1

Mandatory assignment 2
TEK4030

Exercise 1 - Force control - Simulink

a) ~~Forward~~ kinematics

$$xe(q) = \begin{bmatrix} a_1 c_{t1} + a_2 c_{t12} \\ a_1 s_{t1} + a_2 s_{t12} \end{bmatrix}$$

$c_{t1} = \cos(\theta_1)$
$= q(0)$
$c_{t12} = \cos(\underbrace{\theta_1}_{q(0)} + \underbrace{\theta_2}_{q(1)})$

b) since $\phi = 0$, we will use 2×2 Jacobian

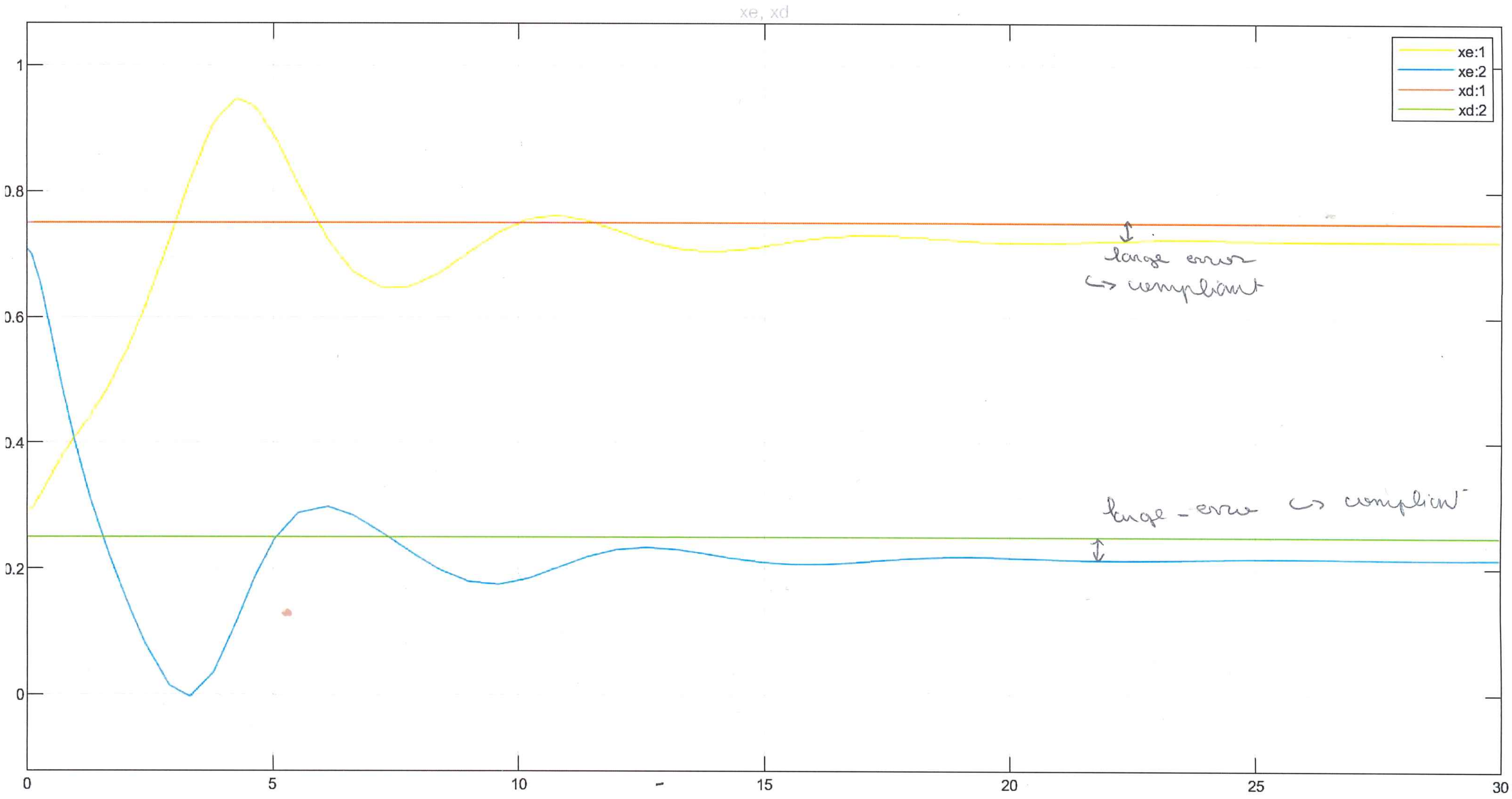
$$J(q) = J_A(q) = \begin{bmatrix} -a_1 s_{t1} - a_2 s_{t12} , & -a_2 s_{t12} \\ a_1 c_{t1} + a_2 c_{t12} , & a_2 c_{t12} \end{bmatrix}$$

c) see plots, comment on plots

d) $$\dot{J}(q) = \begin{bmatrix} -a_1 \dot{\theta}_1 c_{t1} - a_2(\dot{\theta}_1 + \dot{\theta}_2) c_{t12} , & -a_2(\dot{\theta}_1 + \dot{\theta}_2) c_{t12} \\ -a_1 \dot{\theta}_1 s_{t1} - a_2(\dot{\theta}_1 + \dot{\theta}_2) s_{t12} , & -a_2(\dot{\theta}_1 + \dot{\theta}_2) s_{t12} \end{bmatrix}$$

e)
f) $\Big\}$ see plots, comments on plots
g)

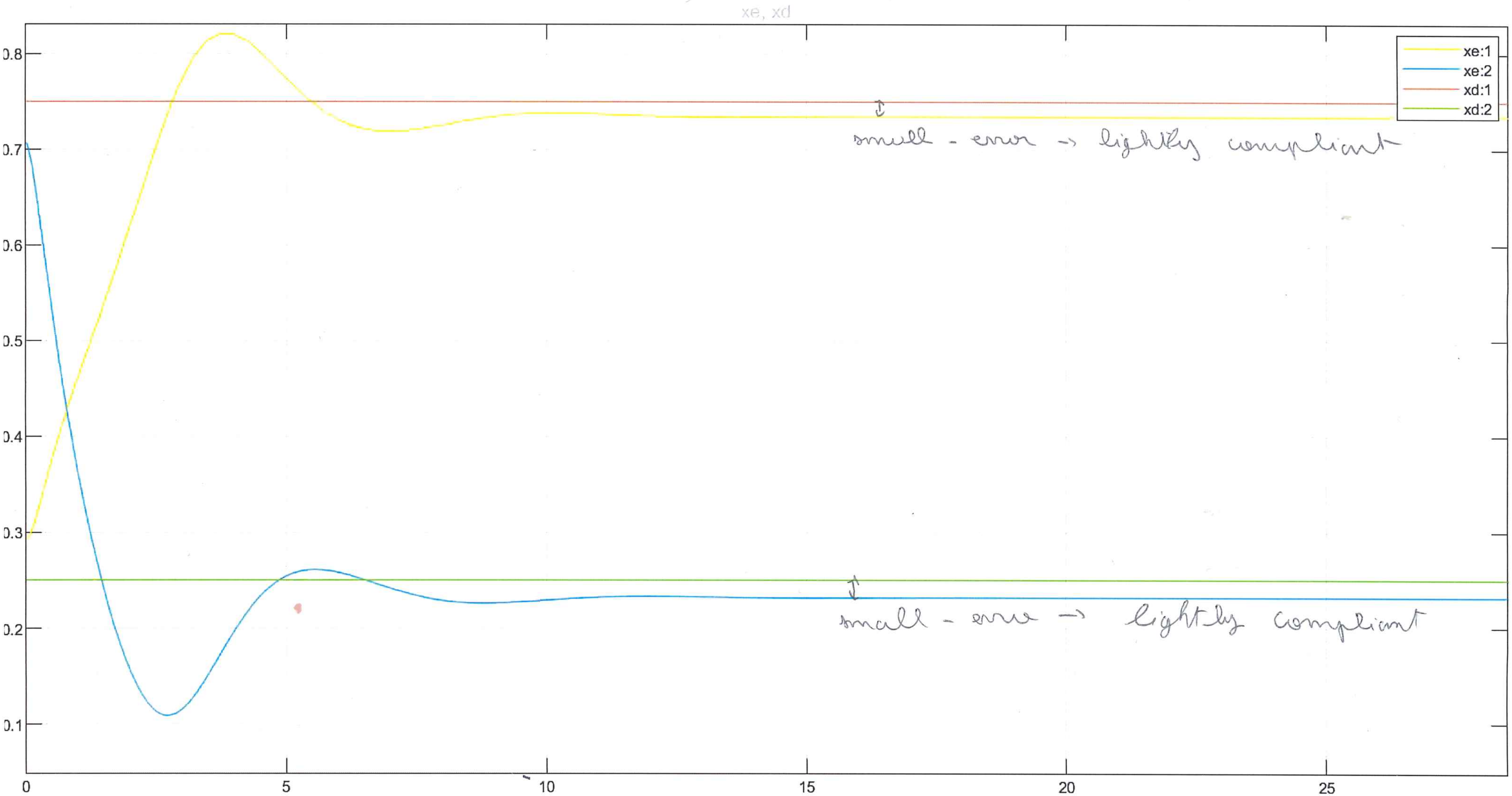
c) Active compliance (9.45) $K_p = \begin{bmatrix} 500 & 0 \\ 0 & 500 \end{bmatrix}$



2

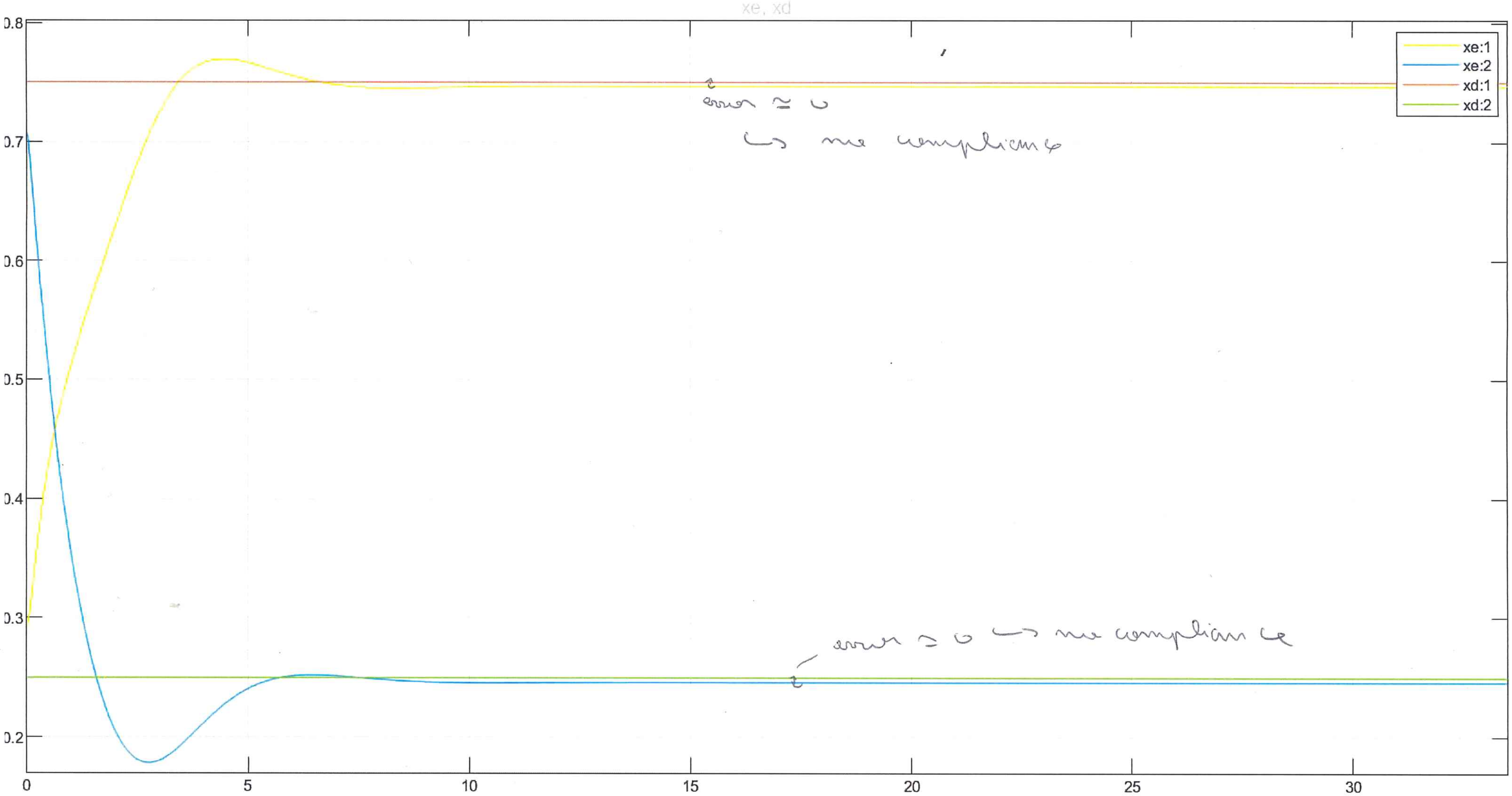
3

Active compliance (9.45) $K_p = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$



3 4

Active compliance (9.45) $K_p = \begin{bmatrix} 5000 & 0 \\ 0 & 5000 \end{bmatrix}$



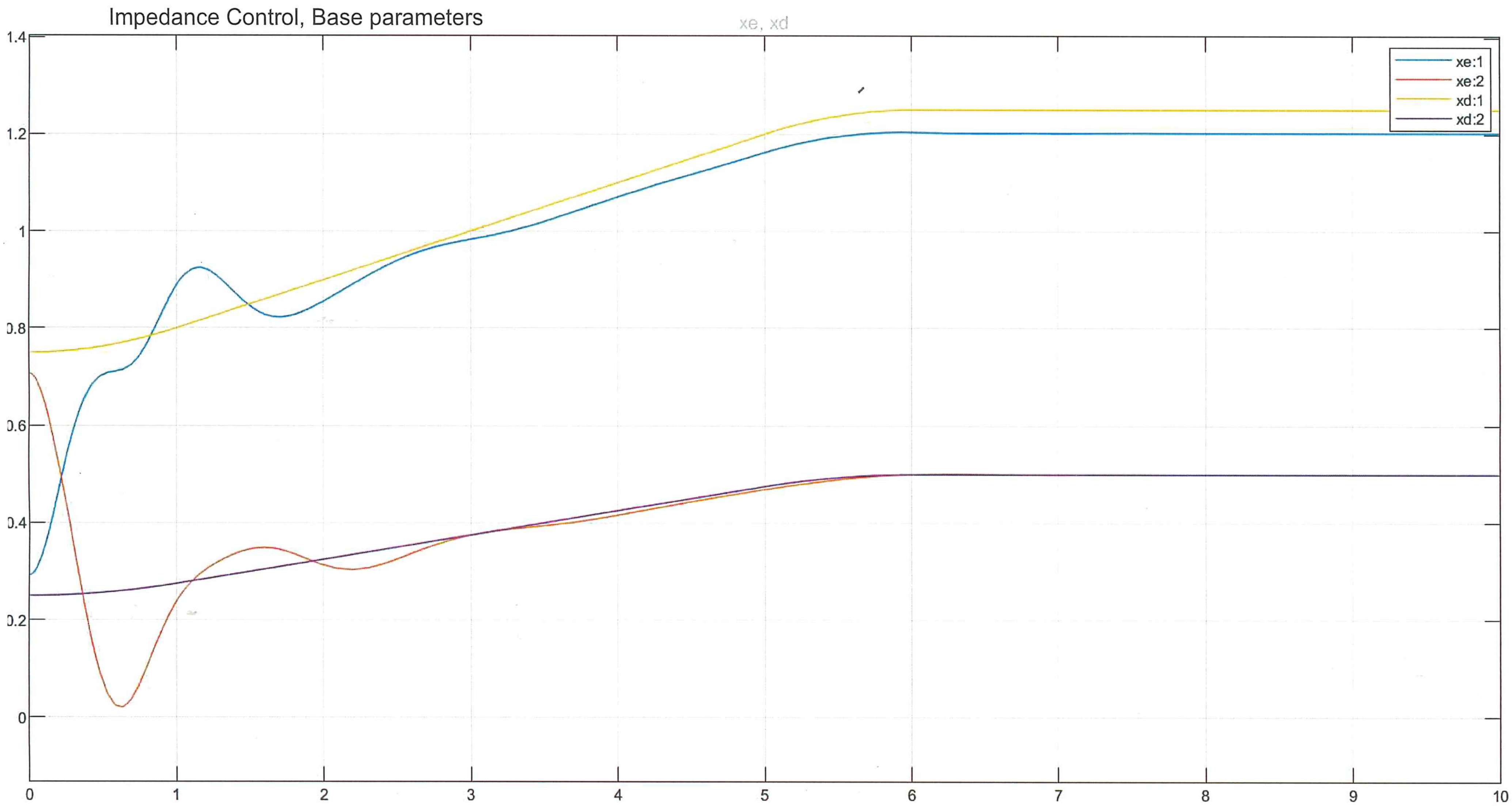
error ≃ 0
↳ no compliance

error ≃ 0 → no compliance

↳ $K_p \nearrow$ => compliance $\searrow$

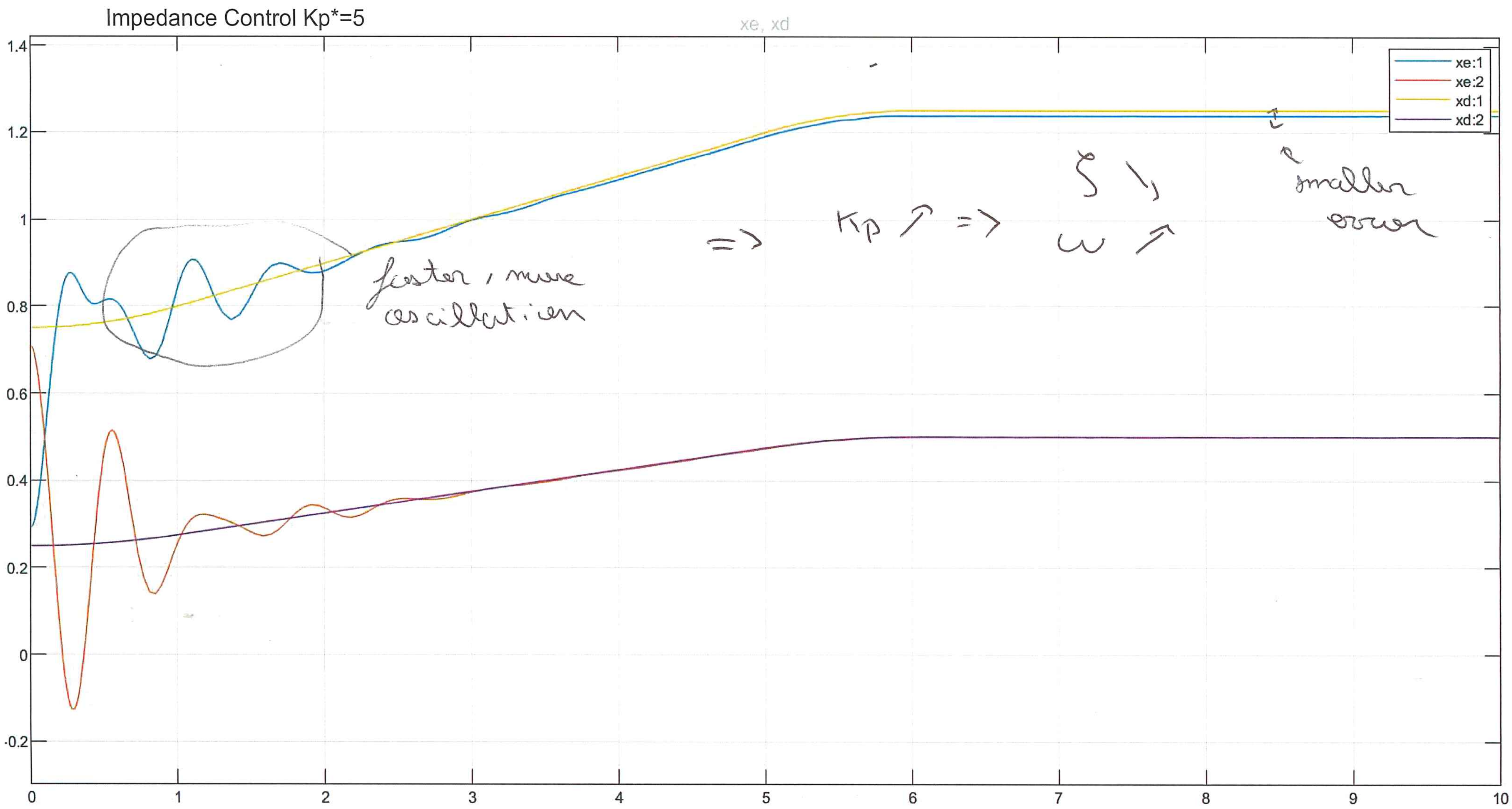
$K_p \searrow$ => compliance $\nearrow$

⑤ e)


Impedance Control, Base parameters
xe, xd
xe:1
xe:2
xd:1
xd:2
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
10

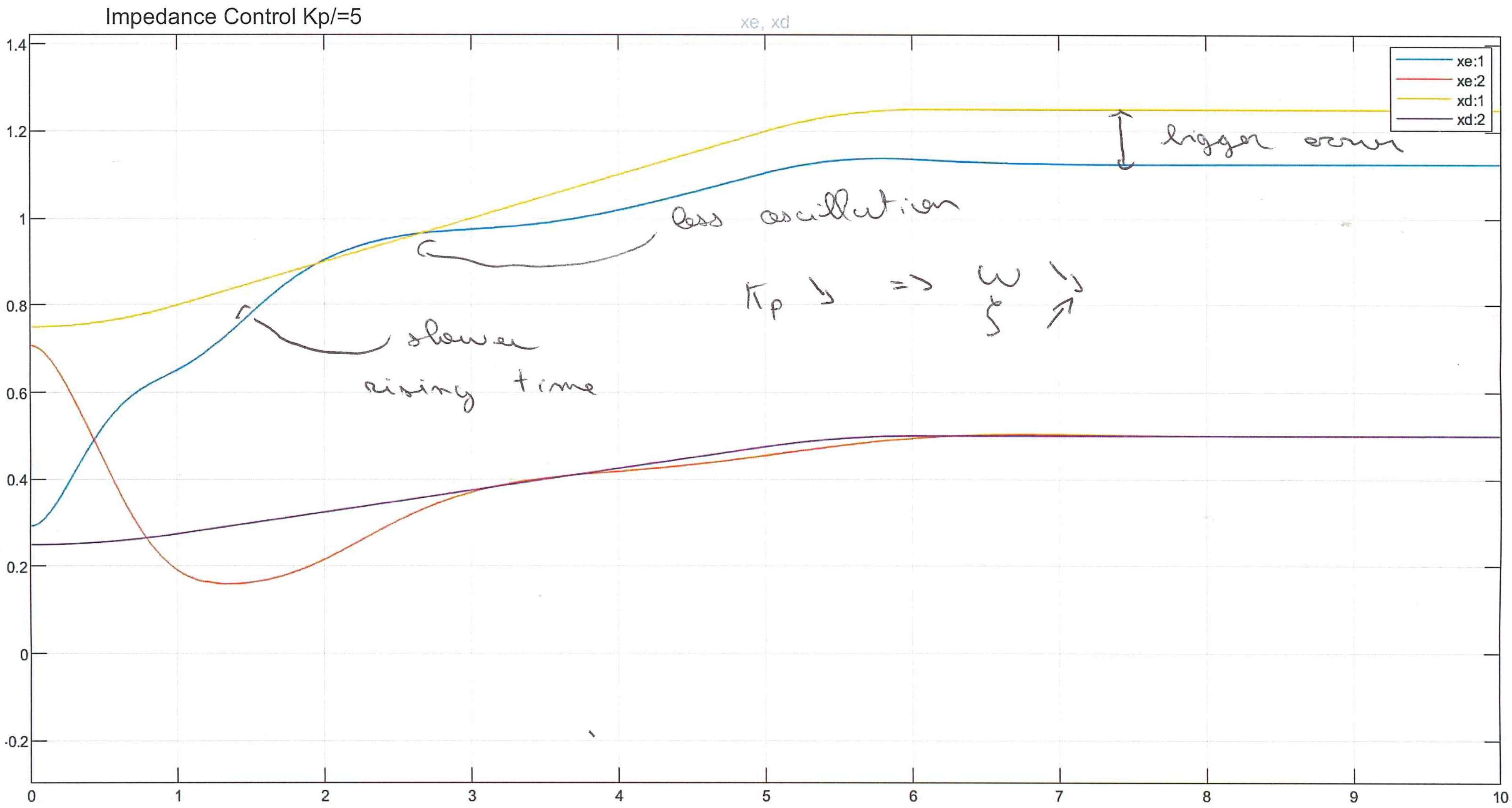
6

8)



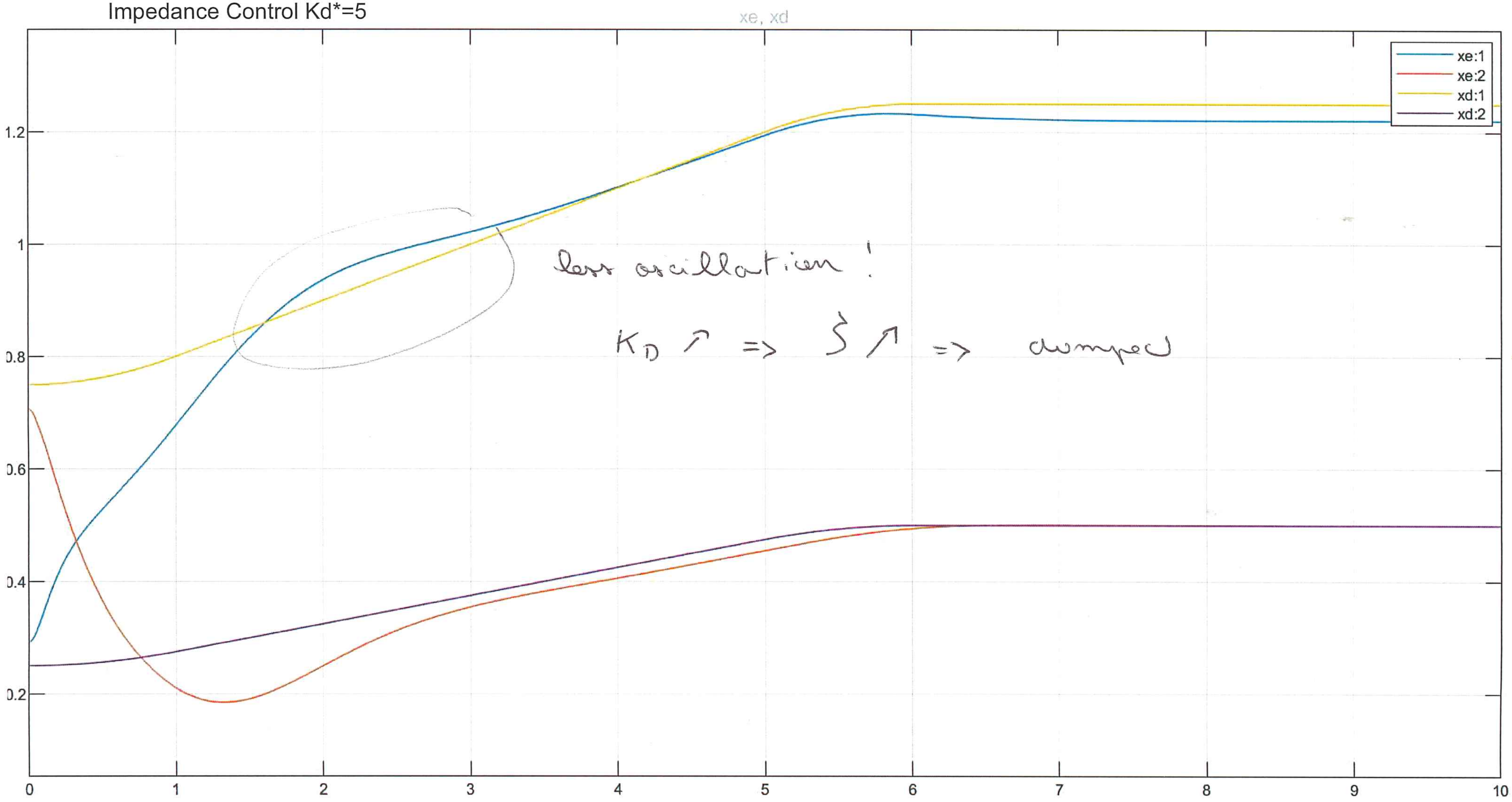
Impedance Control Kp*=5
xe, xd
xe:1
xe:2
xd:1
xd:2
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
1
2
3
4
5
6
7
8
9
10
faster, more oscillation
=> Kp ↗ => ξ ↘, ω ↗
smaller error

⑦


Impedance Control Kp/=5
xe, xd
xe:1
xe:2
xd:1
xd:2
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
1
2
3
4
5
6
7
8
9
10
bigger error
less oscillation
$K_p \searrow \Rightarrow \omega \searrow$
$\zeta \nearrow$
slower rising time

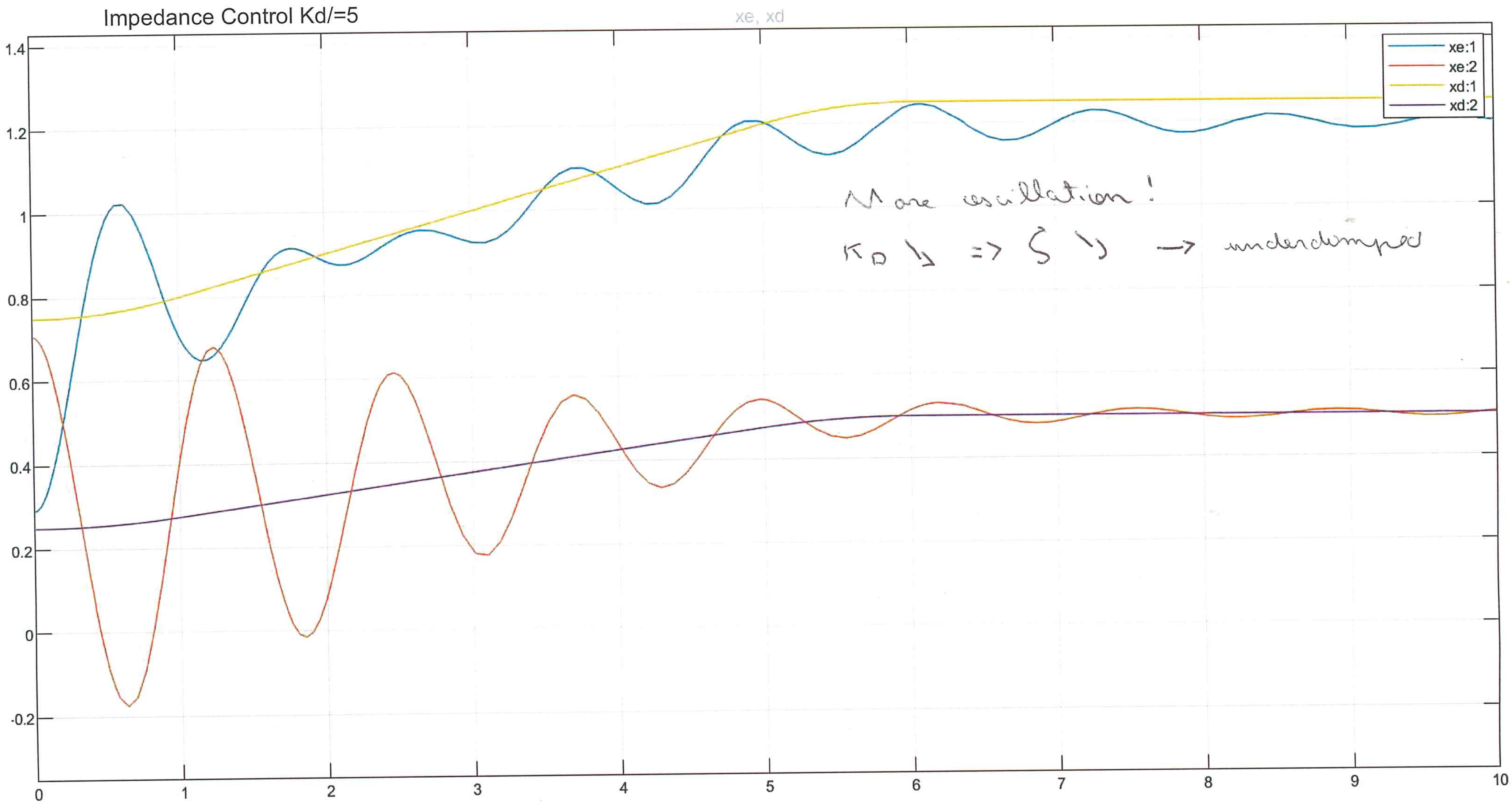
8

Impedance Control Kd*=5

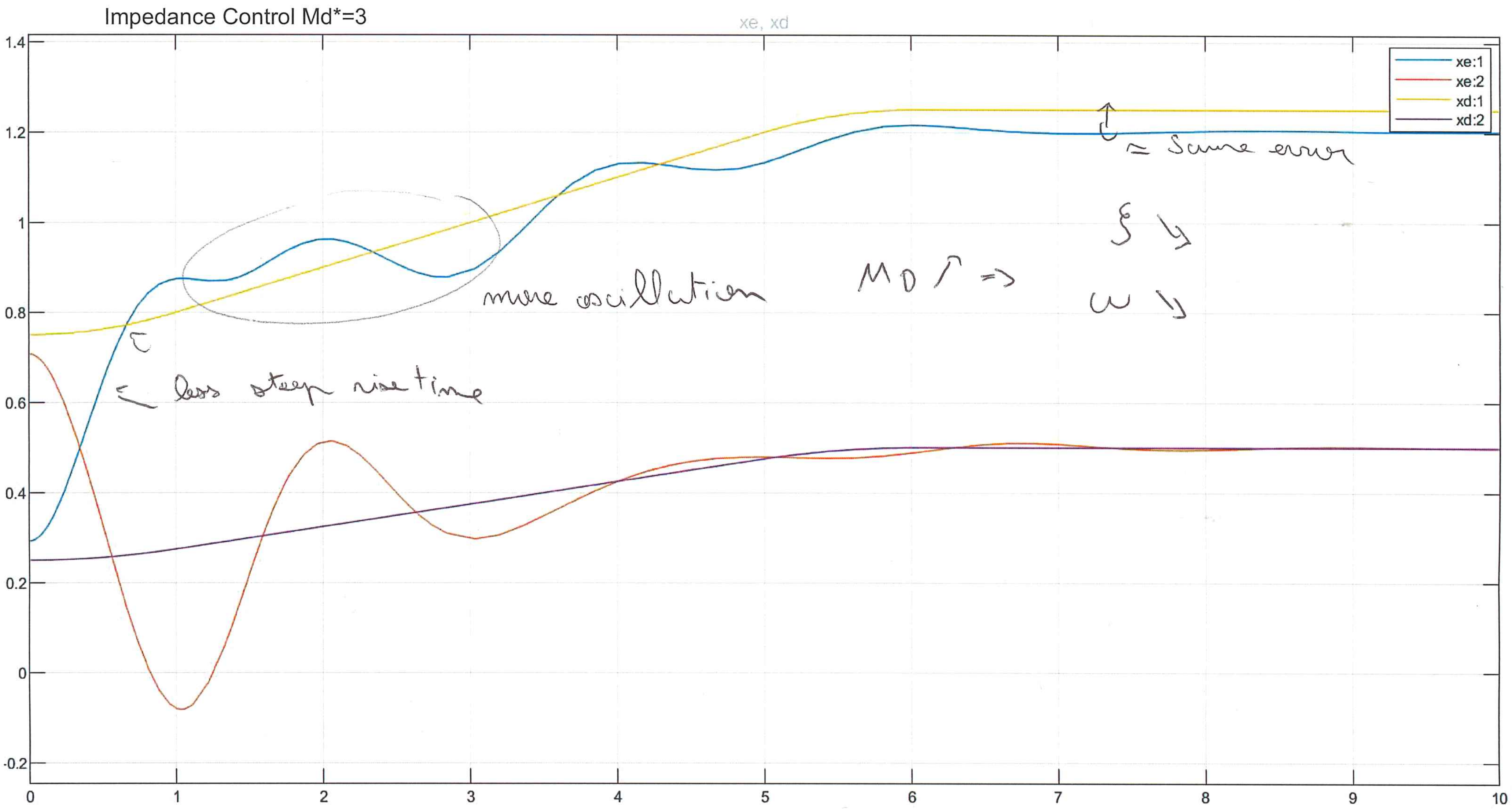


less oscillation !

$K_D \nearrow \Rightarrow \zeta \nearrow \Rightarrow$ damped

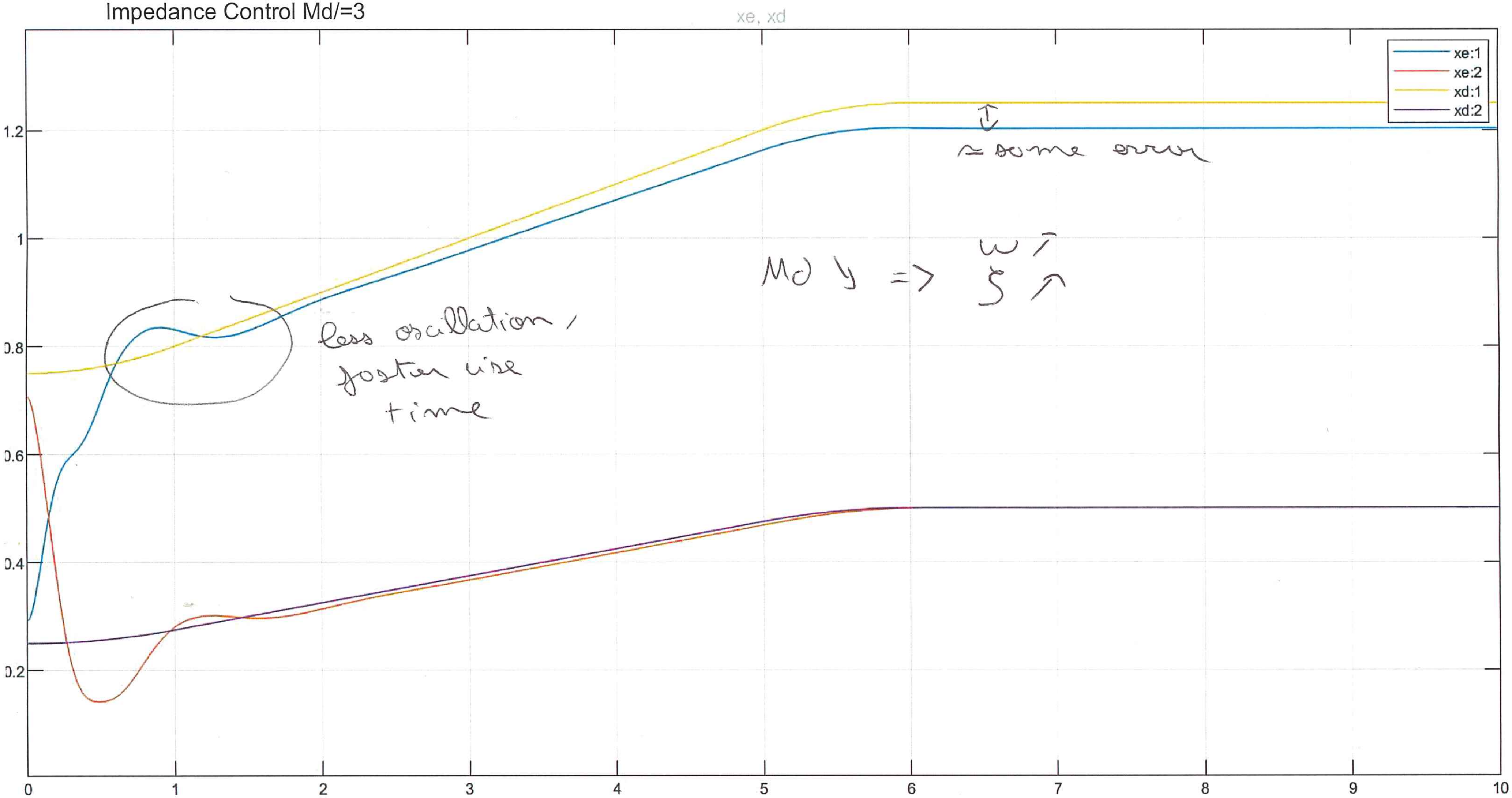

Impedance Control Kd/=5
xe, xd
xe:1
xe:2
xd:1
xd:2
More oscillation!
$K_D \searrow \Rightarrow \zeta \searrow \rightarrow$ underdamped
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
1
2
3
4
5
6
7
8
9
10

10



Impedance Control Md*=3
xe, xd
xe:1
xe:2
xd:1
xd:2
≈ same error
more oscillation
MD ↑ =>
ξ ↘
ω ↘
≈ less steep rise time

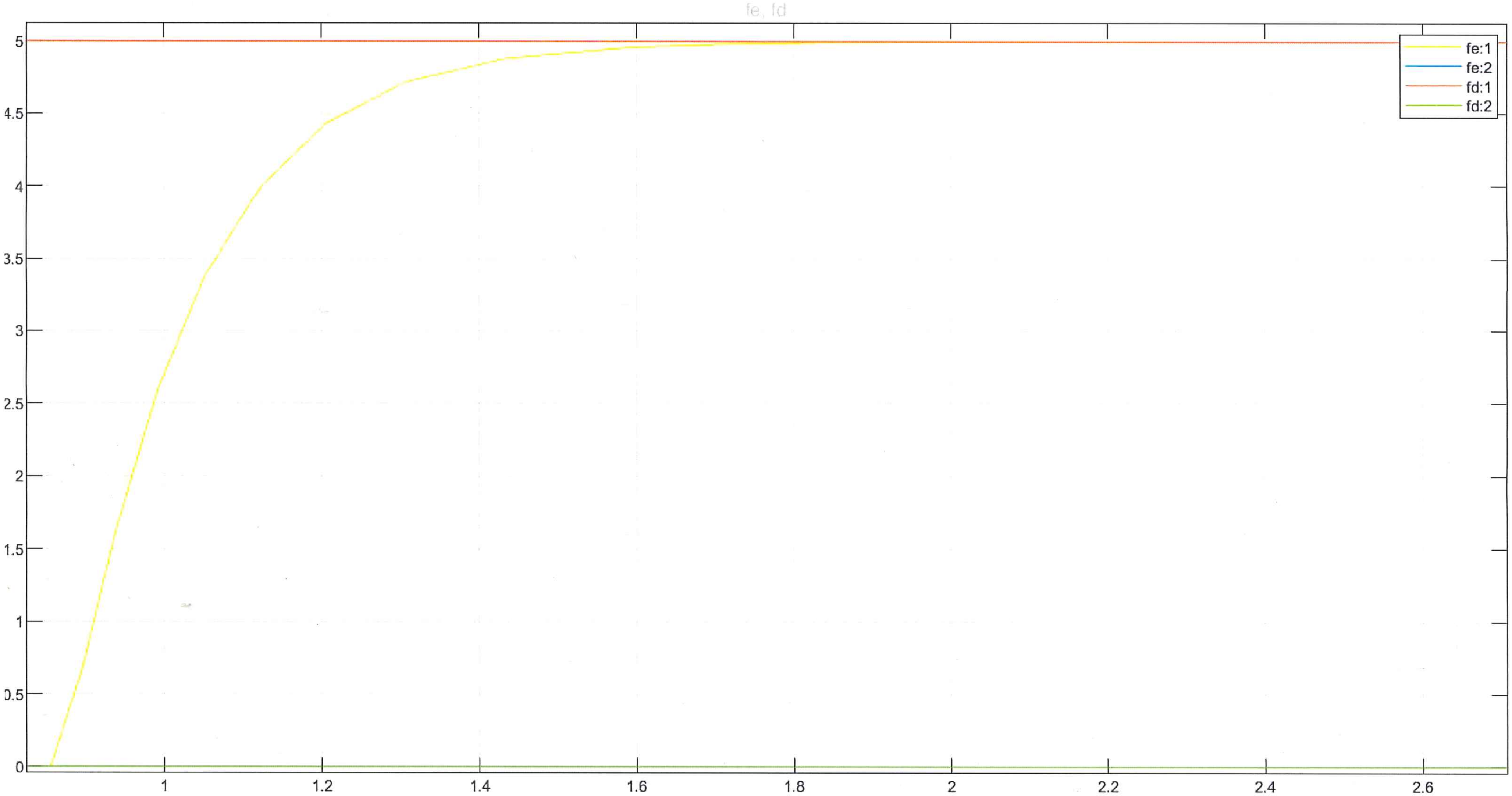
Impedance Control Md/=3



4 92

g) force control $K_f = \begin{bmatrix} 0{,}05 & 0 \\ 0 & 0{,}05 \end{bmatrix}$ $K_p \to$ same as 4.2..

if k_x changes ; $k_x \searrow \Rightarrow$ overshoot ; $k_x \nearrow \Rightarrow$ undershoot, slower

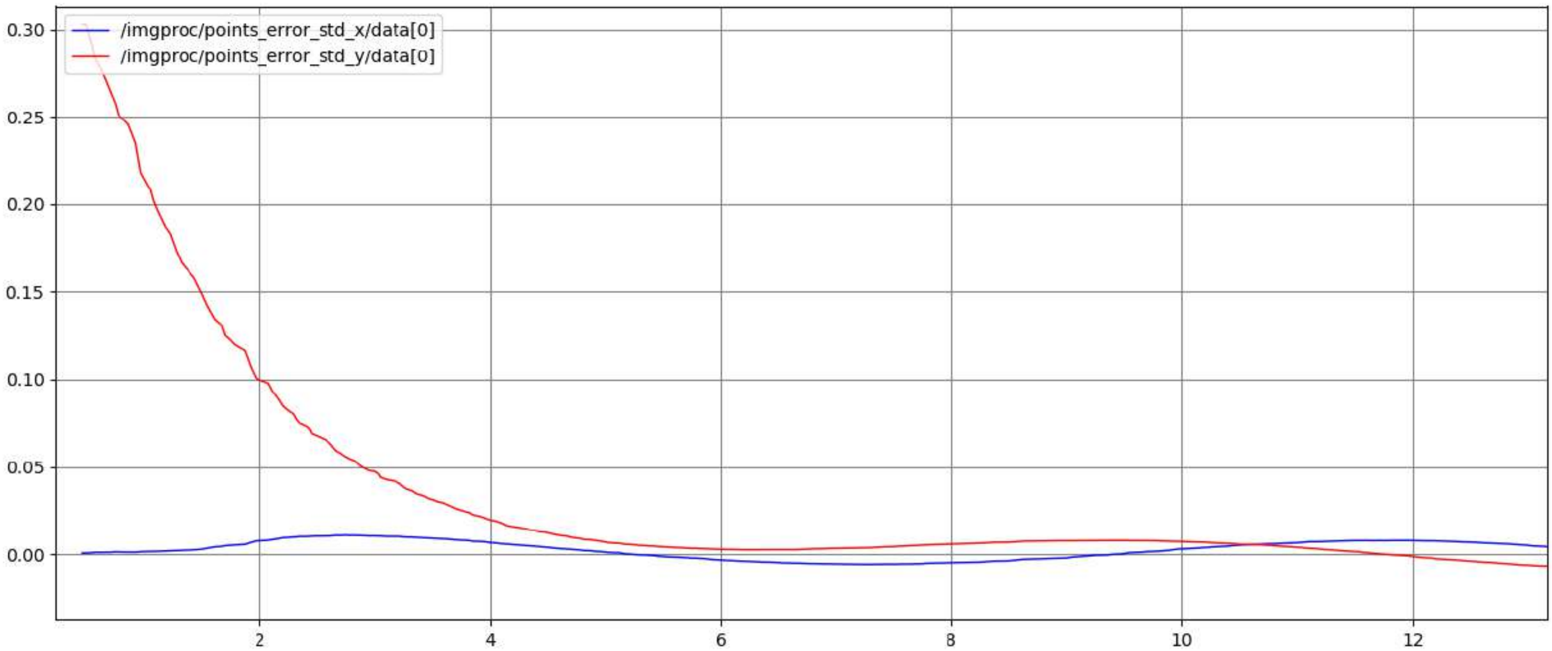


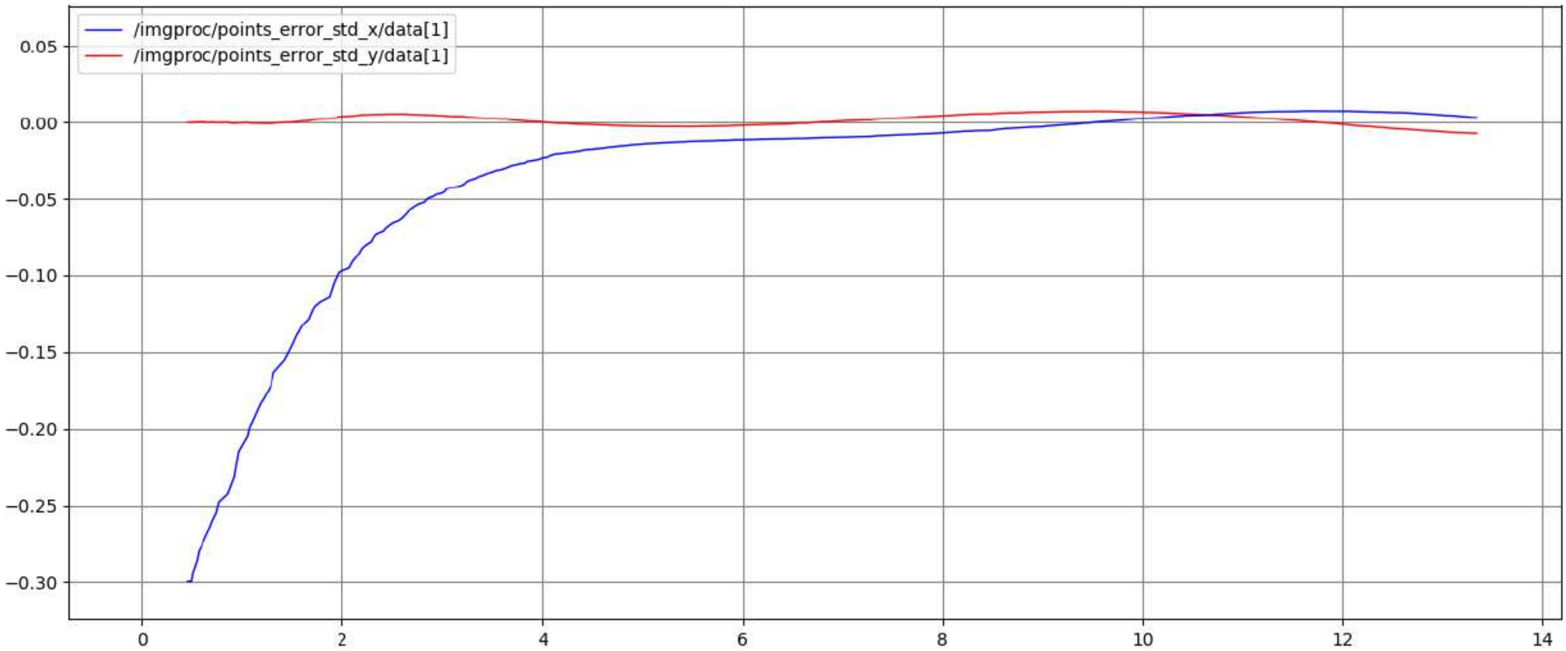
Exercise 2 - Visual Servoing - Ros

a)

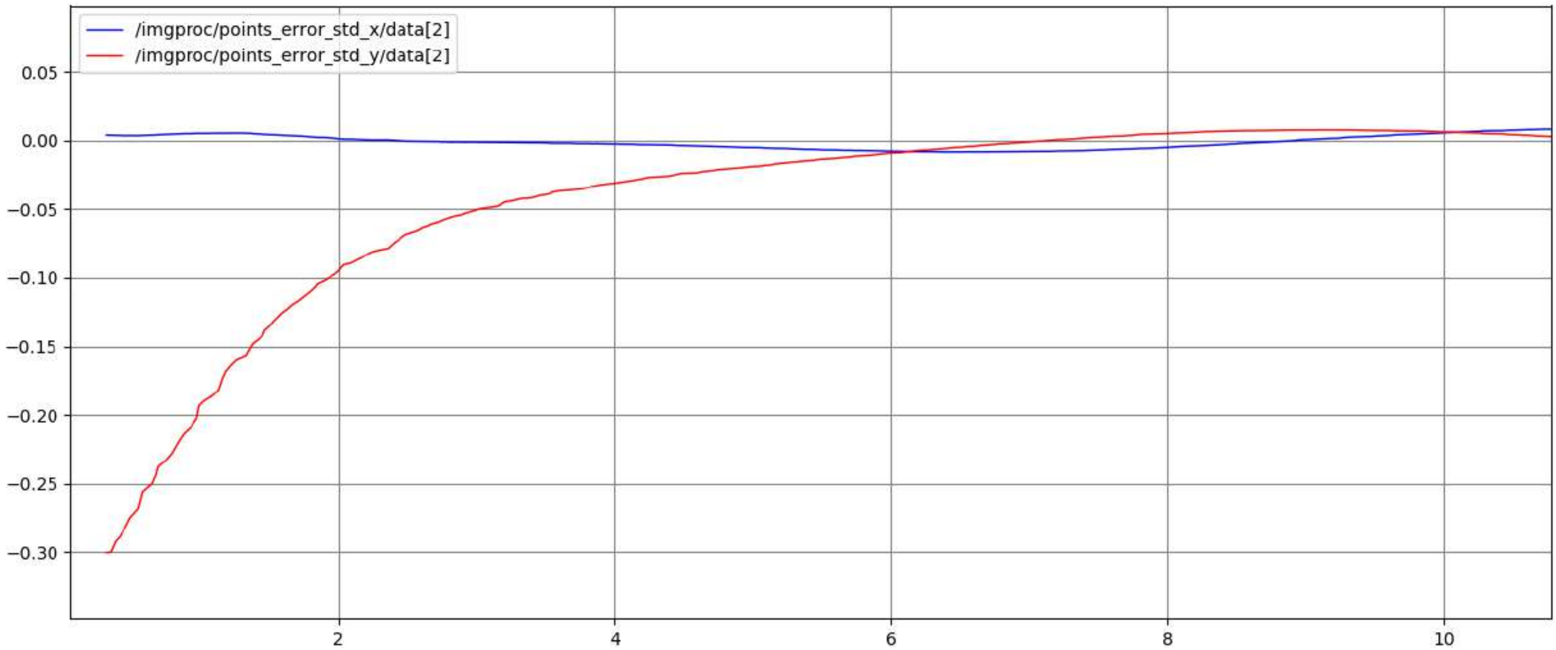
see code : https://github.com/Adrien9o2/TEK4030-A2

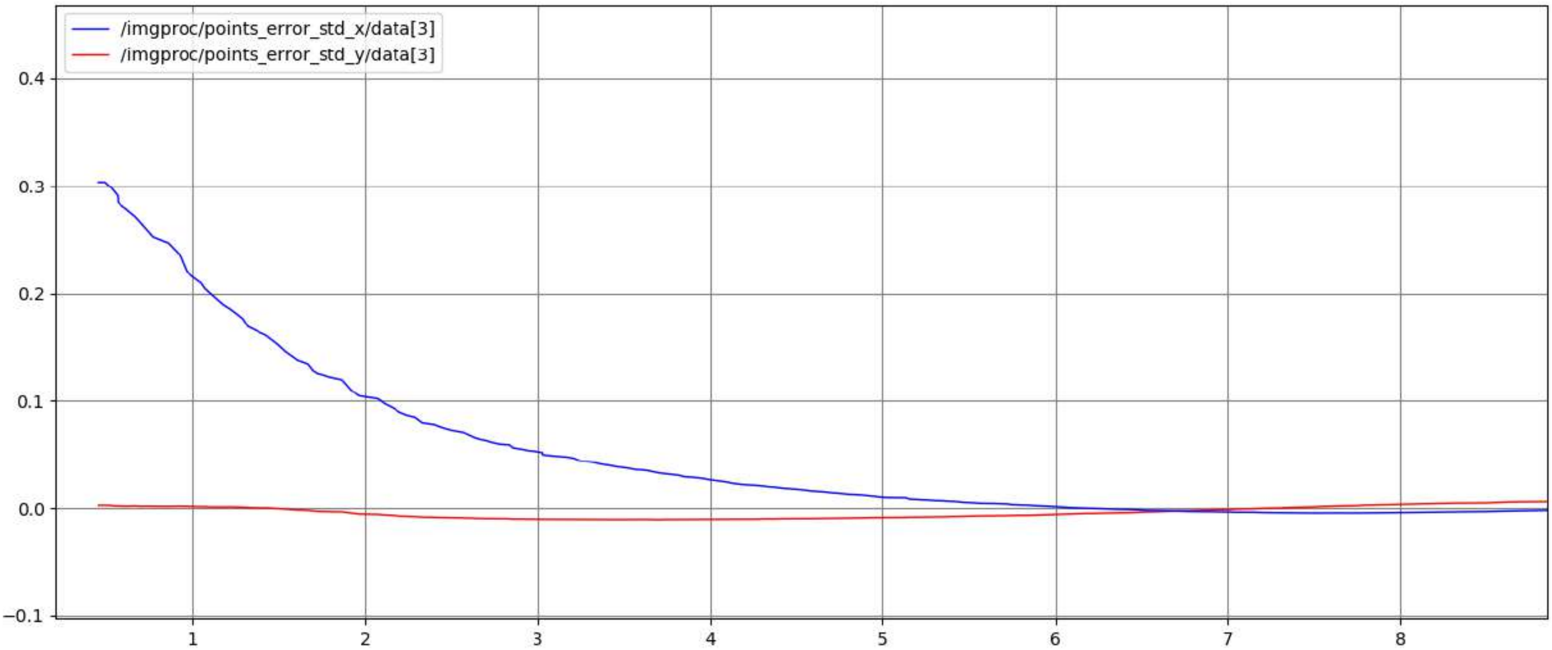
b)





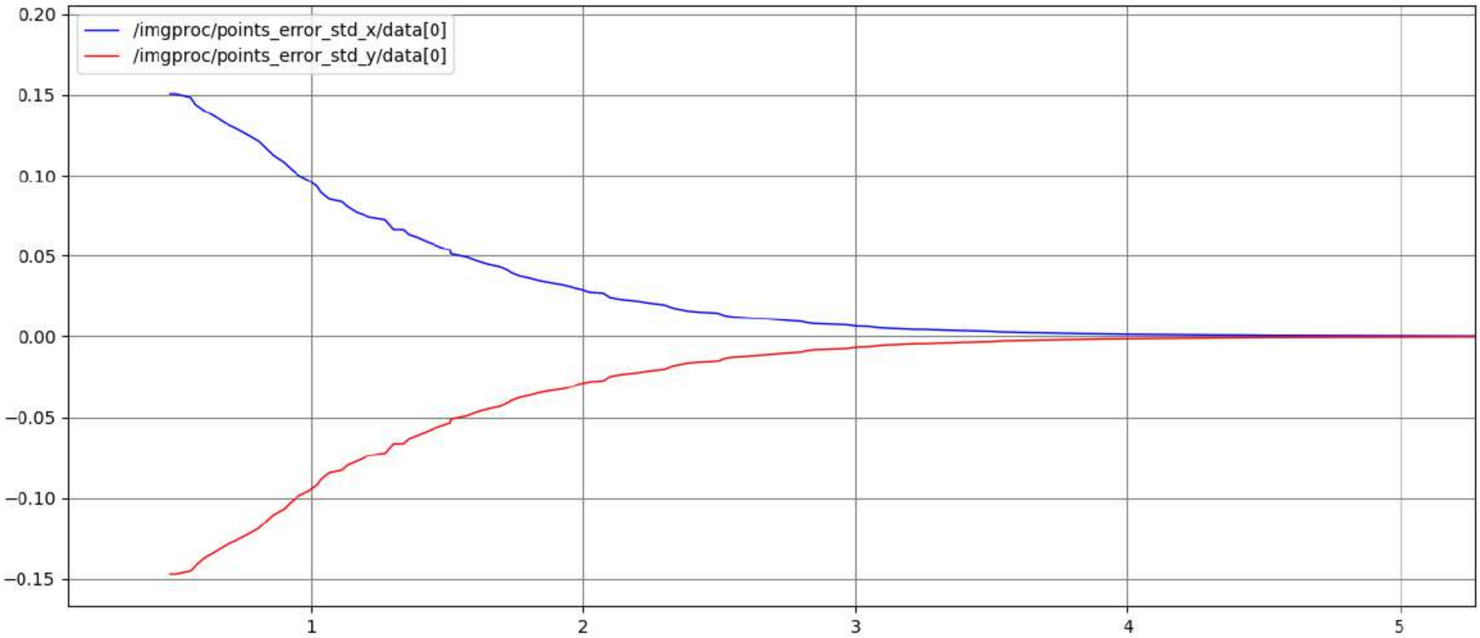
/imgproc/points_error_std_x/data[1]
/imgproc/points_error_std_y/data[1]
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
0
2
4
6
8
10
12
14

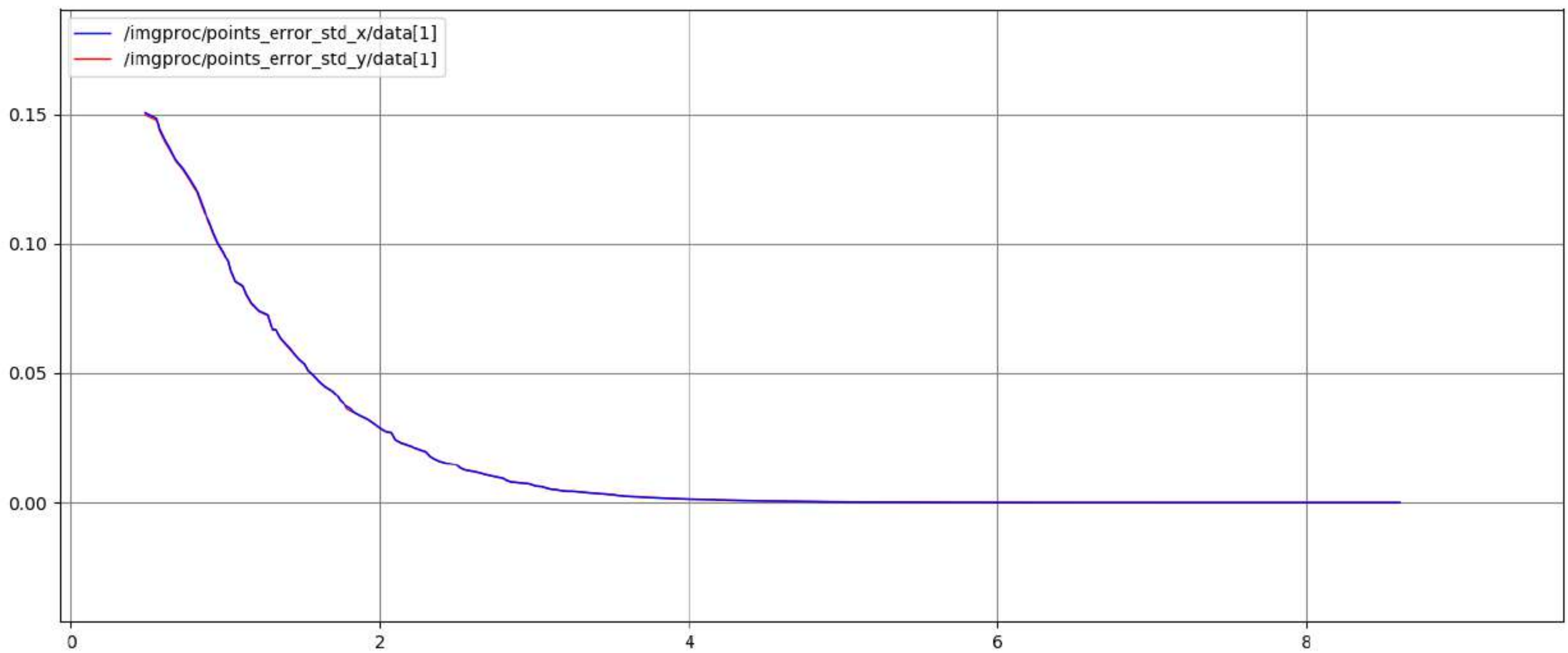

/imgproc/points_error_std_x/data[2]
/imgproc/points_error_std_y/data[2]
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
2
4
6
8
10

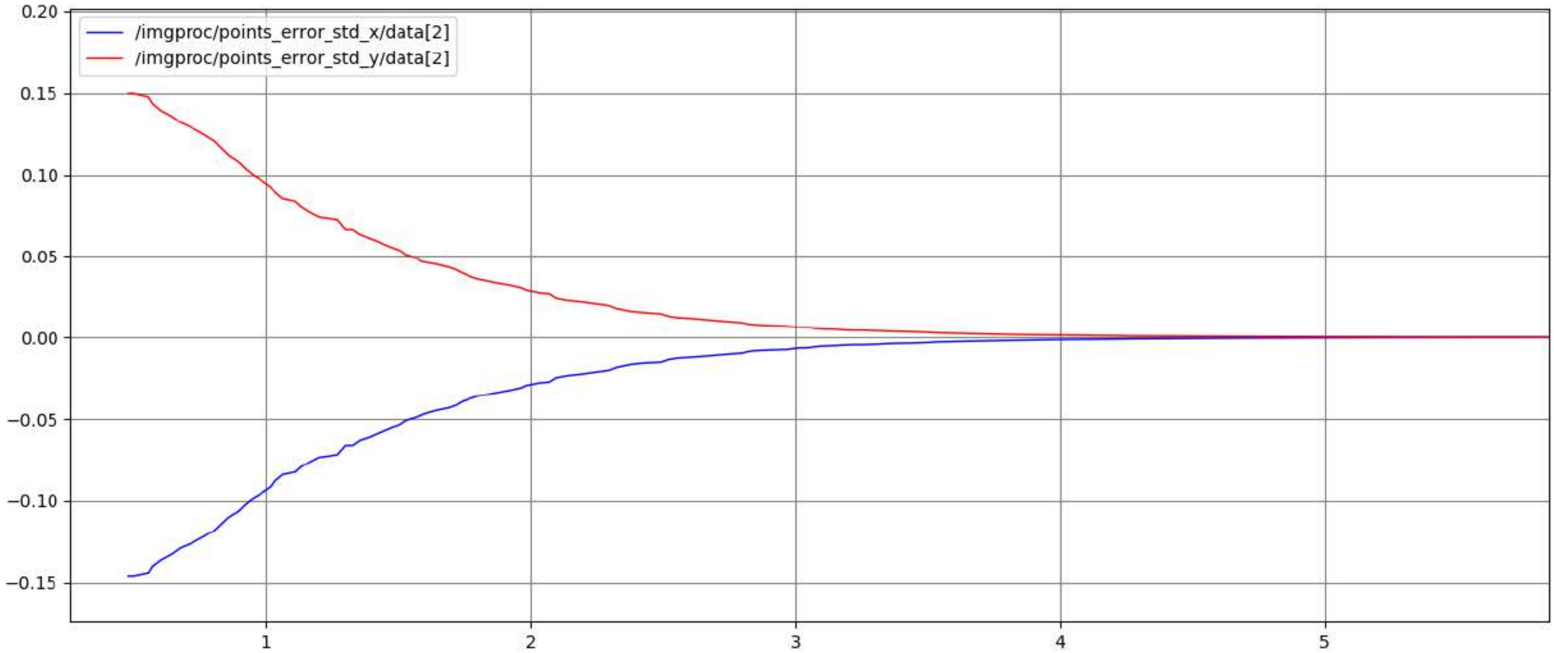


/imgproc/points_error_std_x/data[3]
/imgproc/points_error_std_y/data[3]
0.4
0.3
0.2
0.1
0.0
−0.1
1
2
3
4
5
6
7
8

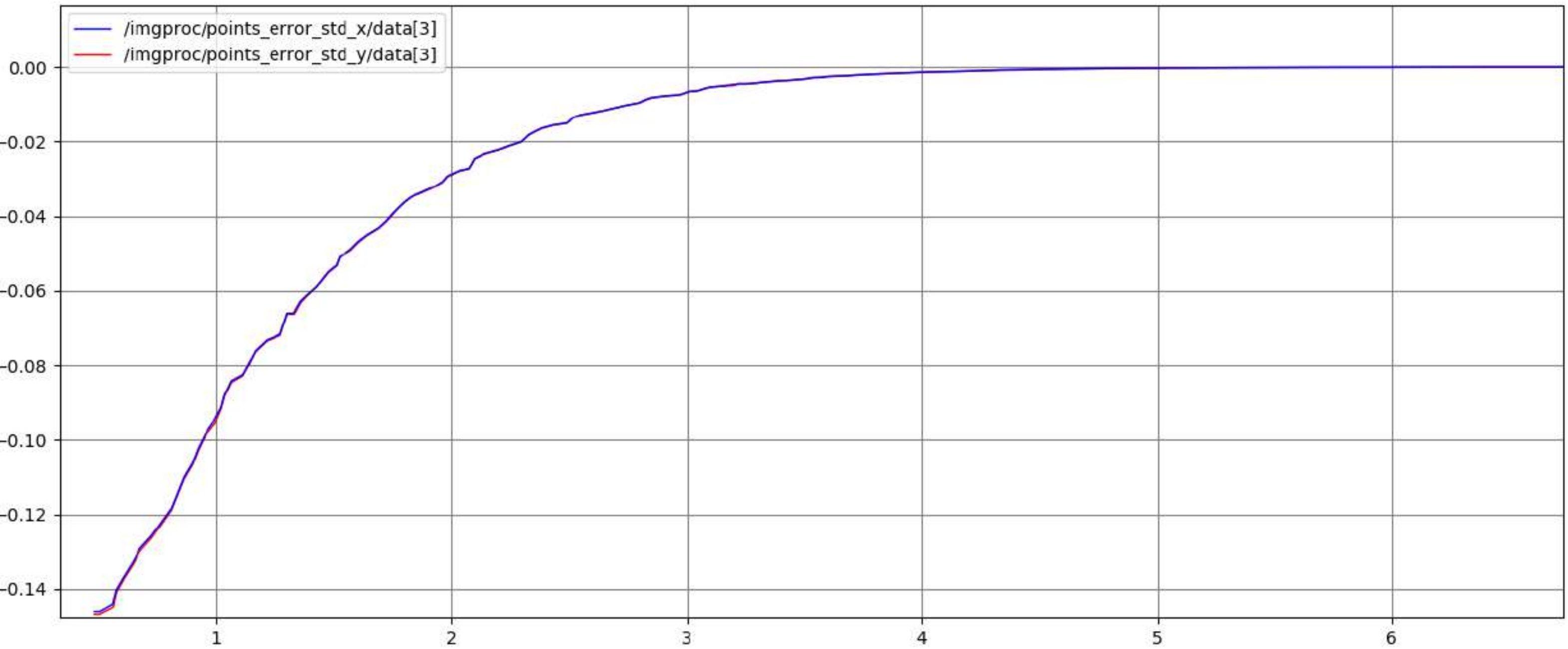
c)


/imgproc/points_error_std_x/data[0]
/imgproc/points_error_std_y/data[0]
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
1
2
3
4
5


/imgproc/points_error_std_x/data[1]
/imgproc/points_error_std_y/data[1]
0.15
0.10
0.05
0.00
0
2
4
6
8



/imgproc/points_error_std_x/data[2]
/imgproc/points_error_std_y/data[2]
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
1
2
3
4
5


/imgproc/points_error_std_x/data[3]
/imgproc/points_error_std_y/data[3]
0.00
-0.02
-0.04
-0.06
-0.08
-0.10
-0.12
-0.14
1
2
3
4
5
6

R1

Exercise 3 - Visual Servoing

a)

Position based visual servoing	Image based visual servoing
user operation space	User image space
Reconstruct target relative pose	compare current and desired image feature
pose estimation	no pose estimation
⇒ no direct control of image features ↳ may cause the object to exit the camera view	Keep the object within the camera field of view
Calibration error issues ↳ in feedback path	non linear mapping between image feature parameters and operational space
	↳ Calibration error as disturbance in forward path

b) ∘ Visual measurements have lower update rates than used in motion control of robot manipulator

∘ As a result the control :

$$u = g(q) + J_{Ad}^{T}(q, \tilde{x})(K_P \tilde{x} - K_D J_{AD}(q, \tilde{x})\dot{q})$$

in position based visual servoing

(PD with gravity compensation)

must have low K_P / K_D

A solution is to use Resolved velocity control with two levels of control

↳ - A high gain motion controller in joint space or operational space

- A visual servoing controller with lower update frequency

c) ~~The cnt~~

The interaction matrix describes the relationship between the velocity of the camera (held by the robot) and the image features

The image jacobian is the matrix such that

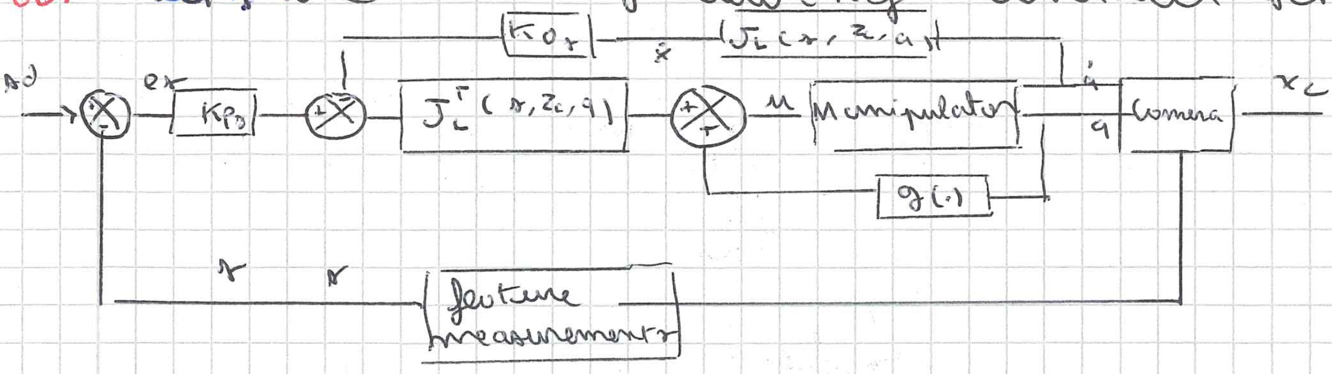
$$\dot{s} = J_s(s)\, v^c_{c,o}$$

that can be rewritten

$$\dot{s} = J_s v^c_o + \underbrace{L_s}_{\text{Interaction matrix}} v^c_c$$

$$L_s = J_s(s, T^c_o)\,\Gamma(o^c_{c,o})$$

d) Let's use the following control scheme:



23

Let's use the following Lyapunov function candidate

$$V(\dot{q}, e_r) = \frac{1}{2}\dot{q}^T B(q)\dot{q} + \frac{1}{2} e_r^T K_{p_r} e_r$$

Because B and K_p are both positive definite ; we have $V(0,0) = 0$;

we have $V(\dot{q}, e_r) > 0 \quad \forall \dot{q}, e_r \neq 0$

$$\dot{V}(\dot{q}, e_r) = \frac{1}{2}\Big[\underbrace{\ddot{q}^T B(q)\dot{q} + \dot{q}^T B(q)\ddot{q}}_{2\ddot{q}^T B(q)\dot{q}} + \dot{q}^T \dot{B}(q)\dot{q} + \underbrace{\dot{e}_r^T K_{p_r} e_r + e_r^T K_{p_r}\dot{e}_r}_{= 2\dot{e}_r^T K_p e_r} \Big]$$

(because $K_p > 0$ symmetric)

using $B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) = u$;

$$\ddot{q}^T B(q) = (B(q)\ddot{q})^T$$
$$= (u - C\dot{q} - F\dot{q} - g(q))^T$$

$$\hookrightarrow \dot{V}(\dot{q}, e_r) = \frac{1}{2}\Big[2\ddot{q}^T B(q)\dot{q} + \dot{q}^T \dot{B}\dot{q} + 2\dot{e}_r^T K_p e_r \Big]$$

$$= \frac{1}{2}\Big[2(u - C\dot{q} - F\dot{q} - g(q))^T\dot{q} + \dot{q}^T\dot{B}\dot{q}\Big] + \dot{e}_r^T K_p e_r$$

$$= \underbrace{(u - g)^T\dot{q}}_{= \dot{q}^T(u-g)} + \cancel{\tfrac{1}{2}}\ \underbrace{-(F\dot{q})^T\dot{q}}_{-\dot{q}^T F\dot{q}} + \dot{q}^T\underbrace{(\dot{B} - 2C)}_{0\ (7.49)}\dot{q} + \dot{e}_r^T K_p e_r$$

$$= \dot{q}^T(u-g) - \dot{q}^T F\dot{q} + \dot{e}_r^T K_p e_r$$

24

$$\dot V = -\dot q^T F \dot q + \dot q^T (u - g(q)) + \dot e_r^T K_{p_r} e_r$$

using $u = g(q) + J_L^T(r, r_c, q)(K_{p_r} e_r - K_D J_L(r, r_c, q)\dot q)$

$\left(J_L = L_r \begin{bmatrix} R_c^T & 0 \\ 0 & R_c^T \end{bmatrix} J(q)\right)$ leads to :

$$\dot V = -\dot q^T F \dot q + \dot q^T [g(q) - g(q) + J_L^T [K_{p_r} e_r - K_D \underbrace{J_L \dot q}_{-\dot e_r}]]$$
$$+ \dot e_r^T K_{p_r} e_r$$

since $\dot r_d = 0$ (constant input)

$$\dot e_r = -\dot r = -J_L \dot q$$

$$\Rightarrow \quad \dot V = -\dot q^T F \dot q + \dot q^T [J_L^T [K_{p_r} e_r + K_D \dot e_r]]$$
$$+ \dot e_r^T K_{p_r} e_r$$

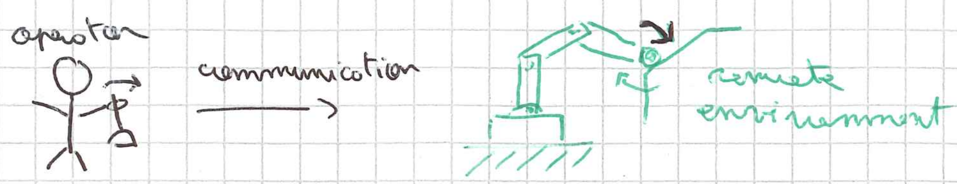
$$\dot V = -\dot q^T F \dot q + \overbrace{(J_L \dot q)}^{-\dot e_r}{}^T [K_{p_r} e_r + K_D \dot e_r]$$
$$+ \dot e_r^T K_{p_r} e_r = 0$$

$$= \underbrace{-\dot q^T \underbrace{F}_{>0} \dot q - \dot e_r^T \underbrace{K_D}_{>0} \dot e_r}_{<0 \quad \forall (\dot q, e_r) \neq (0,0)}$$

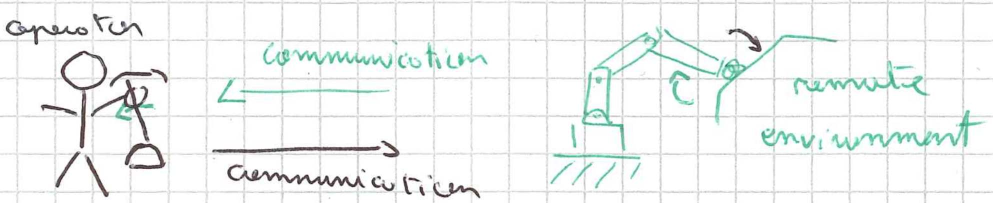
$\hookrightarrow$ stable

Exercise 1 - Tele-operations

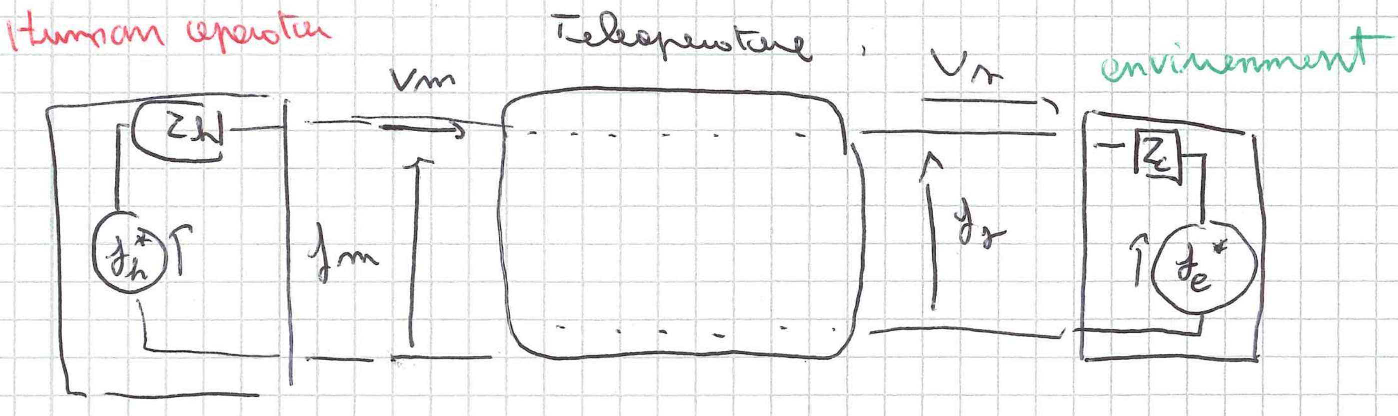
a) • unilateral teleoperation



Bilateral teleoperation



b) considering the following quadrupole (teleoperator



the hybrid matrix is defined such that

$f \rightarrow$ force

$v \rightarrow$ speed

$$\begin{bmatrix} f_m \\ -v_r \end{bmatrix} = H \begin{bmatrix} V_m \\ f_r \end{bmatrix}$$

$Z \rightarrow \frac{f}{v}$

the ideal / transparent teleoperation:

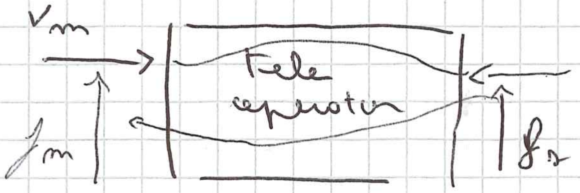
$$f_m = f_r$$
$$v_m = v_r$$
$$\Rightarrow H = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

26

c)

The forward flow controller is a (common) bilateral tele-operation

a force sensor is needed on the slave
contact force (get f_s)



velocity is sent from master to slave
and force is sent from slave to master