Robotic practical 7:

Control of the Micro Delta Direct Drive robot

Group 29

BAILLY Xavier, SPIESS Philipp, URAN Marc

April 2019

1. Introduction

The delta robot is a parallel robot that has 3 motors. There are many types of the delta robot but the one used for this report is one of the simplest ones. At first, we answer the theory questions. Afterwards we will try different type controllers on one motor. In the end we will try different types of controllers for the whole robot.

2. Theoretical questions

a. Kinematic

1. The Grübler formula to compute the mobility defined is as follows:

$$M = \sum_{i=0}^{j} fi \ - 6 \times b$$

n = number of closed loops
j = number of joints
f_i = number of DOF of joint i

We have 5 closed loops (n=5) and 12 ball (3 DOF each) and 3 pivots (1 DOF each) :

$$M = 12 \times 3 + 3 \ \times \ 1 - 6 \times 5 = 39 - 30 \ \rightarrow M = 9$$

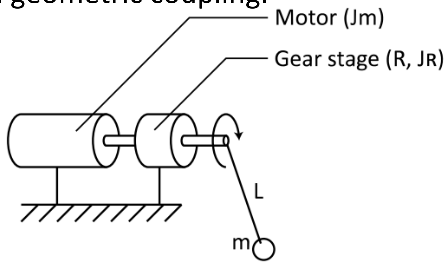
The mobility is equal to 9 but the actual number of DOF of the robot is 3.

2. It is explained by the fact that there are 6 internal DOF. These internal DOF come from the ball joints that allow the forearm's rods to spin.

3. To get rid of these internal DOF we would need to replace the ball joints to cardan joint. The ball joints are cheaper, less complex, lighter and easier to assemble then the cardan joints. It is therefore more advantageous have ball joints.

b. Joint control

1. A single motor is easier to model because it doesn't have to take into account the other motors of the robot. With the whole robot there is dynamic and geometric coupling.



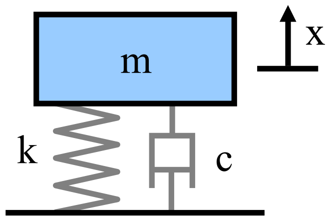
2. The inertia of the mass at the entry of the gear stage is equal to $m \times L^2$. The total inertia at the motor is equal to $J_{tot} = \frac{mL^2}{R^2} + J_{gear} + J_{motor}$

3. A big reduction stage allows to reduce the mass inertia; it is then easier to control because the inertia is less influenced by load change or dynamical coupling. Also, if the resolution of the motor's encoder is not sufficient for the application, a reduction stage allows to be more precise and cheaper motors or encoders. A reductor could also induce more maintenance, weight, complexity and unnecessary costs.

4. The Ziegler-Nichols method is a heuristic method that can work in two ways; open-loop or closed loop. The open loop Z-N method analyzes the results of a step test and determines the PID values. It only works if the system is linear. The closed loop Z-N method increases the proportional gain until the step input gives a sustained oscillation output. This critical gain is then used to determine Kp, Td and Ti. It only works if the system is of first order.

5. The Ziegler-Nichols method is not a good method because the system is of second order and has non-linearity depending on position and speed.

6. This system represents a PD controller for the system. The spring represents the proportional part P and the damper represents the derivative part D.



https://en.wikipedia.org/wiki/File:Mass_spring_damper.png

3. Robot control and trajectories

1. A centralized controller architecture has one controller which controls all the motors at the same time. A decentralized controller architecture has a controller for each motor, it doesn't have any information about the rest of the motors. Centralized control is more complex and needs more expensive hardware. Decentralized control is easier to program and needs less performing hardware but the control functions not as well as with centralized control.

2. The kinematic models allow to describe the motion of the robot and the dynamic models allow to describe the forces responsible for the motion of the robot.

3. The torque felt at the motors is a sum of inertia times acceleration (B), friction(F), gravity(G), external force times the inverse Jacobian matrix, Coriolis and stiffness.

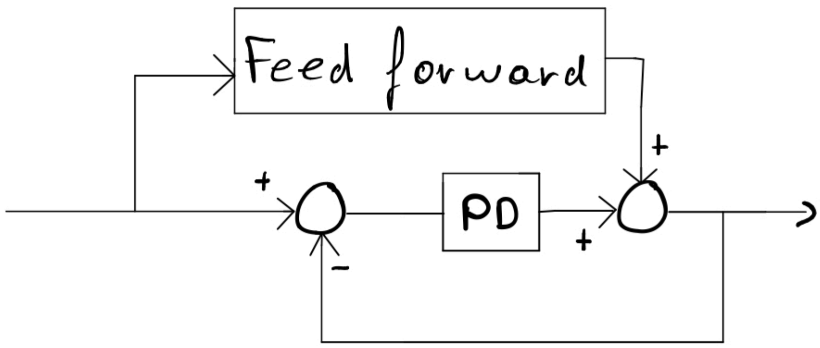
$$\Gamma = B(q) \times \ddot{q} + F(q, \dot{q}) + G(q) + F_{ext} \times J^T + C(q\,, \ddot{q}) + K(q, \dot{q})$$

We consider that the robot is rigid (→K=0) , not going very fast (→C=0) and doesn't have any load (→ F_{ext}=0). The torque becomes:

$$\Gamma = B(q) \times \ddot{q} + F(q, \dot{q}) + G(q)$$

The only component that is very difficult to model is the friction.

4. block diagram of a feedforward regulator with inverse dynamical model:



4. Practical work

Regulation of a single axis

1. P regulator

a. This controller is not sufficient for position control because there is a big overshoot, steady state error and the system is unstable with fast target dynamics.

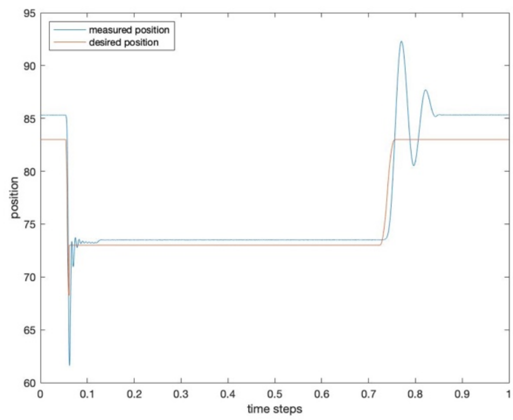


Figure 1

Slow target dynamic with position step of 10°.

100% overshoot compared to step value

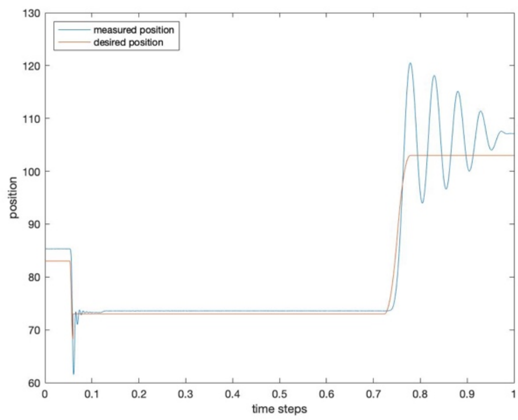


Figure 2

Slow target dynamic with position step of 30°.

56% overshoot compared to step value

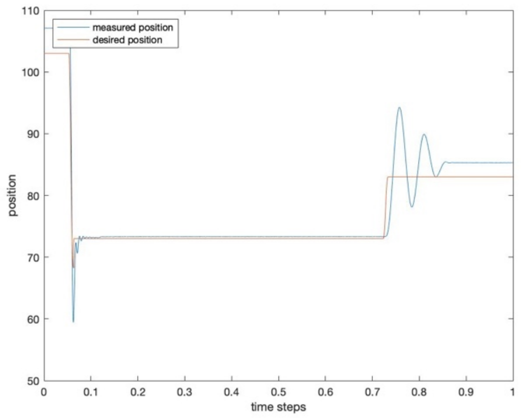


Figure 3
Fast target dynamic with position step of 10°.
100% overshoot compared to step value

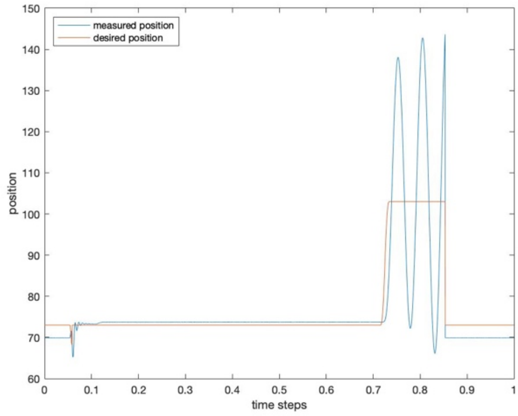


Figure 4
Fast target dynamic with position step of 30°.
140% overshoot compared to step value

b. We can deduce from Figures 1 to 4 there is a bigger overshoot when the step is bigger. We can also observe that in figure 4 the system is unstable. The damping time and overshoot are not increased by the target dynamic (except when unstable), but they are increased by the position step amplitude.

2. PD regulator

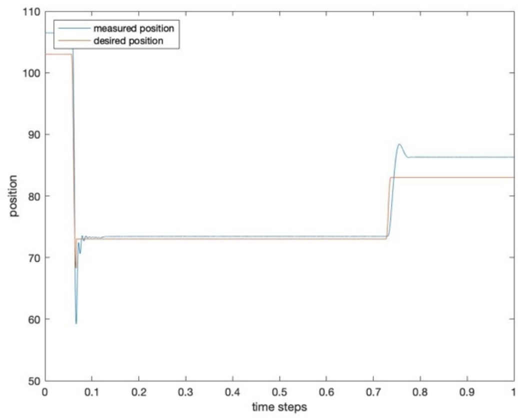


Figure 5
Kp = 0,01, Td=0.01 with position step of 10°.
60 % overshoot compared to step value

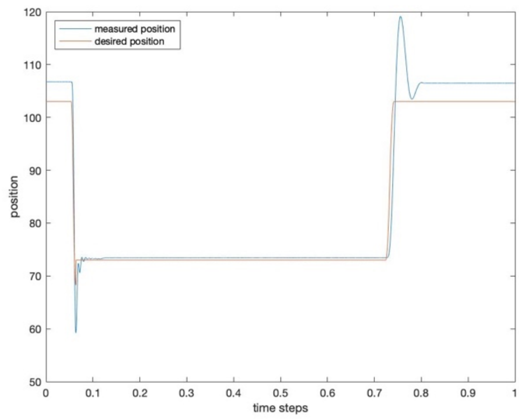


Figure 6
Kp = 0,01, Td=0.01 with position step of 30°.
53% overshoot compared to step value

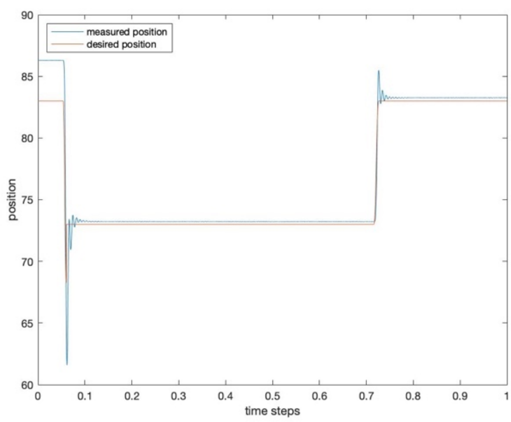


Figure 7
Kp = 0,1, Td=0.01 with position step of 10°.
30% overshoot compared to step value

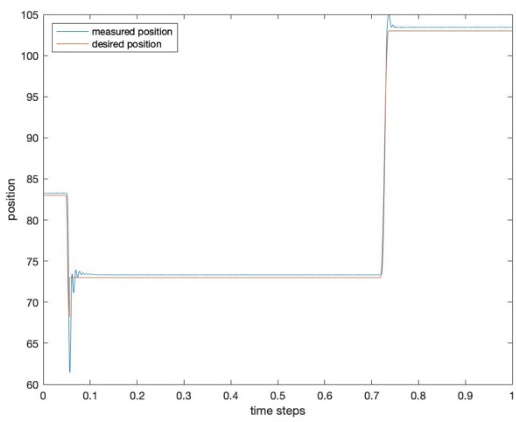


Figure 8
Kp = 0,1, Td=0.01 with position step of 30°.
6% overshoot compared to step value

a. As shown on Figures 5 to 8 the derivator makes sure there is less overshoot and the system is stable. There is still steady state error, but it is lower if we increase Kp.

b. As shown on Figures 5 to 8 even with the fast target dynamic the system remains stable thanks to the derivator.

c. As shown on figure 9 when we put Td = 0.03 the arm begins to oscillate, when a mass is put on the arm, the oscillation stops. An explanation is that the noise of the sensor gets amplified by the derivator because the noise has the system's resonance frequency. Adding a mass changes, the resonance frequency and so the noise is no longer amplified.

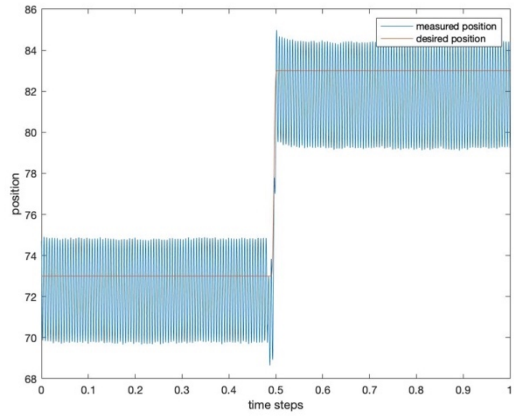


Figure 9
Kp = 0,1, Td=0.03 with position step of 10°.

3. PID regulator

a. As shown on figure 10 the smaller Ti the less error there is before and during the ramp. The steady state error is therefore reduce by the integrator.

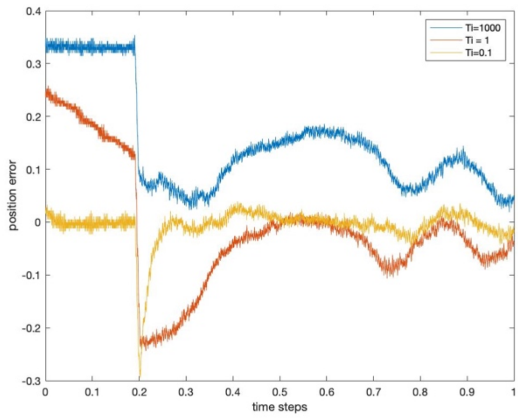


Figure 10
Error due to ramp with Ti = 1000, 1 and 0.1

Regulation of the full robot

1. Singularities

a. To avoid singularities either we reduce the working area or induce an error in the Jacobian to always avoid the singularities. We forgot to take picture of the singularities but they happen when the axis are in the same plane. When the robot arms are totally stretched or totally compressed a new DOF is possible (rotation).

2. Control performance with PID regulator

a. The shape of the target trajectories shown on Figure 11 allows us to visualize the command given to the PID controllers. The target position is reached faster if the speed and acceleration allowed are higher. The desired velocity shown on figure 12 are pulses limited by the max speed which is logical because the derivative of a unit step is δ(t). The desired acceleration shown on figure 13 are 2 pulses of opposed direction maxed by the maximum acceleration.

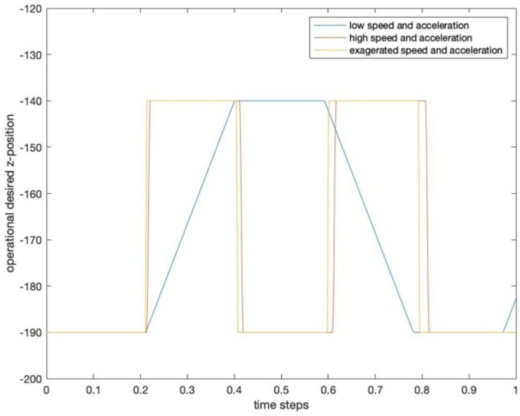


Figure 11
Desired position $z(t)$

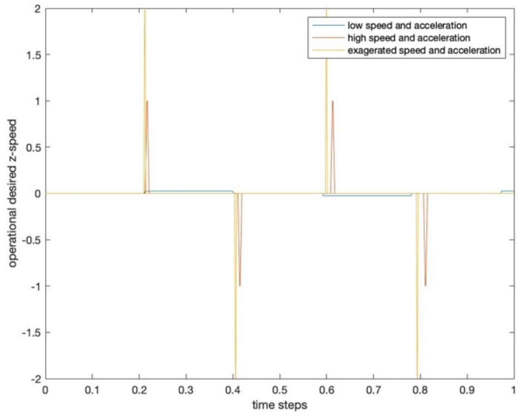


Figure 12
Desired velocity $\dot{z}(t)$

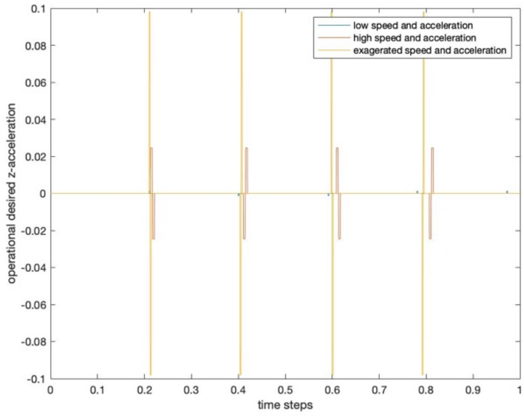


Figure 13
Desired acceleration $\ddot{z}(t)$

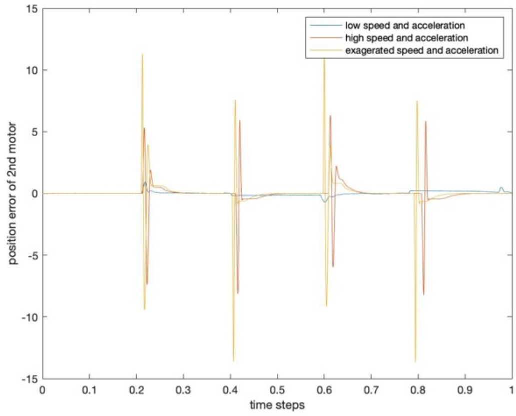


Figure 14
2nd motor position error compared to desired position with PID controller

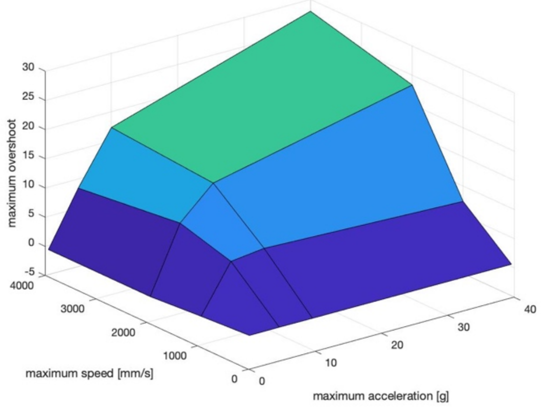


Figure 15
Maximum overshoot in function of maximum acceleration and speed.

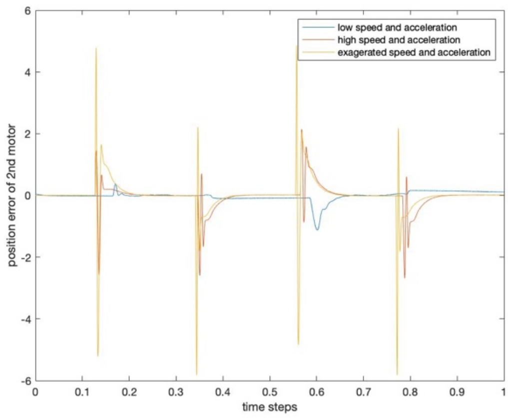


Figure 16
2nd motor position error compared to desired position with PID controller with feedforward

b. As shown on figure 14, the position error peaks most certainly come from the frictiongoing from static to dynamic and the gravity. The small errors after the peaks must come from the inertia of the sytem.

c. As shown on figure 15 the maximum overshoot increases with both maximum speed and maximum acceleration. The acceleration and speed limitations influence the desired position abruptness, it may be too fast for the system.

3. Control performance with a PID regulator with feedforward
   a. As shown on figure 16 the error peaks got reduced thanks to the feedforward. This comes from the fact that feedforward compensated the error due to gravity.

Precision

For all the precision part, some error came from the sensor not being totally immobile.

1. Best-case precision
   a. The repeatability of this robot in this position is 1.15 mm which is not bad for a manipulator robot given the sensor error. Repeatability could be limited by the quality of the calibration and the encoder.
   b. The accuracy of this robot in this is 5.715 mm which is not as important for a manipulator robot because it possible to predict the error in a model. This could come from the sensor error and incorrect model.
   c. We didn't do the precision test, but the limit would come from the encoders and the reductor stage. A non-precise encoder can be used with a big reductor stage and the robot would still be precise.

2. Other cases

Z [mm]	Accuracy [mm]	Precision [mm]
-222	3,5	56,25
-196	1	10
-126	2,5	14,25
-82	7,5	29,25

   a. As seen on the table above, the closer we get from the edges of the working area, the worst the resolution and the precision gets. It is because of the incorrect model.

5. Conclusion

We understood the differences between all the types of controllers and allowed us to understand that the max. speed and acceleration of the robot have an important role in the control. We also learned that a theoretical model is never perfect, and little noises can lead to big oscillations. In conclusion we learned about controlling a parrallel robot and especially the Delta robot.