'Force_sensor', vrep.simx_opmode_blocking)

Connection between nodes to comunicate

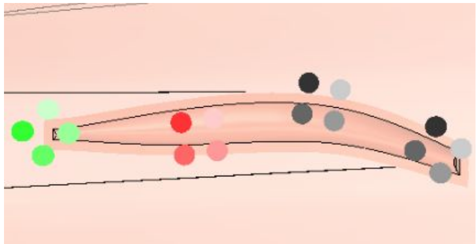
function [sync] = syncronize(ID, vrep, h_joints, h_RCM, h_VS, h_Followed, h_EE, h_FS)



change_color() off

Connection between matlab and Lua childscript functions

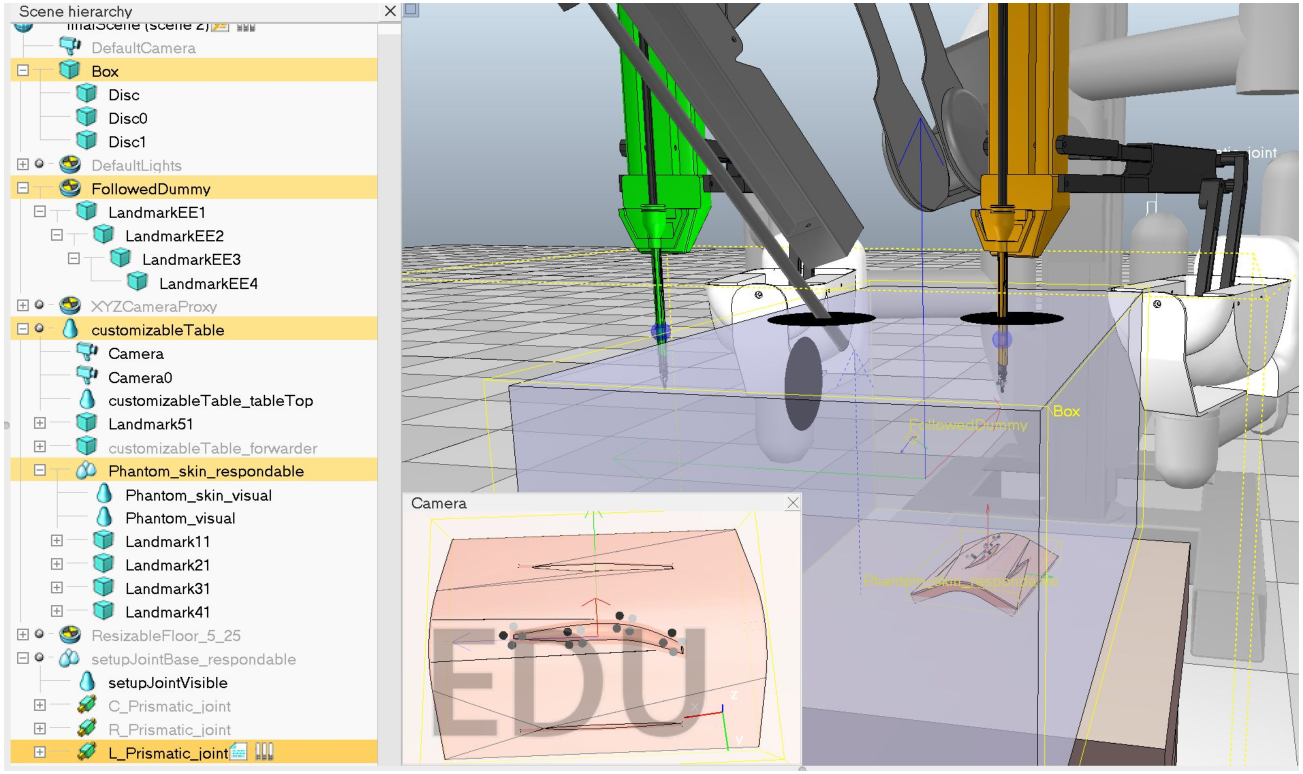
vrep.simxCallScriptFunction()



change_color() on

V-Rep environment

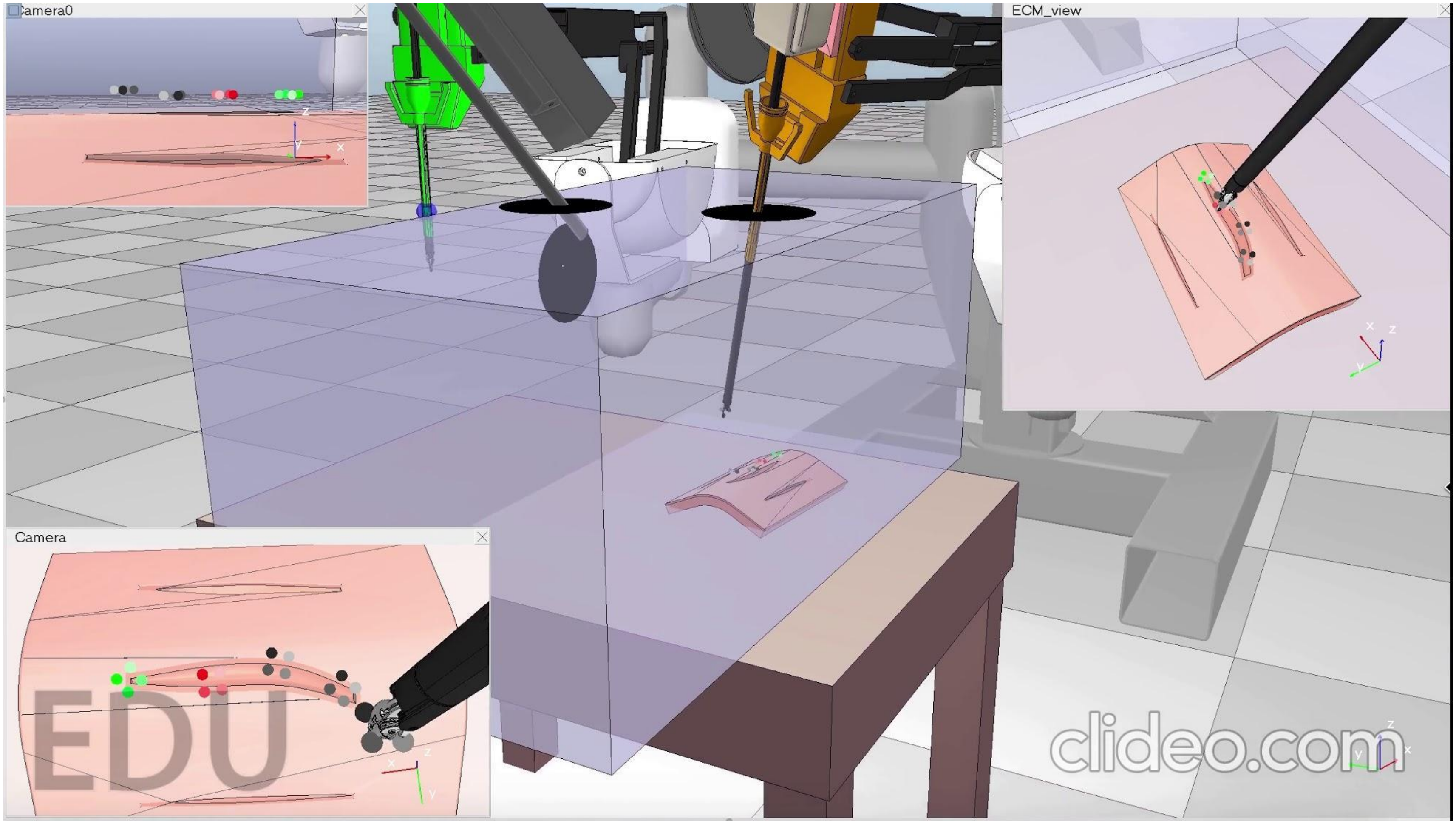
We implemented the official Sapienza env for Da Vinci robot:



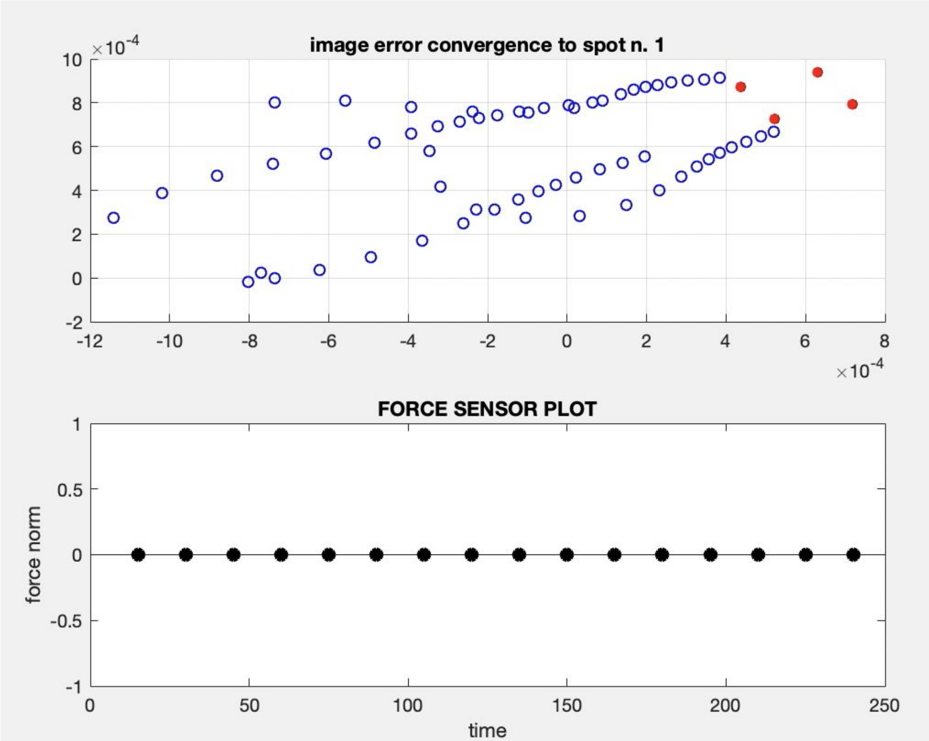
The main important nodes are highlighted

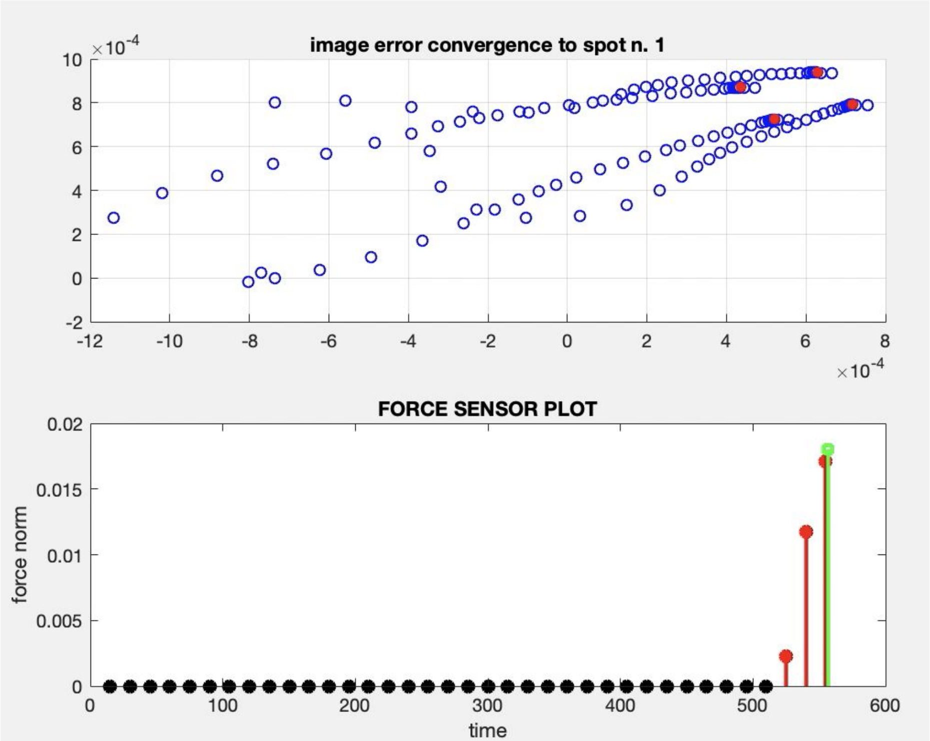
The introduction of a transparent box gives us the idea of “reality” when the robot is operating

Simulation video



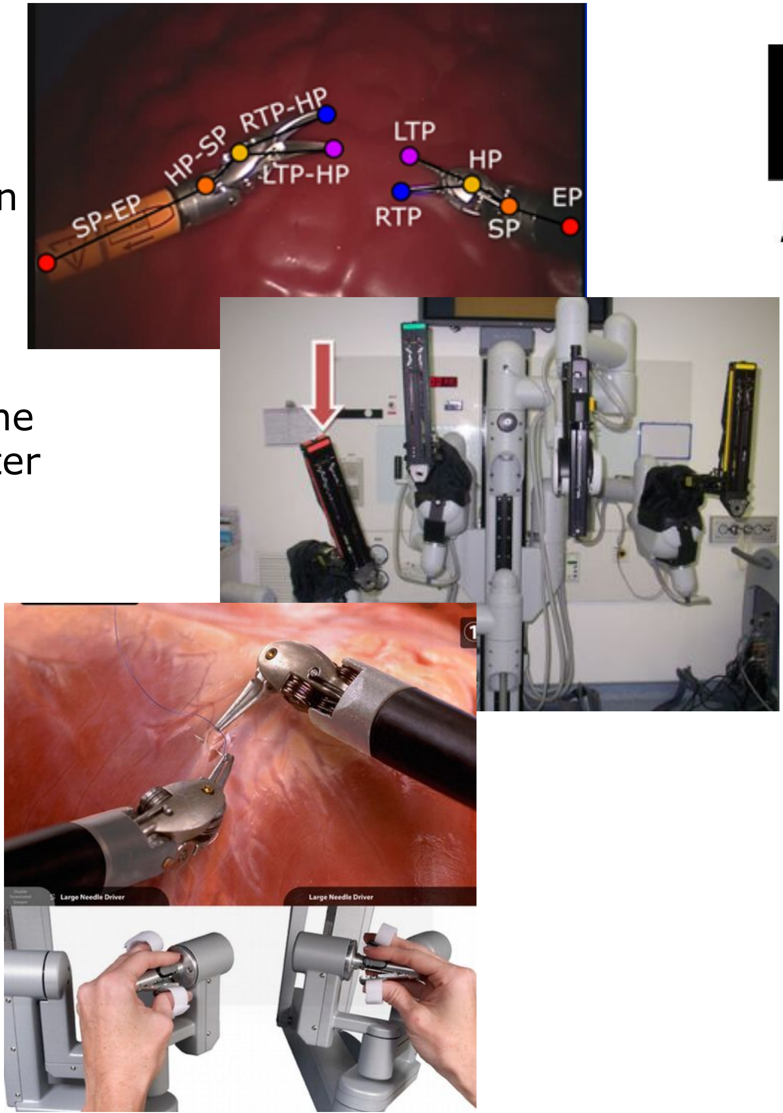
Result achieved: plots commenting





Important for the user: easy to understand and quick check of results in real time

Final thoughts and future works



AprilTag

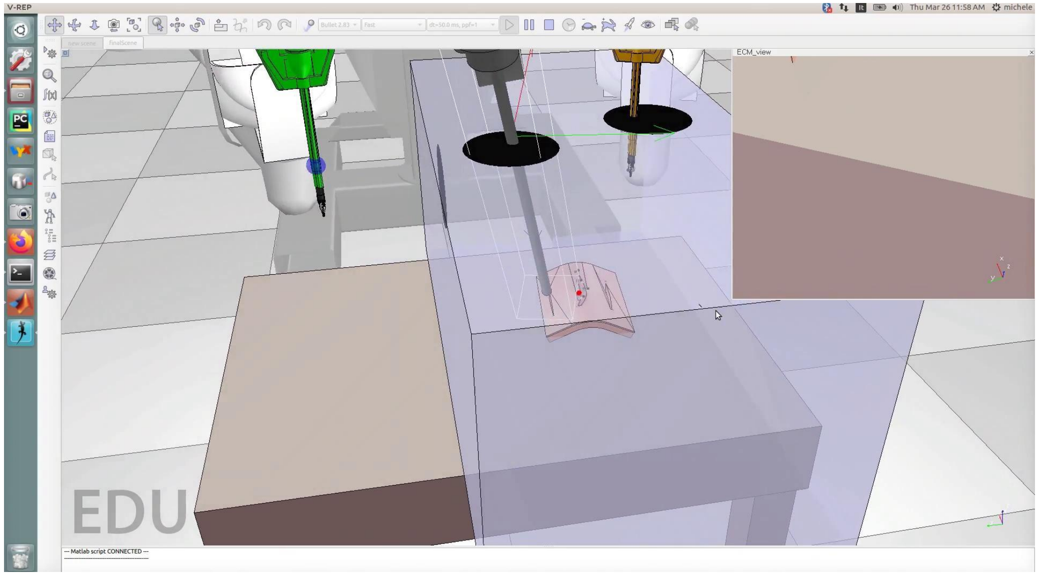
1. **Feature extraction**: extract essence of an image of millions of pixels per seconds.

2. **Autonomous ECM movements**: follow the surgical tip keeping distance or at the center of image.

3. **Take advantage of both tips**

Extra future work and conclusion

Actually we already opened some doors



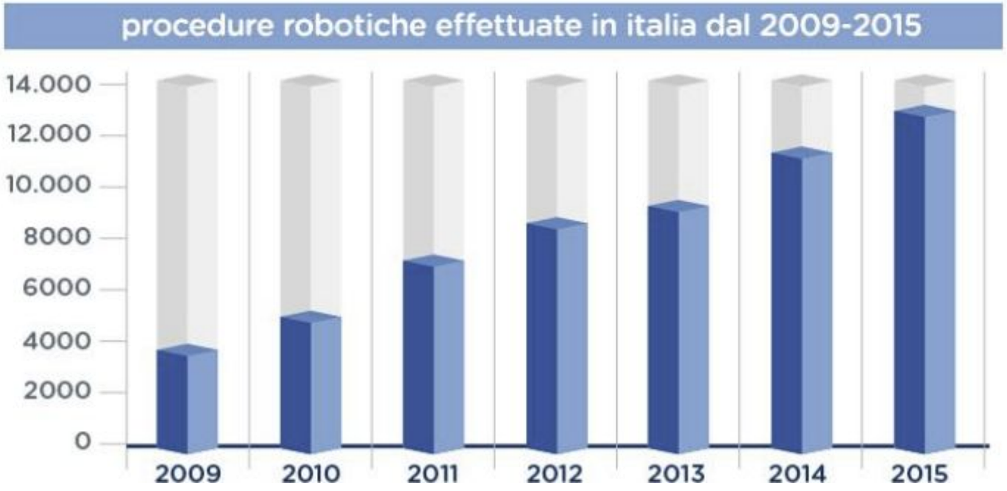
— Matlab script CONNECTED —

ECM control

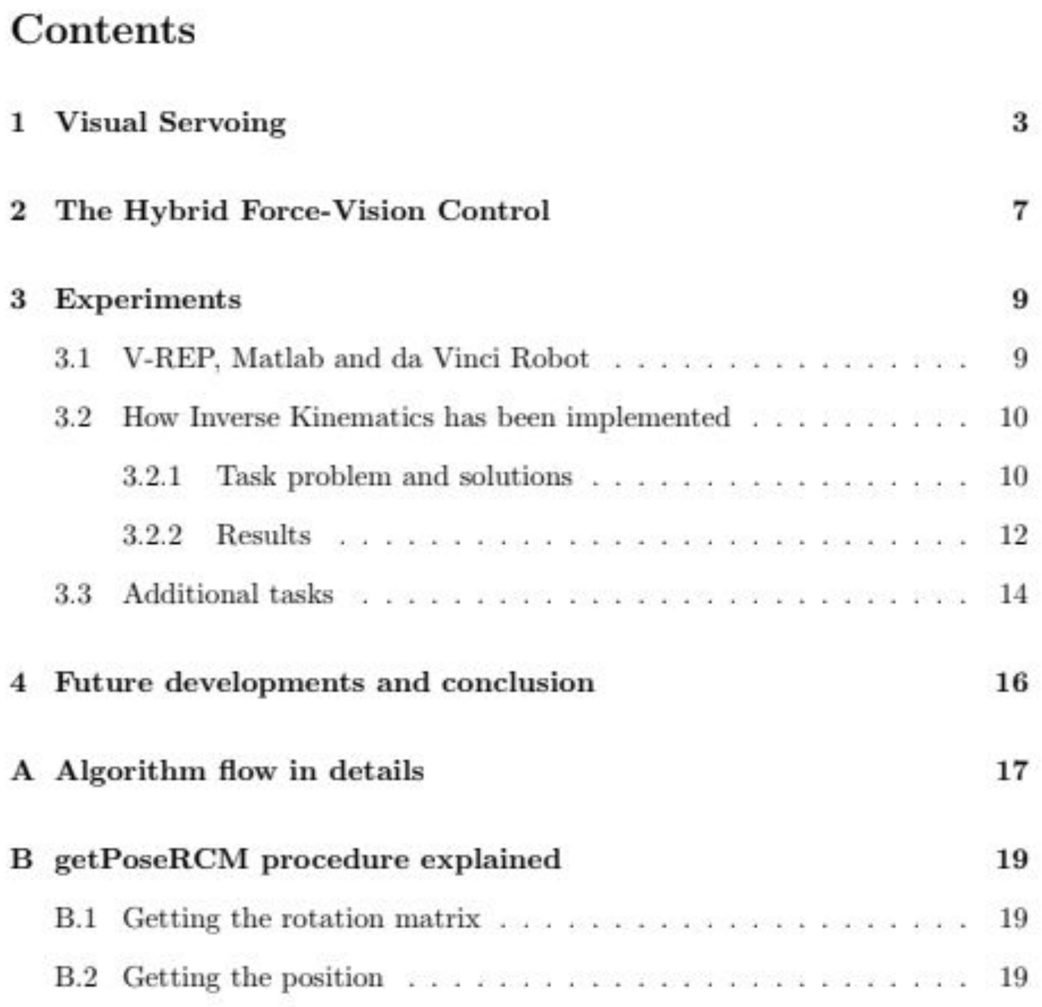
Grasp objects

Conclusion

- ❑ Save human lives
  - ❑ **Telesurgery**: Lindbergh operation 2001
  - ❑ **Telediagnosis**: remote “Remote Presence” Robot RP-6
- ❑ High technology helps doctors
- ❑ Although advanced today, it must improve via research
  - ❑ Artificial Intelligence
  - ❑ **Nanotechnology**



da Vinci
SURGICAL SYSTEM



Contents