University of Burgundy

Masters in Computer Vision

Visual Servoing

IBVS & PBVS

Free Camera 3D Pose Control

by

Tsagkatakis Ioannis

Under the supervision of

Dr. Erol Ozgur

January 14, 2019

1 Introduction

We want to move a camera from its current Cartesian pose to a desired Cartesian pose (see Fig. 1). We assume that we can measure the Cartesian pose of the camera at every instant of time using a sensor.

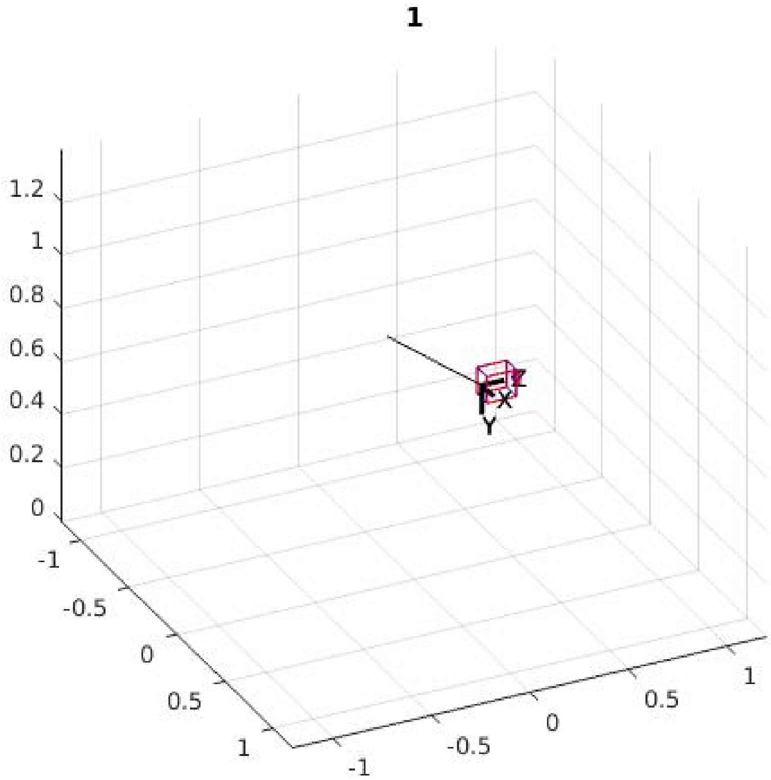


Figure 1: Camera pose control. Blue camera shows the initial pose of the camera, and the red camera shows the desired pose of the camera.

2 The Pose Controll problem

As the location of the camera is always known we have a Position Based Visual Servoing (PBVS) problem. The control system can be either a real system like a robot or a virtual system like in Viruar Reality applications.

The general structure of a PBVS is shown in Figure 3. A PBVS system operates in Cartesian space and allows the direct and natural specification of the desired relative trajectories in the Cartesian space. The parameters

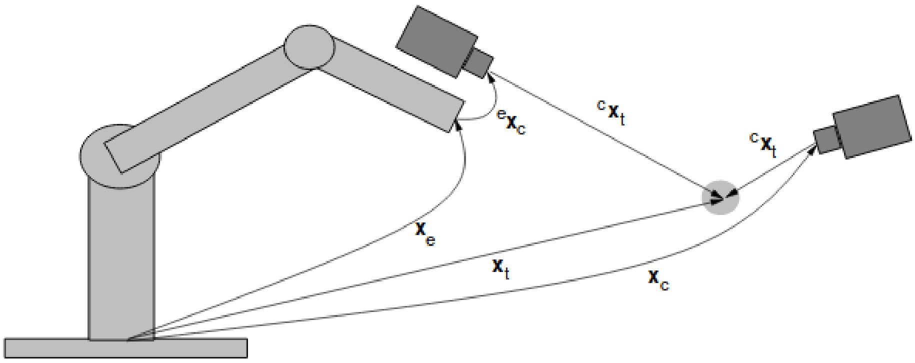


Figure 2: The coordinate frames : word, end-effector, camera and target.

extracted from the image $\mathbf{s} = \mathbf{s}(\mathbf{p}_t)$, are used with the models of camera and object geometry to estimate the relative pose vector $\widehat{\mathbf{W}}$ of the object with respect to the end-effector. The estimated pose is compared with the desired relative pose $\mathbf{W}_d$ to calculate the relative pose error $\widetilde{\mathbf{W}}$. The coordinate frames involved in the process is given in Figure 2 on page 3.

A Cartesian control law reduces the relative pose error, and the Cartesian control command transformed to the joint-level commands for the joint-servo loops by appropriate kinematic transformations. By separating the pose-estimation problem from the control-design problem, the control designer can take advantage of well-established robot Cartesian control algorithms.

2.1 The Control law

The visual features parameters extracted from the image are a function of the camera poses as given by

$$\mathbf{s} = \mathbf{s}(\mathbf{p}_t) \tag{1}$$

By taking the derivative of the above relation we obtain

$$\dot{\mathbf{S}} = \mathbf{L}_s \mathbf{V}\,, \tag{2}$$

where $\mathbf{L}_s$ the so called *Interaction matrix* or *feature Jacobian* and $\mathbf{V}$ the camera (Kinematic screw) denoted by $\mathbf{V} = (\vec{v}, \vec{\omega})$. Thus thw $\mathbf{V}$ contains 3 translatiosn and 3 rotations.

The goal of the control law is to minimize the error given by

$$\mathbf{e}(t) = \mathbf{S}\left(\mathbf{p}_t\right) - \mathbf{S}^*\,, \tag{3}$$

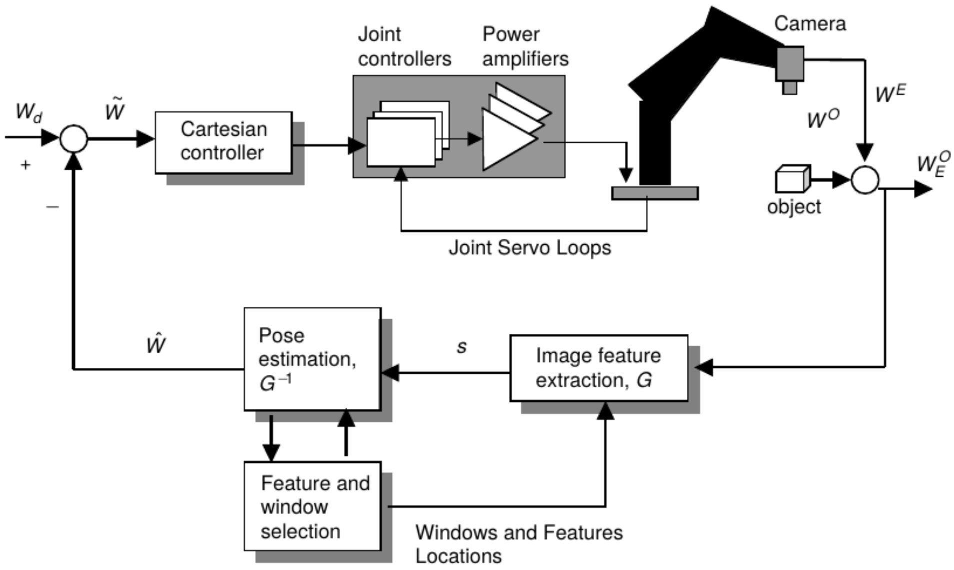


Figure 3: Structure of Position based Visual Servoing (PBVS).

where $\mathbf{S}^*$ the desired value of the feature.

Using equations (1) and (3) then the time variation of the error is

$$\dot{\mathbf{e}} = \mathbf{L}_s \mathbf{V} \tag{4}$$

A good control law is to have an exponential decoupled decrease of the error $\dot{\mathbf{e}} = -\lambda\ \mathbf{e}$. This corresponds to

$$\mathbf{V} = -\lambda \widehat{\mathbf{L}_s^+} \left(\mathbf{S} - \mathbf{S}^*\right) \tag{5}$$

2.1.1 Stability analysis

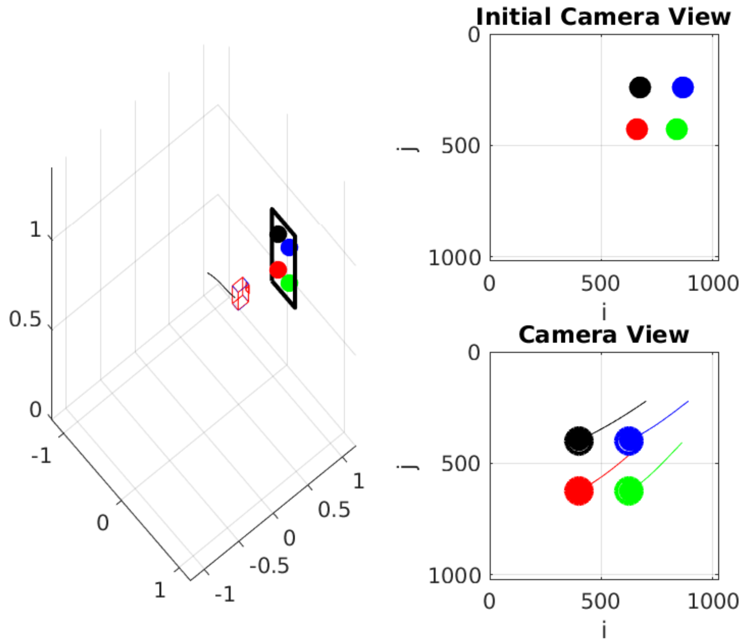
$$\begin{cases} \widehat{\mathbf{L}_s}\widehat{\mathbf{L}_s^+} = \mathbf{I} & \text{Ideal behavior} \\ \widehat{\mathbf{L}_s}\widehat{\mathbf{L}_s^+} > 0 & \text{The error } \mathbf{e} \text{ decreases} \\ \widehat{\mathbf{L}_s}\widehat{\mathbf{L}_s^+} < 0 & \text{The error } \mathbf{e} \text{ grows} \end{cases} \tag{6}$$

3 Image Based Visual Servoing

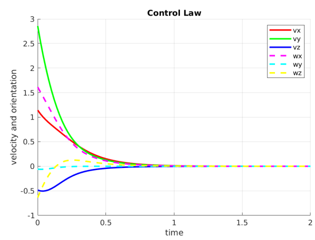
3.1 Matlab Experiments

3.1.1 A Loop of IBVS visual servoing

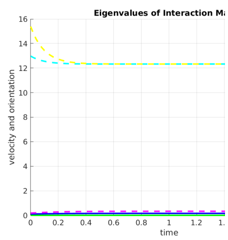
In the first example we have 2 cameras, and the pattern is clearly visible from them. From the simulation results on Figure 4 on page 5, we observe



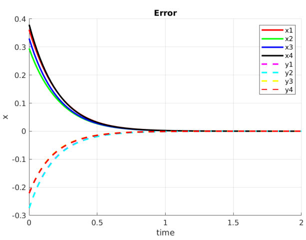
(a) The camera motion



(b) Control Law



(c) Eignevalues



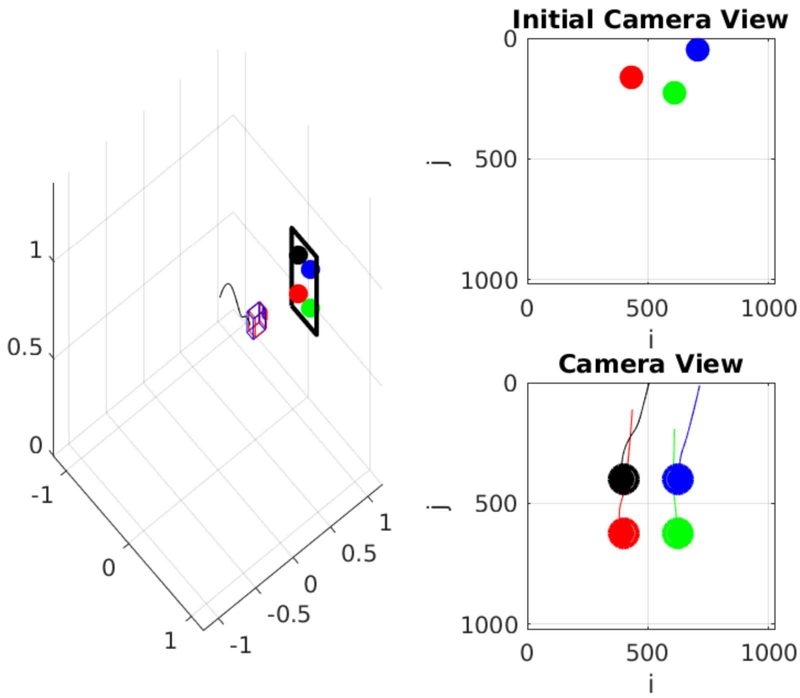
(d) Errors

Figure 4: Simple case : Observable pattern, the control law.

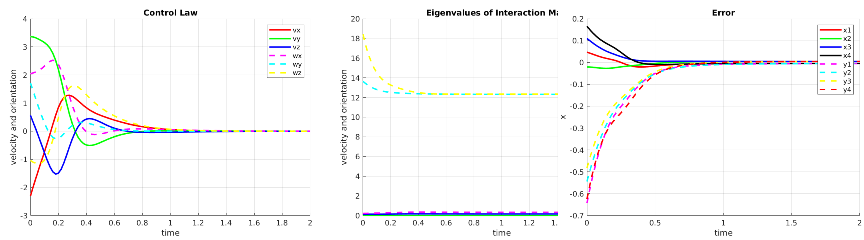
that the camera follows a very good and straight path. The velocity is getting smaller and smaller as we aproaching the target. We notice that the interactiom matrix have some very high eigenvalues and some very small ones. So the DoF of the eigenvalues with big values is the only ones that drives the control of the system.

3.1.2 Loosing Features: 3 points

If we have only 3 points then we get 6 features witch is the minimum required. That not gurantes that we get a good solution in every case. Here the final camera location is almost correct. The movement have some



(a) The camera motion



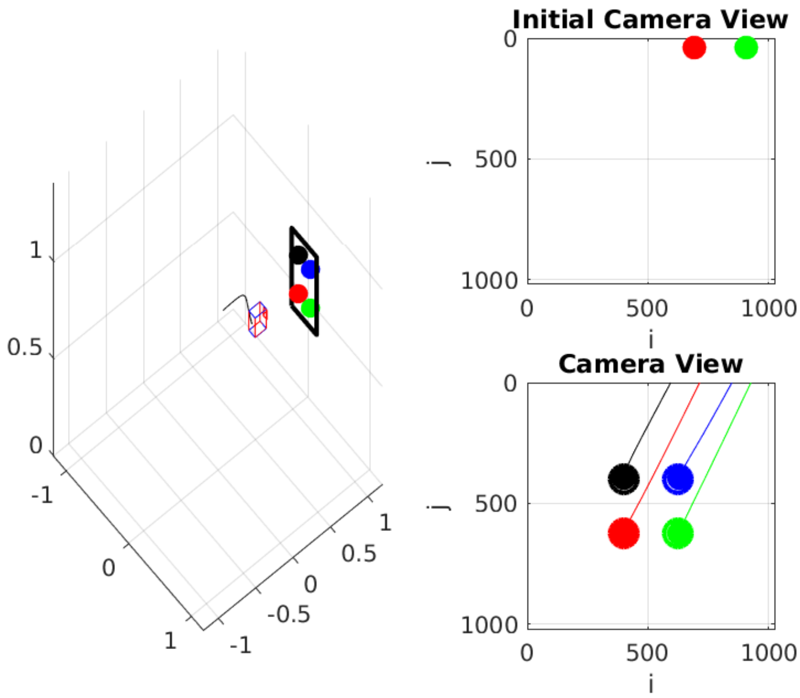
(b) Control Law (c) Eignevalues (d) Errors

Figure 5: Using 3 features : Observable pattern, the control law.

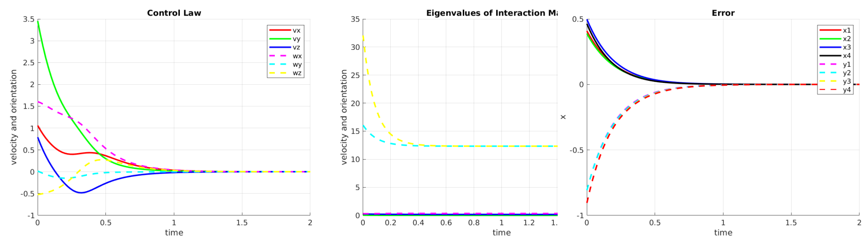
instabilities, but that is expected due to the poorless interaction matrix. The simulation results is given at Figure 5 on page 6.

3.1.3 Loosing Features: 2 points

By using only 2 point we get better results. That was not expected, but thatis because of the initial view have 2 lines that is directed connected with the target lines it was easy to get a very smooth solution. The simulation results is given at Figure 6 on page 7.



(a) The camera motion

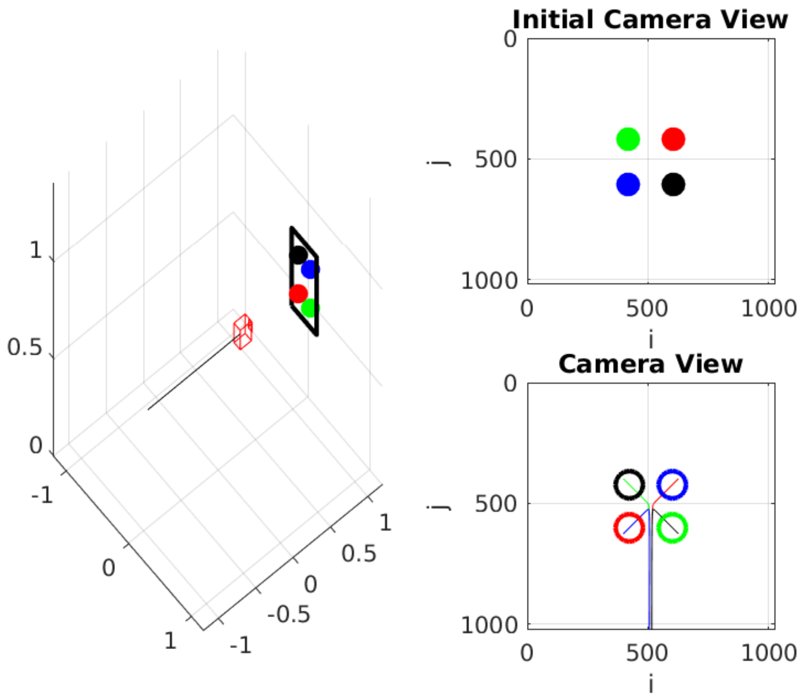


(b) Control Law (c) Eignevalues (d) Errors

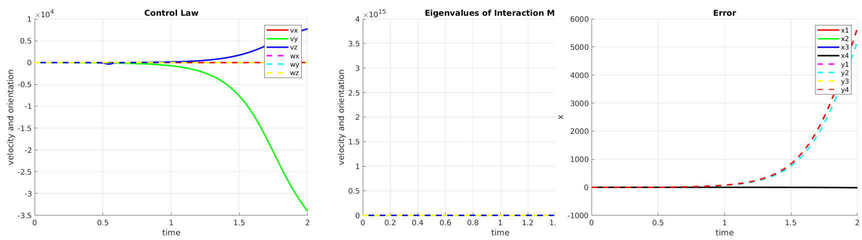
Figure 6: Using 2 features : Observable pattern, the control law.

3.1.4 The Coplanar Problem

Here the human solution is just a rotation. But the solution to minimize the distances is to move the camera backwards as the error is decreasing. All the eignvalues are small in this case. The simulation results is given at Figure 7 on page 8.



(a) The camera motion



(b) Control Law (c) Eignevalues (d) Errors

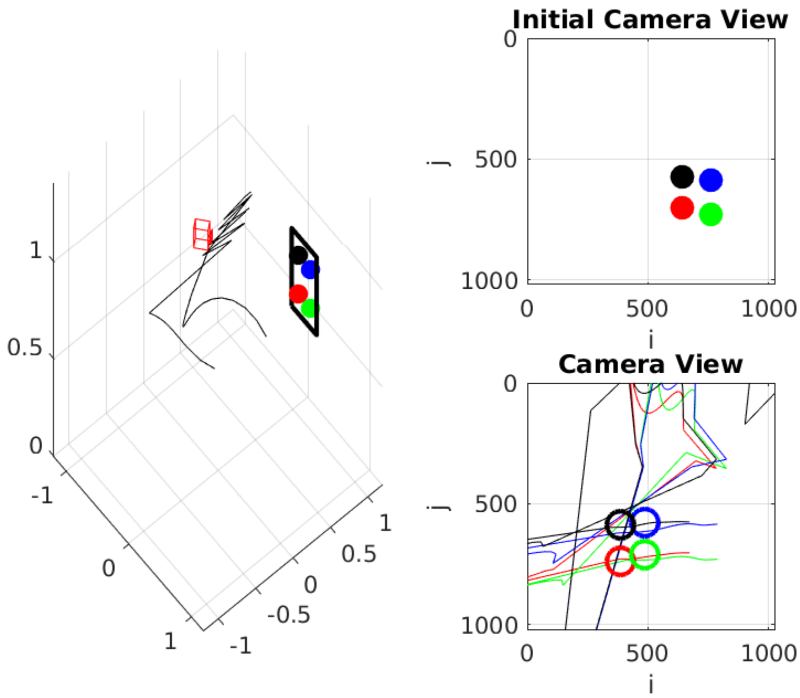
Figure 7: The coplanar problem.

3.1.5 The Local Minimuma Problem

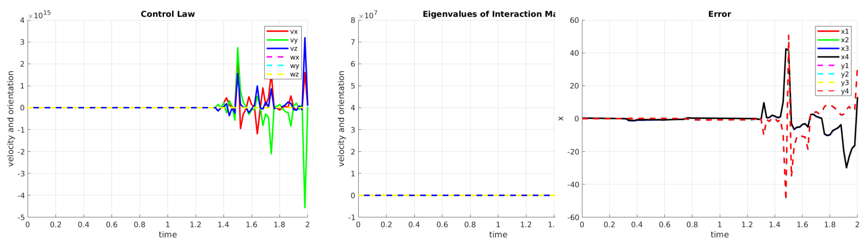
The simulation runs into a local minima and it was unable to find a smooth path or a valid solution. The simulation results is given at Figure 8 on page 9. The motion is shown in Figure 9 on page 9.

3.1.6 The Diverge Problem

The simulation results is given at Figure 10 on page 10. The motion is shown in Figure 11 on page 10.

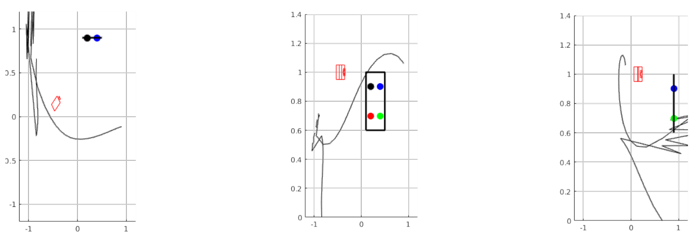


(a) The camera motion



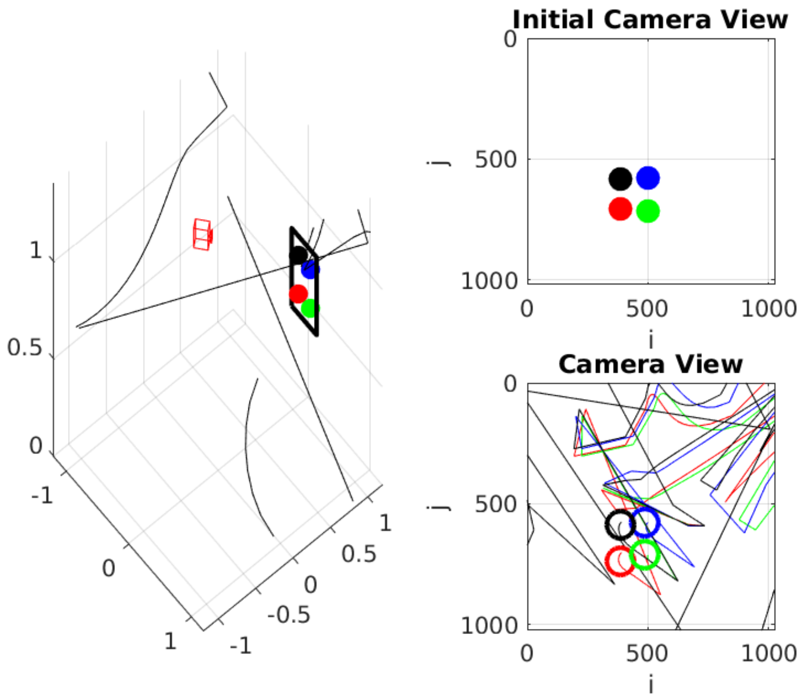
(b) Control Law (c) Eignevalues (d) Errors

Figure 8: The local minima problem.

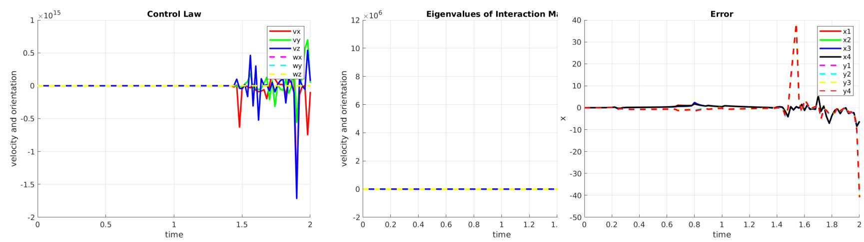


(a) XY view (b) XZ view (c) YZ View

Figure 9: The local minima problem: Motion Views

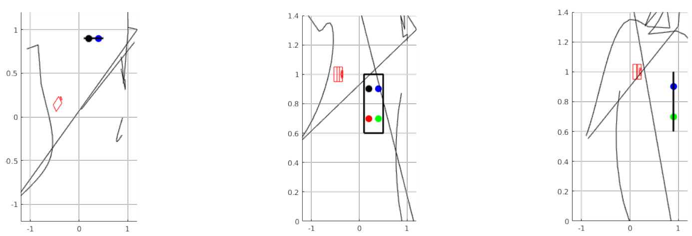


(a) The camera motion



(b) Control Law (c) Eignevalues (d) Errors

Figure 10: The diverge problem.



(a) XY view (b) XZ view (c) YZ View

Figure 11: The diverge problem: Motion Views

4 Position Based Visual Servoing

Here the camera pose at each time step is known. The movement of the camera is very smooth and it reach the desired location without problems. The simulation results is given at Figure 12 on page 11.

5 Conclusions

5.1 MatlabCode

The code of this project is on Github repository `https://github.com/jtsagata/VisualServoingLab`.

6 Bibliography

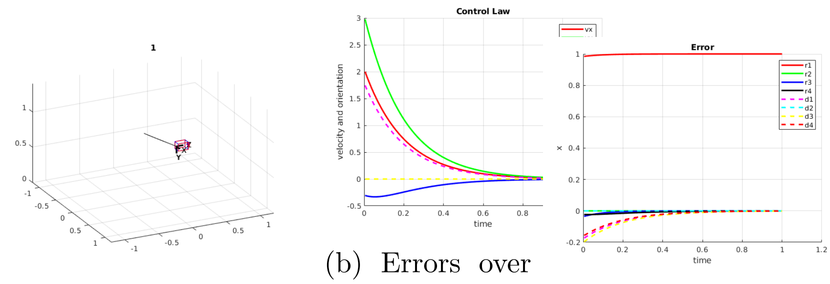
•Visual servoing is an effective technique to control robots: Vision provide a rich information about the environment Developed to deal with dynamic environment Can applied to several kind of robots and in several fields

The code of this project is on Github repository `https://github.com/jtsagata/....`

references Visual Servoing : Theory and applications (Chapetr 15 from some book)

F. Chaumette, S. Hutchinson. Visual Servo Control, Part I: Basic Approaches. IEEE Robotics and Automation Magazine, 13(4):82-90, December 2006.

S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. IEEE Trans. Robot. Automat., 12(5):651—670, Oct. 1996.



(a) Camera movement (b) Errors over time (c) Camera control law

Figure 12: Position Based Visual Servoing

F. Chaumette, S. Hutchinson. Visual Servo Control, Part II: Advanced Approaches. IEEE Robotics and Automation Magazine, 14(1):109-118, March 2007