EPFL REHAssist

Robotics Practicals 2025 – Lab Worksheet

The MicroDelta

Teaching Assistant
Pierre Oppliger

Collaborators
Nicolas Péteut
Emilie Grandjean
Nicolas Furrer
Sara Heidaripour

Lab's professor
Dr. Mohamed Bouri

In case of questions, absences, or special cases, please write to
pierre.oppliger@epfl.ch

Lab description

Introduction

A Delta robot is a type of parallel robot known for its remarkable speed and precision, invented at the EPFL in 1989.
Designed with three arms connected to universal joints at the base, it is widely used in high-speed automation applications, particularly in industries like packaging, food processing, electronics, and pharmaceuticals. The robot's design allows for rapid movements and quick adjustments, making it ideal for tasks like sorting, picking, and placing small items with great accuracy.

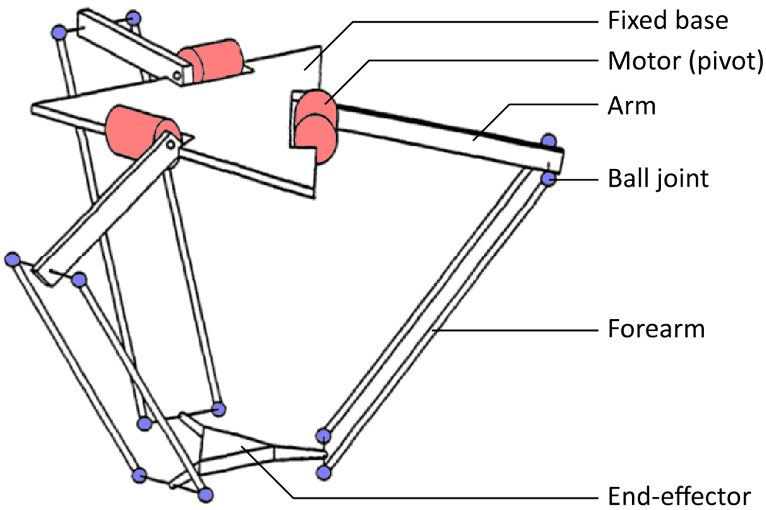

Figure 1: Simplified drawing of the Delta robot

The objective of this lab is to have an applied introduction to parallel robotics using a micro delta robot.

Structure

This lab can be divided into three parts:

- Theoretical questions about robotics, more particularly about delta robots
- Practical use of a micro delta robot
- Simulation using Simulink

Each part is (relatively) independent. We advise no specific order except that you should use the lab session to complete all the experiments requiring the delta robot themselves.

Assessment

This lab is a graded component, and the resulting grade will contribute towards the academic credits for the course, provided you are enrolled in MICRO 453.

The following guidelines apply to the assessment for the lab component of the MICRO 453 course:

- Each group member will receive the same grade; however, full participation is mandatory. Absences must be communicated in advance and appropriately justified. Non-participation will result in penalties for the individual concerned.
- The assessment (for enrolled students only) will be based on a report, which will address the questions outlined in this present document. This report should be submitted as a single PDF file. Ensure that all group members' names, SCIPER numbers, and the code of your delta robot (found on the sticker atop the robot) are noted. It is crucial to include the robot code, as slight variations in behaviour are already programmed into each robot.
  The PDF report must be uploaded to Moodle by the specified deadline.
- Strict adherence to academic integrity is required. Plagiarism and cheating will not be tolerated. Properly cite all sources used, and be aware that anti-plagiarism software will be employed to detect any dishonest behaviour. Any instances of cheating will be met with severe consequences.
- All plots must be based on actual measurements. If any plots are found to be inaccurate, and no alternative data is available, clearly state this and provide an explanation. Do not attempt to generate artificial plots using AI or any other means.
- Ensure that the values you use for plotting and analysis are derived from your assigned robot. Using data from another robot may be perceived as cheating, as while all robots share similar behaviour, there are subtle differences among them.
- When answering a question, an exercise, or similar tasks, please explicitly state in your report what you are responding to. We will not infer your report structure based on your answers.
- Don’t write your report during the lab, as you might otherwise run short on time. Take notes instead and do the writing at home.

Lab’s delta robot

This robot has been entirely developed in-house by the lab to highlight the key components and features of the delta robot.
Please handle it carefully, as there is no tech support or auto-fix.

Electronic Hardware Architecture

The Delta robot electronics is divided into four parts:

- *CANable module*: The communication protocol used in this robot is the CAN bus (Controller Area Network). This protocol is heavily used in the car industry due to its robustness and reliability. The CANable module is used as a USB-CAN converter.
- *Development board:* This ESC (Electronic Speed Controller), model B-G431B-ESC1*,* is offered as a development kit. It is used to drive the brushless motors of the delta. It contains an STM32G431CB microcontroller and is accompanied by a daughterboard using an ST-Link for programming the board. The datasheet is available on Moodle.
- *Three-axis Motherboard*: This custom PCB is used as a motherboard for the different motor drivers and the different electronic components needed to connect everything. The schematics are available on Moodle.
- *Motor/encoder*: The brushless motor, model 4221G048BXTH from Faulhaber, is a BLDC motor with external rotor technology. The motor is coupled to a magnetic encoder, model IEF3-4096L. The datasheet is available on Moodle.

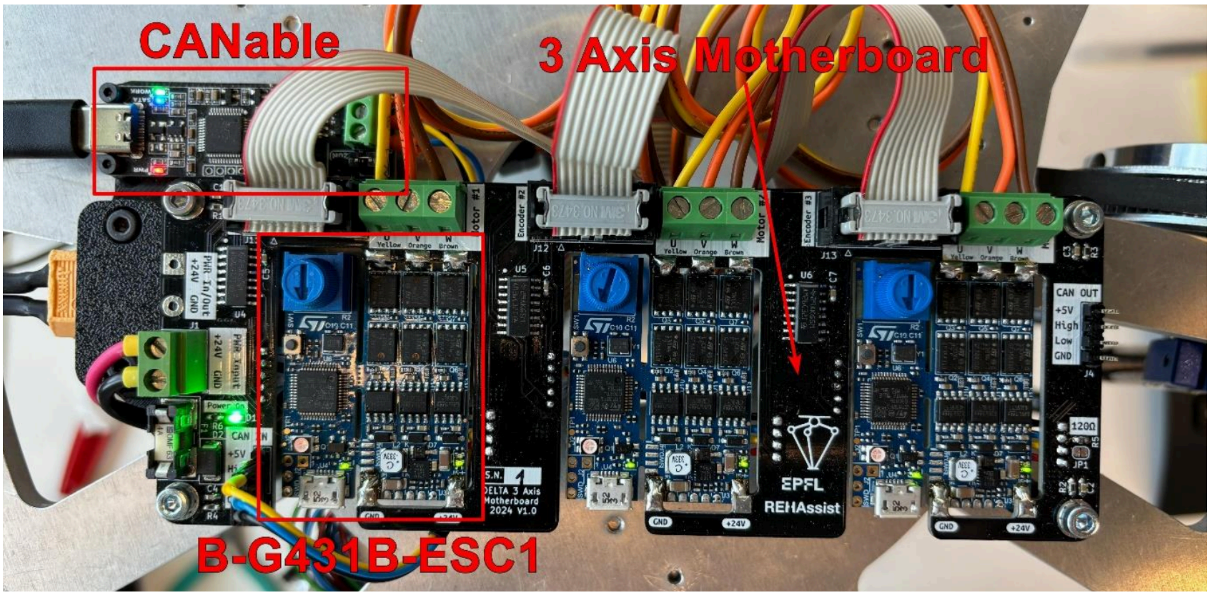

Figure 2: Electronic hardware

Physical Structure of the Micro Delta

Consider the angular Micro Delta robot, as developed for the robotics practicals and represented by Figure 3. A delta robot kinematics chain with three motors in rotation consists of three arms attached to a fixed base and a mobile end-effector. Each arm is connected to the end-effector through a virtual spherical joint and to the base via rotational motors at their respective pivot points. The motors control the angles of the arms, allowing the end effector to move within a defined workspace. This setup allows the delta robot to achieve precise and fast movements, commonly used in applications like high-speed picking and assembly.

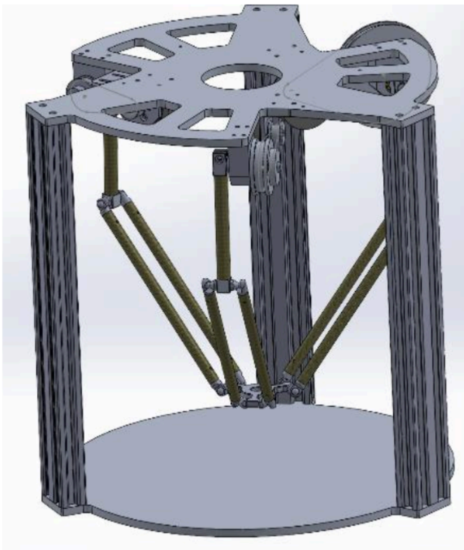

Figure 3: Angular Delta robot with three motors. Axis rotational angles θ_1, θ_2, θ_3 are equal to 0° in the legal horizontal plane and 90° in the legal vertical plane.

You might need to consider the following parameters (careful with the units):

- Arms made of fibre carbon tubes
  - Moment of inertia of the arm segment according to the joint axis: J_a = 48 g · cm^2
  - Mass of the arm segment: m_a = 2.25 g
  - Mass of the elbow, as composed of the spherical balls: m_e = 6 g
  - Length of the forearms is: l_{fa} = 160 mm
  - Length of the arm is: l_f = 80 mm
  - The mass of each carbon tube of the forearm is: m_{fa} = 10 g
  - Mass of the end effector: m_{ef} = 15 g
- The three rotational axes
  - Motor Flat 42
    - Series BXT4221, from Faulhaber
    - Nominal voltage 48 V
  - Reduction transmission
    - Motor pulley diameter: d_{pm} = 12 mm
    - Motor pulley inertia: J_{pm} = 1.2 g · cm^2
    - Output pulley diameter: d_{pa} = 72 mm
    - Output pulley inertia J_{po} = 1135 g · cm^2
    - Efficiency assumed as: 100 %

You can find additional technical information on Moodle as well as the specifications of the main subcomponents.

Software

The robot operates autonomously, but it can be controlled through dedicated software that provides instructions. Once installed, the software is simply an executable that, when run, provides an interface to commands to the robot. The robot functions independently but is being guided by the computer.
You can find a ready-to-use version for Windows 10/11 of this software on Moodle. Please use it in the practicals' room and not on your personal computer. We cannot guarantee compatibility with other OS versions.

Theoretical questions

Here is a list of questions and small exercises to help you better understand this lab. You might answer them before or after your lab session.
Your answer should be in your report. For every question, explain your reasoning, provide numerical values when applicable using data for this lab's Delta robot, and justify every approximation. For example, if we ask the size of the robot arm, we are, of course, talking about this lab's Delta robot.
If you need the numerical value of a point for the next step but are unable to find an exact answer, attempt to approximate its value and justify your reasoning.

Parallel robots VS serial robots

Please look at the numerical values given above and/or check the datasheets on Moodle.

1. In general, what are the advantages and the disadvantages of parallel robots compared to serial robots?
2. Is the MicroDelta robot used in this lab a parallel robot or a serial robot? Justify your answer.
3. Give examples of hybrid robots (semi-parallel or semi-serial robots). If this lab's Delta robot is not already a hybrid robot, propose how it could become one. If it is one, propose how it could stop being one.
4. How many degrees of freedom are there at the end-effector of our Delta robot?
5. We add a gripper (on / off) on the end-effector. How many degrees of freedom will the picked object have?
6. Consider one of the motors described above and its corresponding reduction on an arm where the end-effector is connected to a single arm. (No other arms) What is the moment of inertia concerning the still-attached motor when the forearm is aligned with the arm? (You are welcome to make the approximations and assumptions you want to simplify the computation.)
7. Same question, but this time on the robot, there is an unused motor with its own reduction, electronics, and so on, weighing a total of 950 g at the elbow; a configuration quite similar to the SCARA robot. Compute and comment. What can you tell about the differences between Delta robots and SCARA robots?
8. Let's suppose we can have a human-like 6 DOF robotic arm for the same price. Explain why, for pick-and-place tasks, the Delta robot is still preferable.

Delta robots VS humans:

Please look at the numerical values given above and/or check the datasheets on Moodle.

1. If the rotation speed of a motor is 8000 min^{-1}, what is the angular speed of the arm and the linear speed of the tip of the arm?
2. What current do we need to apply to the motor to make the motor produce a torque of 100 mNm?

3. Supposing only one arm is connected to the end-effector (the other arms are disconnected), supposing that the forearm is aligned with the arm, and supposing a current of 3 A is applied to the motor, what is the force F_t applied to the end-effector by this motor?
4. The exact dynamics linking the motors' positions, speed, and acceleration are very complex and will be seen in the simulation part. We will naively assume that 50% of the power coming from the power board ends up as displacement on the end-effector. What acceleration, a_t, does a 40 g biscuit reach with a current burst in the motors of 3 A during 1 second?
5. List what has been approximated in finding a_t the previous point. (Do not compute anything to answer this question.) Why would it not be possible to give a current burst for 10 seconds?
6. Explain why Delta robots are particularly good constructs for reaching high acceleration. Why is a 5 G acceleration perfectly conceivable for a Delta robot? Compare it qualitatively to a human's precision of their hand starting from the shoulder. (Tip to answer: serial vs parallel)
7. Consider the following packaging sequence done by the robot 24h/24, 7d/7:
    a. A biscuit is picked from a conveyor belt by the robot in 20 ms
    b. This biscuit is transported to an empty packaging with an acceleration of a_t followed by a deceleration of a_t on 50 cm.
    c. This biscuit is being dropped in the packaging in 100 ms.
    d. The robot goes back to the conveyor belt. This movement is two times faster than the duration seen in point b).
    e. The robot’s motors cool down as the tracking vision system waits for the next biscuit to be picked. This takes 150 ms.

    And, consider a human employee doing these very repetitive tasks in a high-volume factory working from 7 am to 8 pm every day, 6 days a week, 30 minutes break at lunch, 10 minutes break every hour, 2 minutes break every 15 min. of work, to pick-and-place a biscuit takes an average time of 45 sec. (This movement is very fast and easily done at the beginning of the day, and tends to be very slow at the end of the day.) How many human employees can the work be lightened by a single robot? (Compute time taken at every step)

8. (Bonus question) Consider a human employee who is paid 3.20 CHF per hour (work is outsourced outside Switzerland) and costs 40 CHF to be trained, and a robot costs 0.08 CHF per hour to operate (including electricity, spare parts, maintenance every month, etc.) but costs 12’000 CHF to be installed. What is the quantity of biscuits in kg to make the robot a better choice?

Delta robot kinematics

1. Why is it much harder to compute the forward kinematics equations for a Delta than for a robotic arm? (Give examples) What property behind the beauty of the Delta robot is an obstacle here?
2. Redemonstrate the equations for the inverse kinematics for this particular Delta robot. If you are missing some geometric parameters, measure them during the lab,

or if you are doing this part after the lab, write us an email. Tips: 1/There is no rotation of the end-effector. 2/The arms move only in three defined planes

3. Give the code of an implementation of the inverse kinematics in Python 3 or Matlab® with the numerical values already substituted in the formulas.

Practicals

Safety notice

Be aware that you are working with a live robot. Don't leave your fingers near the end effector, the arms, or the motor whenever the delta is moving. Using wrong parameters, such as too high PID values or speed/acceleration values, could lead to the robot dislocating its arm, thus leading to flying delta arms. I would strongly recommend orienting the delta robot in such a way that any flying arm can't reach you.

The 3-axis motherboard is a fragile electronic PCB. Be careful not to touch the PCB using anything metallic. If you do have to touch the PCB (for example, to connect the uUSB cable), please first touch the metal body of the delta to discharge yourself from any electrostatic charge. Don't put your finger on the PCB when the robot is operating, as it can have enough electricity running through it to be painful for you.

Finally, in case of any weird behaviour of the Delta or in case of any doubt, immediately activate the ***Emergency stop button*** and choose the ***Transparent mode*** in the GUI. It is always better to stop and restart the robot than to have to repair something.

Installation

Simply go to Moodle and download the folder "*GUI MicroDelta - Windows.zip*". Extract it locally.
You can run the software as any other Windows software by double-clicking the executable.

GUI Explorations

In this part, the Graphical User Interface will be used to control the micro delta robot. The GUI is separated into different sections:

1. Delta Information: This section is separated into four main parts.
    a. Status Robot: You will find the actual values of the robot.
        i. XYZ Values: The X, Y, and Z positions of the end effector of the delta in a Cartesian coordinate system.
        ii. $\theta_1\theta_2\theta_3$ Values: The angles of the three different motors, with θ_n = 0° being at the horizontal.
    b. Infos Trajectory
        i. XYZ Target: The XYZ position target of the end effector
        ii. Send Target: Button used to update the target.
    c. Trajectory Target:
        i. Vmax, Amax, Dmax: The maximum speed, maximum Acceleration, and maximum deceleration used for trajectory planning.
    d. PID Value
        i. Kp, Ki, Kd: The proportional, integrator, and derivative factor of the position PID controller.

   e. Angle Target
      i. Motor 1, Motor 2, Motor 3: Button used to select which motor you want to control.
      ii. Target angle: The target angle that you want your motor to reach
2. Mode
   a. Homing: Button used to home the motors and update the real position of the motors. Once the index of each motor is found, the real position is updated.
   b. Home: Button used to bring back the end effector to its home position, being at [x, y, z] = [0,0, −250].
   c. Transparent: Button used to disable any control of the motor. The motors are therefore transparent and are free to move without any control loop running.
   d. Position: Button used to put the delta robot into position mode. The greyed-out zone from the “Infos Trajectory” becomes accessible.
   e. Params: Button used to access the different parameters of the control loop and the trajectory planning. The greyed-out zone from the “Trajectory Target” and “PID Value” becomes accessible.
   f. Angle: Button used to access the angle target mode. The greyed-out zone from the “Angle Target” becomes accessible.
3. Status
   a. GUI Message: The different status messages when sending a command
   b. DELTA Message: The different status messages related to the state of the delta.
4. Plot
   a. Snapshot menu: Allow selection of the folder where the snapshots will be stored and a button to take a snapshot of the last recorded states of the delta robot.
   b. Top plot: Shows the real-time angle of each motor. The desired angles have to be selected in the top zone.
   c. Bottom Plot: Show the real-time position of the end effector. The desired positions have to be selected in the top zone.

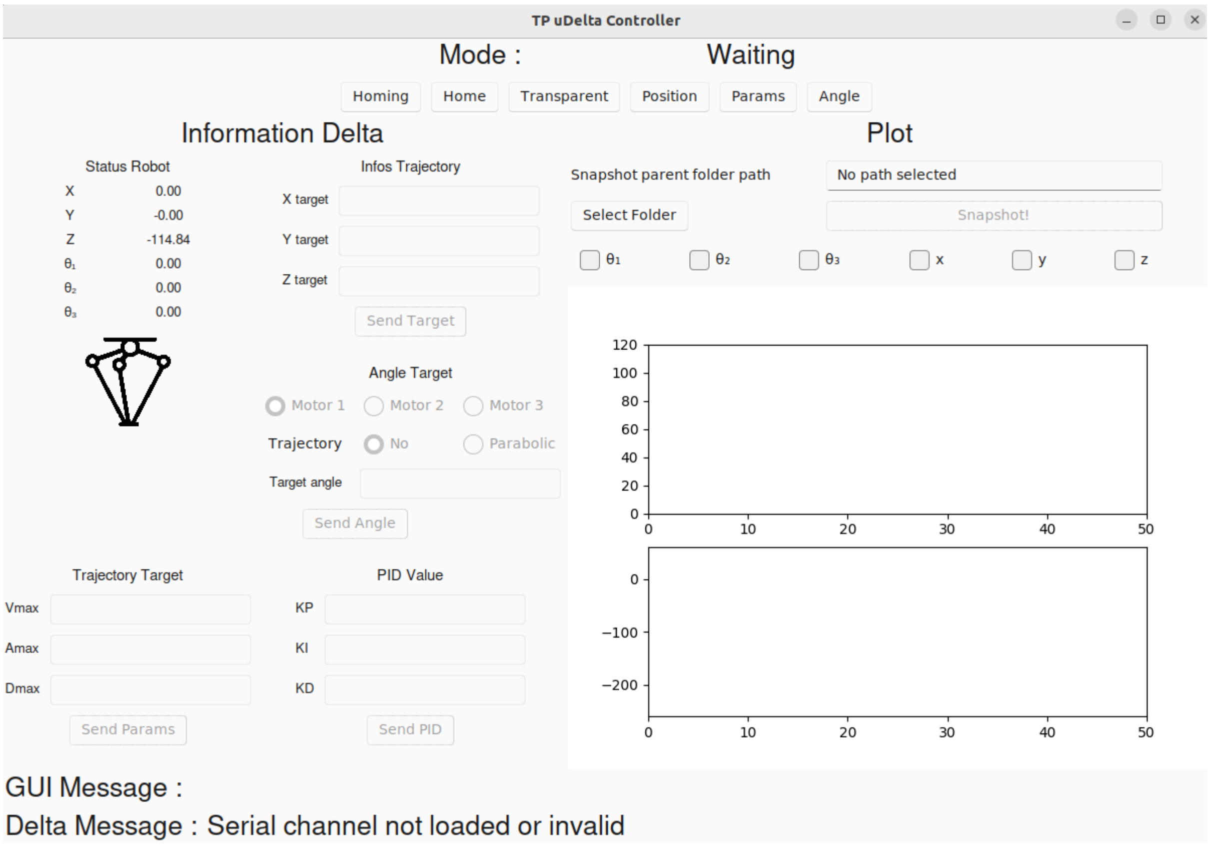

Figure 4: An example of the MicroDelta control software running

Homing procedure

Be careful and read this section well before attempting anything.
It is advised to re-run this procedure after every abnormal system failure or crash.

Start by plugging the emergency stop into the delta robot. Then you can plug the power supply into the emergency stop.

To move the robot, you will have to first home each motor.
First, start by pushing the platform at its lowest position, leading to align the arms and forearms together. It will allow each motor to be in the right sector to then get the index. Once the delta's arms are in the correct position, you can plug the power supply and you can now connect the robot to the computer, launch the program with the correct COM parameter, and select the "Homing" button in the GUI. Each motor will start a homing procedure, slowly moving until reaching the index. Once the index has been found, the motor will show an angle of approximately 90°.

Should you close the program, and the robot does not enter a fail-mode, you can simply rerun the program, but if the robot enters a fail-mode, you should unplug the robot, which will cause all the robot's lights to be off, before restarting the homing procedure.

XYZ Direction

After a correct homing procedure, activate the *Transparent mode*. In this mode, you can freely but delicately move the robot end-effector with your hand. You can also rotate the outer pulley by touching only the metallic part. Doing so, don't touch the rubber bands as they would interfere with what the robot has measured during the homing procedure. Move the end-effector delicately and guess how the axes are.
In your report, provide a schematic of your Delta robot with its number (written on top of it) and with the x, y, and z axes correctly placed. Draw where the coordinate (0;0;0) should be.

Home position

After a correct homing procedure, or after the transparent mode where you put the end-effector in coordinates around (0;0;-250), go to the *Home* position by pressing the *Home* button. What could be this home position? Try to gently push the end-effector with your hand (remove your face from the immediate vicinity of the robot, do not put your finger inside the hole of the end-effector). What do you observe, or did you observe?
In your report, explain the difference between the results after pressing the *Homing* button and after pressing the *Home* button. And explain why this point has been chosen as the "home" position. What happened when you tried to push the end-effector?

PID observation

As a reminder, on our MicroDelta robot, the PCB board controls motors in current only and can read the motors' positions through encoders. There are no other sensors or actuators on this robot. Knowing that, propose a control diagram to control in position of the end-effector. The position to be reached by the end-effector is given by the computer to the robot. Use a PID as a regulator and don't use an open-loop regulator.
On our MicroDelta robot, the position of the end-effector is controlled by a PID. Activate the plotting of all variables (the three angles and the three axes). Using the *Position* mode of the robot, generate some movements. Observe the angle values and the position. And take a screenshot of these plots. Try to find what the unit and graduation scale of the plot's x-axis are (give your answer in seconds ± 0.1 seconds). Hint: It is a round number.

In your report, give your proposed control diagram (1) and justify the choices you made (2). Provide one screenshot of the positions and angles when you use the robot (3). Give the unit and graduation scale of the plot's x-axis (4). And explain how you have measured it (5). You must include in your report the code written on top of the robot. The answers might vary from one robot to another.

Snapshot

From the previous point, you should have realised it is very hard to read and understand on these plots the values corresponding to the movements you order.

In the software, there is the Snapshot option, which will dump the raw information from the robot directly. New pieces of information are generated when the robot estimates that there are values relevant to be transmitted. If the robot stands perfectly still, then there is no new information to be transmitted. If it is moving, then there is constantly new information being generated.
A snapshot dumps in a CSV file the last 2,000 relevant information.

Select a folder, for example, the *Desktop* folder, in the *Snapshot* menu, and click the *Snapshot* button. Load this file with Excel, LibreOffice, or OpenOffice. Try to understand its format. Do not hesitate to ask for help if needed.

After having well understood how to use this *Snapshot* tool, try to snapshot a step movement along the horizontal plane of at least 40 mm. Make sure not to have changed the default parameters in the Delta robot with the *Params* mode, for example. If you did, shut down, unplug, and restart your robot as well as the program on the computer.

<u>In your report</u>, provide the plot you have just generated. Specify clearly what corresponds to the axes, specify the units, and scale. What is the type of PID implemented in this Delta robot, and why? (Underdamped, critically damped, or overdamped)
On your chart, plot the following parameters and explain them if needed:

- The initial value
- The final value
- The overshoot in per cent (if any)
- The steady-state error in per cent (if any)
- The delay in seconds
- The rise time in seconds
- The peak time in seconds (if any)
- The decay ratio (if any)

Beware: Plotting and computing all these parameters can take some time. During the lab session, just be sure to leave with at least a valid measure.

PID Tuning

You will now explore in a “guided tour” the effects of the PID parameters. In that case, a PID is a regulator. As a reminder, a PID (Proportional-Integral-Derivative) controller is a feedback control system that adjusts a process variable to match a desired setpoint. It consists of three terms:

- P (Proportional): Corrects based on the current error.
- I (Integral): Corrects based on the accumulated error over time.
- D (Derivative): Corrects based on the rate of change of the error, anticipating future behaviour.

Please check Prof. Bouri’s course on PID if you have forgotten about it.

Make sure you have the robot with the default PID params before starting. If not, shut down, unplug, and restart the robot as well as the software on the computer. During this "guided tour", you will use the *Home* and the *Position* mode. Make sure to have well understanding of how to use them before starting.
You will be asked to observe certain behaviours or phenomena. If they don't occur, slightly increase the weights given to you. If they still don't occur, call an assistant.

1. Go to *Home* position, which is (0, 0, -250), and then go to the position (0, 0, -240). Try to gently but firmly move the end-effector by hand. You should not note anything particular, but remember for the next steps how the system behaves.
2. Go to *Home* position and set the *Params* to Kp = 20, Ki = 0, Kd = 0 (we have deactivated the integrator and the derivative). Then go to the position (0, 0, -240). You should note *something* different. Try to very gently move the end-effector by hand. You should note *something*. In your report, explain these *things* and why they have now appeared.
3. Go to Home position and set the *Params* to Kp = 20, Ki = 0, Kd = 1 (we have deactivated the integrator only). Then go to the position (0, 0, -240). That is a problem of the previous point has now disappeared (or is being mitigated), but *something* new has now appeared. Do you hear some noise that could be a symptom of that *something*? Do you note any difference in the values, especially of the angles, on the computer interface? In your report, explain these *things*. What has now disappeared, and why have new things now appeared?
4. Go to Home position and set the *Params* to Kp = 30, Ki = 0, Kd = 0.01 (we have deactivated the integrator only). Wait that the robot is indeed in coordinates (0, 0, -250). Then try to move the end-effector with your hand. When you move the end-effector and release it, observe how fast it reaches again the coordinates (0, 0, 250). At the end, did the end-effector really reach these coordinates again?
   Now set the *Params* to Kp = 30, Ki = 1, Kd = 0.01, and do the same observation as before. What do you observe now? In your report, write the difference and explain why this happened.
5. Go to Home position, and this time try to set yourself the PID values you think will imitate or beat the performance you have seen in point 1. In your report, give these values and also provide a *Snapshot* of a step movement. Write down explicitly what this movement was.

Singularities

Sadly, the robot delta presents multiple singularities. As a reminder, a mechanical singularity is a physical configuration where the behaviour cannot be controlled or predicted. They may appear when reaching the limits of the working environment, for example. The number of degrees of freedom changes in a singularity configuration.

Put the robot in *Transparent* mode. You can now freely move the end-effector with your hand the end-effector. The end-effector coordinates are still being displayed on the screen.
For each of the following singularity configurations, try to reach the point of singularity (or *a* point without) without breaking the robot. Write down the end-effector coordinates x, y, z, and

the angles displayed on the software interface. Draw on paper a small schematic of the configuration.

Singularities to observe:

- The arms and their respective forearms become collinear
- The forearms are all aligned in the same plane
- The workspace becomes degenerate
- A mechanical limit has been reached.

In your report, give for every singularity the coordinates and angle values, the schematic of the configuration, and which degree of freedom is won or lost. Also, write a small proof or explanation on why it is a singularity.

Two-armed robot (bonus)

Start the robot as usual, after a legal *Homing*, go to the *Home* position. Now, carefully disconnect the third arm from its forearm. (Remove your face from the robot and don't leave your fingers in the robot, be ready to press the emergency stop button.)
Did the robot move very violently? Why or why not? What are now the "new" targeted values for each motor and why? Comment on the angles' value, but also their noise.
In your report, write your observations.

Simulation

How to install

Download the provided zip file from Moodle and extract it to your desired position.
Note: You should have a recent version of MATLAB already installed. The simulation is implemented with MATLAB 2024a.

How to run

Open MATLAB and go to the directory in which you extracted the zip file. Right-click on the “Simple Delta Simulation” folder. Click on the “Add to Path” button and then select “Selected Folders and Subfolders”. Open the “Simple Delta Simulation” folder. First run “startup.m”. Then run “DeltaModelMain_v2.m”. Finally, run “main_TimeSimulation.m”. You should now see the window showing the movement of your delta robot.

Experiments

Inside the “main_TimeSimulation.m”, you can see the input and target joint angles given to the Simulink. Try changing them to different values and observe the results. You are supposed to implement the Inverse Kinematics function and call it here so that the values provided by you are changed from joint angles to end effector position. You can observe the functions used in the Simulink inside the FunctionLib folder.

Try observing the singularities of the delta robot in the Simulink.

Bonus Question 1: The controller used in the “main_TimeSimulation.m” function is called “DeltaVerticalControl.m”. Open this function (located inside the FunctionLib folder) and observe it. As you can see, this function controls the robot only in the case of vertical movements (equal joint angles). Implement a similar function called “DeltaControl.m” which can handle other movements of the delta robot (including horizontal movements of the end effector). Be careful about the results (especially location, velocity, and acceleration of the end effector) generated. These results should be compatible with the animation generated. Note that your function should reproduce the same results as the “DeltaVerticalControl.m” in case it is given three identical joint angles.
Bonus Question 2: Try changing the input and target values. Observe the plots generated. In some cases, you should see unexpected jumps in the acceleration plot, which are not compatible with the behaviour (the generated animation) of the robot. Can you explain the reason for producing these unexpected jumps inside the plots?

Reporting

In your report, include your implementation of the inverse kinematics. Go from one state to another first with end effector position as input, and then with the corresponding joint angles as input (the case without inverse kinematics). Provide the outputs showing that the 2 cases are equivalent. (Including the output in the command window and the plots)

Provide the plots of Joint Angles, Motor torques, End Effector position, End Effector velocity, and End Effector acceleration for:

1. a single movement between two states
2. a movement containing/going to a chosen singularity

Analyse the plots and explain the observations.