Delivery 4

Física orientada a la Modelització i Animació Realista  
**Automatic computation of the jacobian.   
Solving the inverse kinematic problem by jacobian pseudo-inversion.**

Júlia Gasull i Navarro, 4964322**9**Y

Junjie Ji Chen, 512880**9**0E

Aarón Acosta Fernández 4924430**9**K

# 

# Index

[**Index**](#_uztcx5x9ah75) **2**

[**Delivery**](#_mhasi8q67ttu) **3**

[Exercise 1](#_skx7ejb7n1un) 4

[Exercise 2 & 3](#_nn99skw95lat) 5

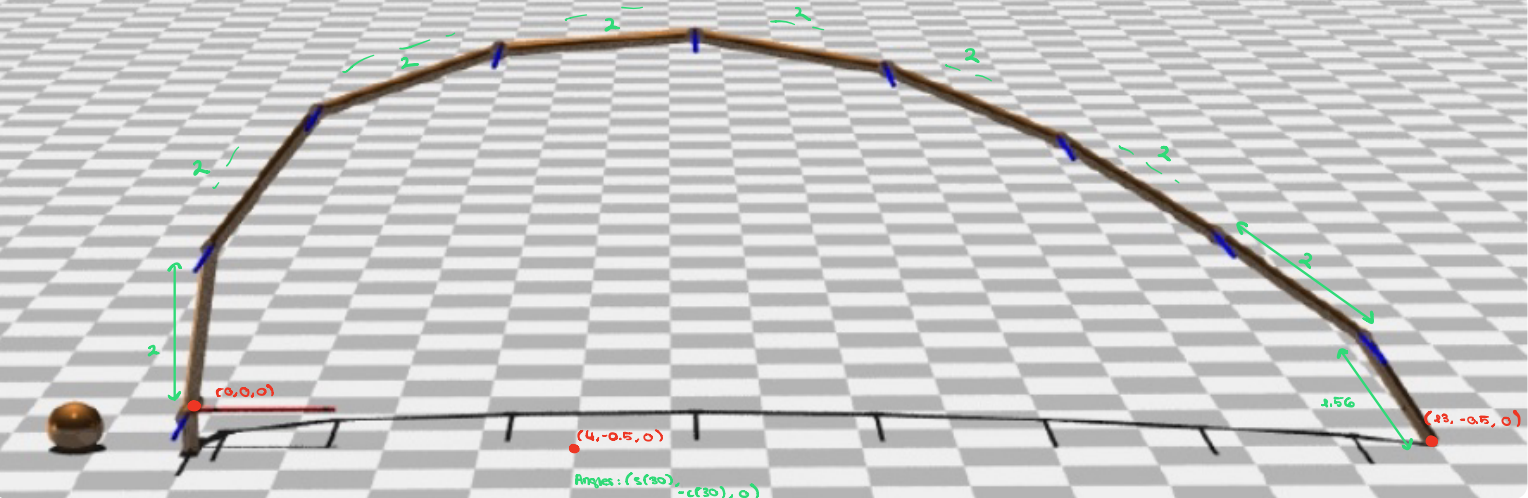
[Exercise 4](#_tehnjs8wmjww) 6

[Exercise 5](#_va0uot74259i) 6

# Delivery

The aim of this exercise is to obtain the jacobian of a system directly from its Denavit-Hartenberg table and to find the evolution of the system's internal variables by means of the pseudo-inversion technique of the jacobian, allowing a complex system to execute a prescribed movement.

The system under consideration is the one depicted in the figure:



You have the scene described in the jcb.pov file, and it’s Denavit\_Hartenberg table is contained in the taula-DH file.

In this file the values of the joint angles are the ones represented in the figure, in which the position of the clamp (far edge; "pinça"), described using the Base system, is (x=13, y=-0.5, z=0), and the direction of the clamp element is described by the unitary vector: (sin 30º,-cos 30º, 0) .

The exercise consists in generating an animated sequence where the arm clamp moves at constant speed from the starting position (x=13, y=-0.5, z=0) towards the final position (x=4, y=-0.5, z=0), while the last element keeps maintaining a fixed orientation.

The length of all elements is L=2, except for the last one which is L\_pinça=L\_clamp=1.56.

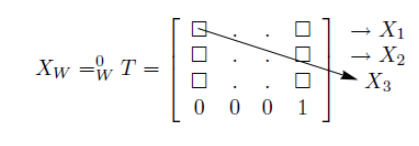
In order to complete the exercise, you must solve the following points:

## 

## Exercise 1

**Specify a procedure, based on the homogeneous matrices between consecutive reference frames and its derivatives, which allows finding the Jacobian for the system considered. In particular, you must identify three elements from the homogeneous matrix 0\_T\_clamp useful for the purpose of imposing the said conditions onto the trajectory. You can find some help in the jacobian\_calculation.pdf notes.**

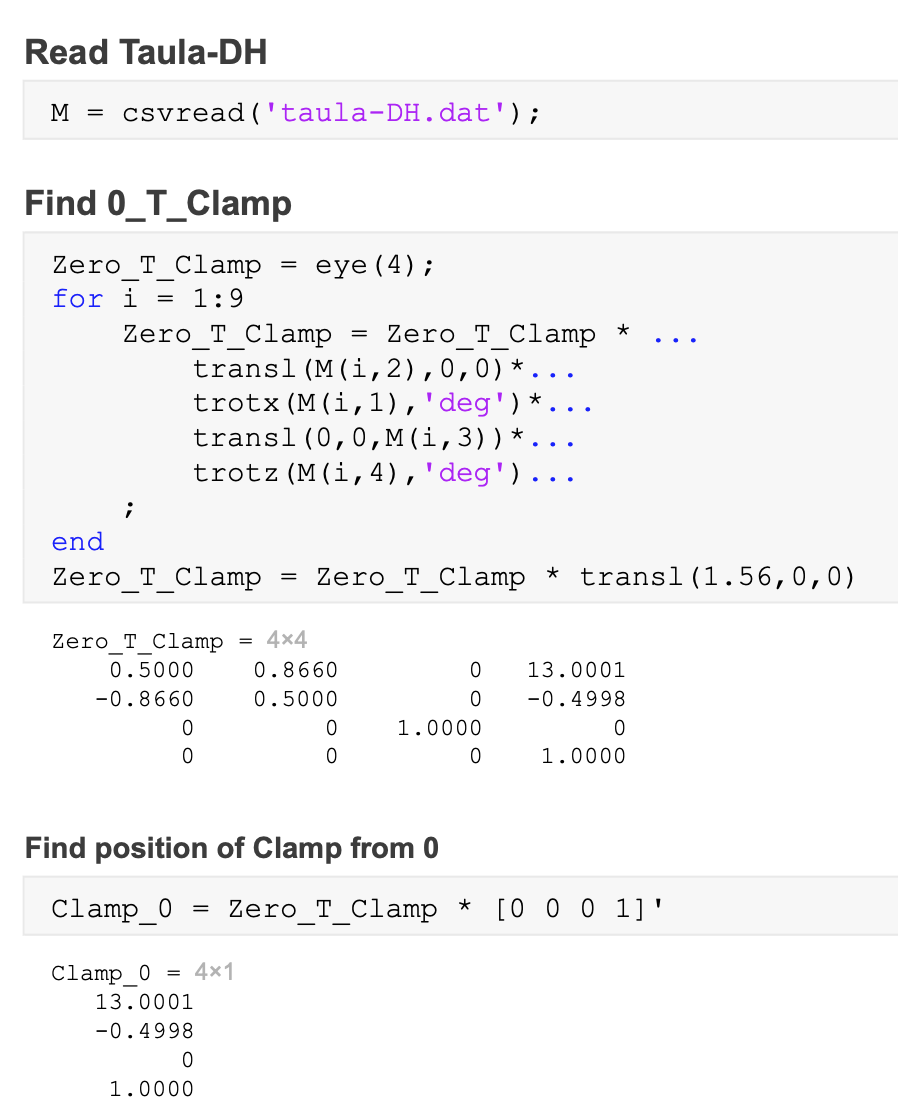
1. Identify the M external variables (X1, X2, X3), in the homogeneous matrix:



1. Build the matrices T and D for each line of the D-H table
2. For each joint variable j, we compute the matrix D\*X, which is equal to the J matrix.



Our results testing if the Clamp is at the right point (13, -0.5, 0):



## Exercise 2 & 3

**Write a code of your own which ...**

1. **… reads the file containing the DH table and computes the Jacobian matrix (3x9) for the values of the joint variables specified in the DH table.**

**Note that you have the results for the starting position in the "homogenous\_matrices\_checks" directory, so that you can be sure that your code produces the expected results,**

**and ...**

1. **… it solves the inverse kinematic problem by means of the pseudo-inversion method.**

**In other words, you must obtain the increments of the 9 internal angular variables that give rise to the variations of the external variables specified by the desired trajectory (for example, you can divide the sequence in 90 steps; then at each step your code will take care of updating the 9 internal angular variables and dump them into the jcb.pov file).**

**Again you have test-results for the starting position in the jacobian\_checks directory.**

**In particular, the file complete\_jacobian\_results\_first\_step.dat shows the results you should obtain for the jacobian in the starting configuration, and in the file first\_step.dat you have the values for the change of the internal degrees of freedom that you should obtain in the first time step.**

**Include in your program a do-loop inside which the full calculation of points 2 takes place, updating the data contained in the jcb.pov file, and issues a system command to the operating system which renders the image with pov-ray and renames it with a tag number, until the whole trajectory has been completed. Then create a sequence showing the movement.**

Video: [lab4-ex3.gif](https://drive.google.com/file/d/1gkANmzQiFFd2DEu6z5ys3ZRhD4YIqFpO/view?usp=sharing)

Code: [lab4-code3.pdf](https://drive.google.com/file/d/1fUr-Siv8ab2QsQ00vb8SBBilGfIhnVna/view?usp=sharing)

## 

## Exercise 4

**Introduce the required changes in the algorithms so that the trajectory is subject to the additional constraints (only one of them applies at the same time):**

1. **The angle joint #5 is frozen until the remaining joints compensate for this effect, so that the movement of the clamp remains unchanged its initial value at all times (). Hint: right after the jacobian matrix has been obtained, try zero-ising in your code any one of the 9 columns of the jacobian, and let the rest of your code proceed intact (note that all following calculations will provide different results than in point 3).**

Video: [lab4-ex4a.gif](https://drive.google.com/file/d/1Xk_EWlEINdq7ebveuJJTQVVULfF1ke9O/view?usp=sharing)

Code a\*: [lab4-code4a.pdf](https://drive.google.com/file/d/1blB0cKBc5SXecjrXbXCQyt-68_EcHVfa/view?usp=sharing)

1. **Based on the last digit “d” of your DNI document greater or equal to 5** *(all of us have a 9 as last digit greater than 5)***: modify the algorithms so that the trajectory of the arm clamp moves at non-constant speed from the starting position (x=13, y=-0.5, z=0) towards the final position (x=9, y=-0.5, z=0) , while the last element keeps maintaining a fixed orientation. The intermediate x positions should be:**

**where i = 0,...,Nf is the frame number, Nf is the number of frames you use, and d is the digit coming from your DNI number.**

Video: [lab4-ex4b.gif](https://drive.google.com/file/d/1L1ZLZsqUt8M2A-S3VkDeKkvOAt2iYsOj/view?usp=sharing)

Code b\*: [lab4-code4b.pdf](https://drive.google.com/file/d/1PedSYUkjmfLiNEtD83cQvRay1WusSHt5/view?usp=sharing)

|  |  |
| --- | --- |
| **🗒 Note\*** | *The code for exercise 4 is a duplicate of the code for exercise 3, but with some modifications added.* |

## Exercise 5

**Write a report, including your name and DNI number, of points #1 and #4, and deliver it together with your codes and animations of points 2, 3 and 4.**

In the delivery.