Robust Control of Mechatronic Systems **(Due : 18 October 2020)**

TP Embedded Control of an Electric Steering Wheel

Professor: John-Jairo Martinez-Molina *Students:*Raquel Tirach, Lukas Schild, Gia Quoc Bao Tran

Objective :
The objective of the TP is to control the angular position of a motorized system with an incremental encoder. The quantization noise produced by the encoder as well as the couples resistors (unknown) acting on the motor shaft will have to be considered when synthesizing the control laws. In particular, the resistive torque should be estimated and then used to control the angular position of a steering wheel.

1 System description

The system consists of an Arduino board connected to power electronics elements that supply voltage and current to a DC motor. To increase the motor torque, a reducer (*torque amplifier*) is used and connected to the wheel whose position we want to control. To measure the angular position, we have an incremental sensor.

2 Physical model description

2.1 System modelling

The mechanical equation is $J\dot{\omega} = \tau_e + \tau_{fric}$
where $\tau_e = K_c I = K_e I$ and $I = \frac{V-E}{R}$ (inductance neglected).
Therefore, $J\dot{\omega} = \frac{K_e}{R}(V - E) + \tau_{fric}$.
And since $E = K_e\omega$, the system can be modeled as follows :

$$\boxed{J\dot{\omega} = \frac{K_e}{R}(V - K_e\omega) + \tau_{fric}}$$

2.2 State-space representation

The following state-space representation has been chosen in order to model the mechanical system as described above, with $x_1 = \theta$, $x_2 = \dot{\theta} = \omega$, $y = x_1 + \eta$ and $J\dot{x_2} = \frac{K_e}{R}(V - K_e \cdot x_2) + \tau_{fric}$.
This gives the following system :

$$\boxed{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\frac{K_e^2}{R \cdot J} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \frac{K_e}{R \cdot J} & 1 \end{pmatrix} \begin{pmatrix} V \\ \tau_{fric} \end{pmatrix},\ y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \eta}$$

The two disturbances are τ_{fric} (input) and η (measurement noise). The reducer's ratio is 53 and the encoder has 12 points. So η is bounded by $\frac{2\pi}{12 \cdot 53} = 0.0099$ which is a high resolution for the encoder.

3 Synthesis of an extended system state observer

So we have two disturbances where one is already bounded. We want to construct an estimator of the resistance torque based on an optimal state observer.
We assume that the disturbance torque is slowly varying :
$\dot{\tau}_{fric} = -c\tau_{fric}$ (5) with for example $c = 1e-9$

We are going to define an extended state and establish a state representation :

$$\begin{cases} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\tau}_{fric} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\frac{K_e^2}{R \cdot J} & \frac{1}{J} \\ 0 & 0 & -c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \tau_{fric} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{K_e}{RJ} \\ 0 \end{pmatrix} \mathrm{u} \\ \mathrm{y} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \tau_{fric} \end{pmatrix} + \eta \end{cases}$$

3.1 Discretization of the system

In order to discretize the defined system, the Matlab command `c2d()` has been used. It takes as input both the extended system as well as the sample period and gives the following discrete system, where $T_e = 0.01s$.

$$A_d = \begin{pmatrix} 1 & 0.0095 & 0.0071 \\ 0 & 0.9071 & 1.394 \\ 0 & 0 & 0.99 \end{pmatrix}, \; B_d = \begin{pmatrix} 0.0010 \\ 0.1978 \\ 0 \end{pmatrix}, \; C_d = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \text{ and } D_d = 0$$

3.2 State observer

We want a state observer that allows to estimate the complete state x. The gain of the L_{obs} observer is obtained from the solution of an optimization problem of the Kalman filter type (dual LQR problem) :

```
W = [0.1 0 0; 0 0.1 0; 0 0 1];
V = 0.01;
Lobs = dlqr(Ad',Cd',W,V)';
```

The choice on W and V ...
In order to test the observer for zero voltage, we first tested it in Matlab with the following commands :

```
Xinit = [1 1 1]';
sys_obs = ss(Ad-Lobs*C, B*0, eye(3), zeros(3,1));
initial(sys_obs, Xinit)
```

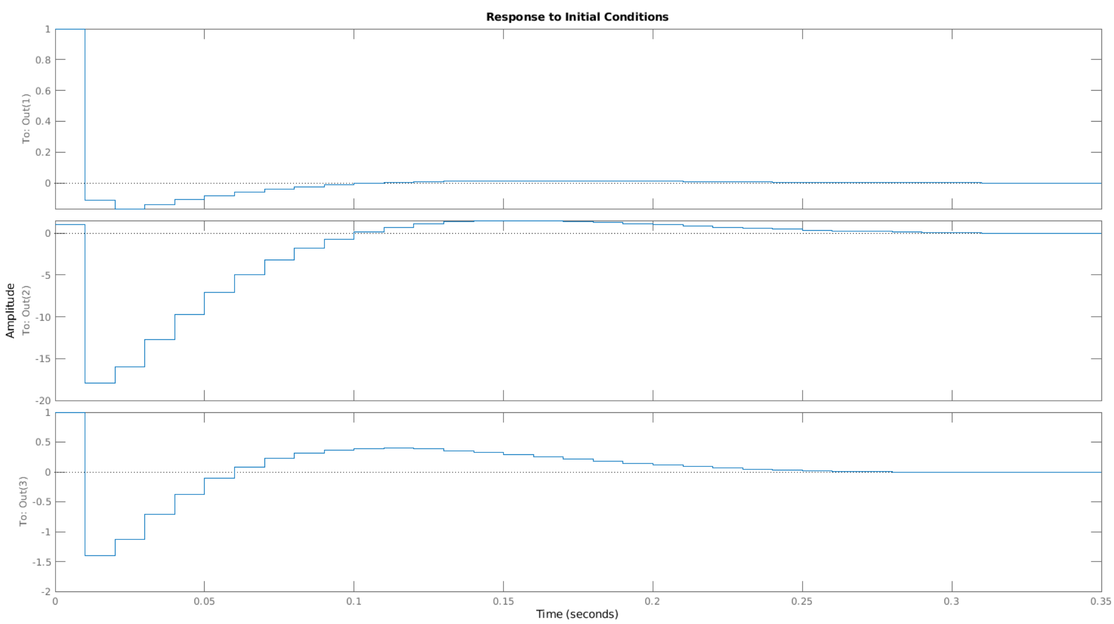

FIGURE 1: Initial function with $V = 0$.

The observer is correctly estimating the states and quickly converging.

3.3 Coding and testing of the observer

In order to test the constructed observer, we implemented an algorithm which uses the observers characteristics to calculated the estimated state of the system. The following `C` code has been used to test the observer. While testing the motor voltage has been set to **0**.

```c
// Constants taken from Matlab
void process() {

// Get measured angular position
angPos = ticks*0.0099

// Update estimated states
omegan = Ad1*omega + Ad2*theta + Ad3*tauFric + Bd1*V - Lobs1*omega + Lobs1*angPos;
thetan = Ad4*omega + Ad5*theta + Ad6*tauFric + Bd2*V - Lobs2*omega + Lobs2*angpos;
tauFric = Ad7*omega + Ad8*theta + Ad9*tauFric + Bd3*V - Lobs3*omega + Lobs3*angPos;

// Update state variables
omega = omegan;
theta = thetan;
tauFric = tauFricn;
}
```

The observer works as expected, which verifies the calculations made to construct the observer as well as the implementation in `C`. Figure 2 shows the output obtained by the serial communication with the Arduino. The X-axis is showing the time in number of reading periods, where one period equals $10ms$.

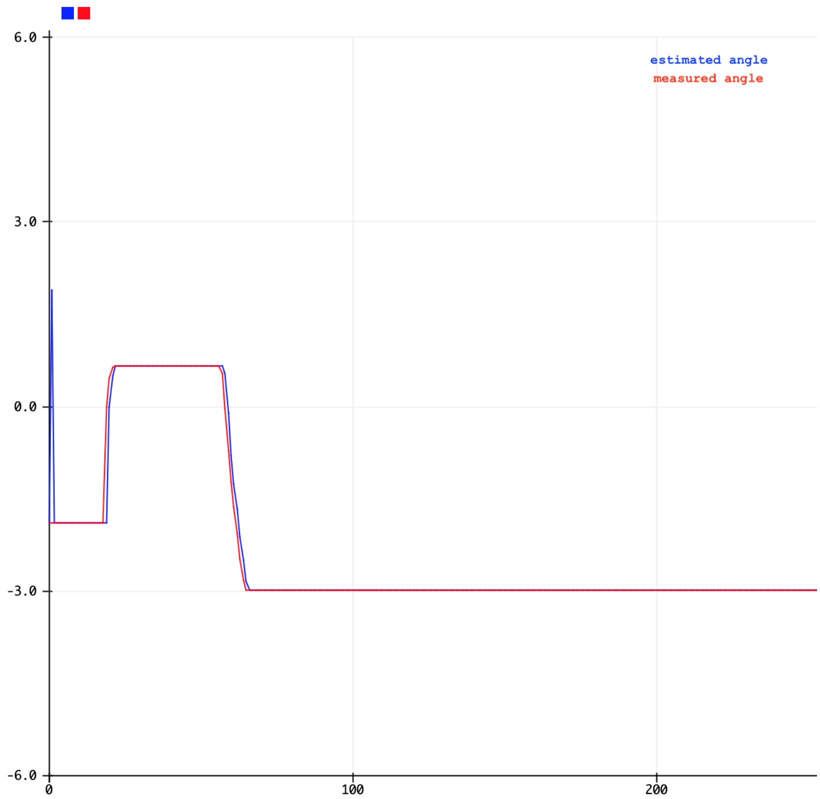

FIGURE 2: Output Arduino

CASE 1 : Angular position control

State-feedback control

First, we compared across combinations of the matrices Q and R to find the best one. The values are $Q_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 10 \end{pmatrix}$, $Q_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $R_1 = 1$, $R_2 = 0.01$. The four combinations are (Q_1, R_1), (Q_2, R_1), (Q_1, R_2), and (Q_2, R_2). To simulate the system's response, we used :

```
time = [0 : Ts : 500*Ts]';
r = [ones(length(time), 1) zeros(length(time), 1) zeros(length(time), 1)];
lsim(ss(sysD.a - sysD.b*F, sysD.b*F, sysD.c, sysD.d, Ts), r, time);
set(gca, 'FontSize', 11);
grid on;
```

Here r is the reference where we used the constant value 1 for the angle and 0 for the two other ones. Attention has to be paid to the new B matrix as :

$$\dot{x} = Ax + Bu = Ax + B(-Fx + Fr) = (A - BF)x + BFr$$

That explains the B we used in the code. The results of simulation are as follows :

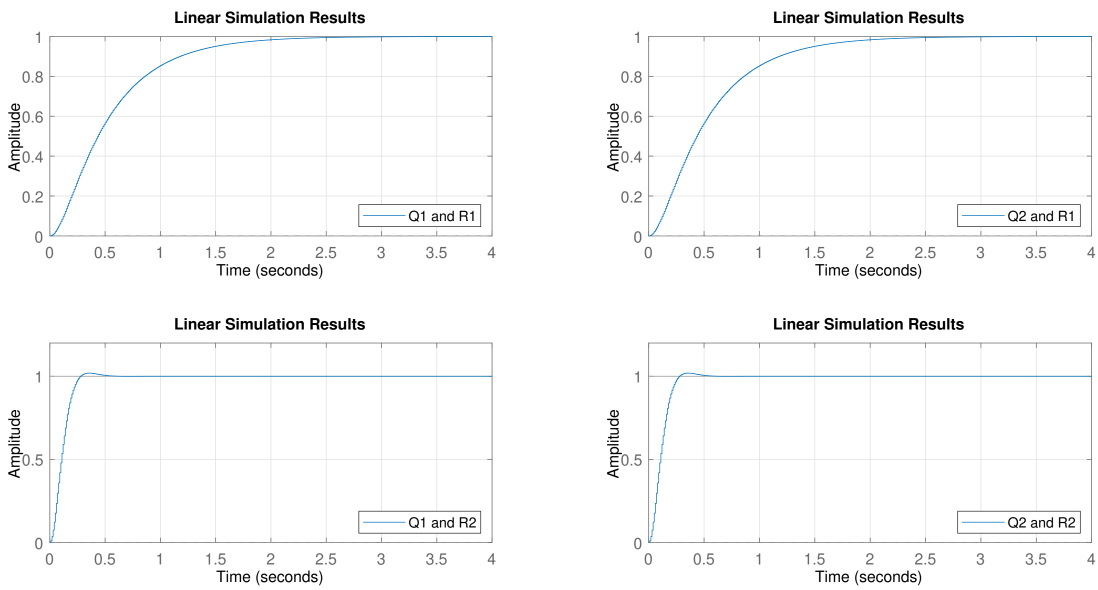

FIGURE 3: Comparison across different choices of the weight matrices Q and R.

We chose the set (Q_1, R_2) and executed the code :

```
Q = [1 0 0; 0 0 0; 0 0 10];
R = 0.01;
F = dlqr(sysD.a, sysD.b, Q, R);
```

The result is :

$$F = \begin{pmatrix} 9.3788 & 0.5983 & 7.0851 \end{pmatrix}$$

In order to test the constructed observer on the experimental system, we used the following `C` code.

```
//Reading the potentiometer value
  int pot = analogRead(pinPot);
//Angels
 angposNew = ticks*0.0099 ; //angle mesur en ticks et puis convertit en radians
 angRef = 3.14159*(pot-2048)/2048

//Commande par retour d'tat
```

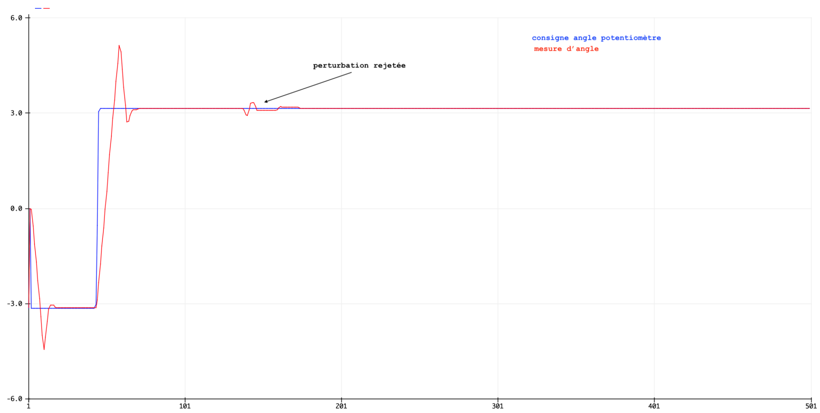

FIGURE 4: Observer State Feedback on the Arduino

```
commande = - F1*(xd1 - angRef) - F2*xd2 - F3*Tdd;

//Observateur avec commande par retour d'tat
xdd1 = Ad1*xd1 + Ad2*xd2 + Ad3*Td + Bd1*commande - Lobs1*xd1 + Lobs1*angposNew;
xdd2 = Ad4*xd1 + Ad5*xd2 + Ad6*Td + Bd2*commande - Lobs2*xd1 + Lobs2*angposNew;
Tdd = Ad7*xd1 + Ad8*xd2 + Ad9*Td - Lobs3*xd1 + Lobs3*angposNew;

//Incrementation
xd1 = xdd1;
xd2 = xdd2;
Td = Tdd;
```

The result of running the above code on the Arduino and testing the observer on the experimental system is being shown in figure 4.

Improvement : Integral action

Since the disturbance torque is already being estimated, there is no need to add an integral action. Indeed, the estimation of the torque already implies that disturbances are being rejected.

Limits of CASE 1

There are some limits for this case :
- The maximum electrical motor torque is $\tau_{max} = \frac{K_e \cdot V_{max}}{R} = \tau_{max} \frac{R}{K_e^2} = 10.87rad/s$
- The maximum angular speed is when the angular acceleration becomes 0 and no friction disturbance $\omega_{max} = \frac{\tau_{max}}{R} =_{max}= \frac{0.46 \cdot 5}{3.3} = 0.7Nm$
- In terms of energy, some is lost due to friction : $(J\dot{\omega}) \times \omega = (\tau_{fric} + \tau_{fric}) \times \omega => J\dot{\omega}\omega = \tau_{fric}\omega + \tau_{fric}\omega$ and this means $P_{store} = P_{give} + P_{lost}$ (P_{lost} is negative)

CASE 2 : Electrical assistance system

Simulation of the electrical assistance system

In this configuration, the system is designed to assist humans by producing a torque proportional to the (estimated) human torque. We define a constant α and the control objective is to make the wheel's torque α times the human torque estimated using a state observer.
First, an attempt was made to simulate the system. We would like to simulate both the observation and tracking by using :

$$\dot{x} = Ax + Bu$$
$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly$$

where u is the control input (the voltage). As our objective is provide assistance to the human, we would impose :

$$v = \frac{\alpha R}{K_e}\hat{\tau}_{human} + K_e\beta\dot{\hat{\theta}} => u = F_a\hat{x}$$

where $F_a = \begin{pmatrix} 0 & K_e\beta & \frac{\alpha R}{K_e} \end{pmatrix}$ denotes a constant gain. We defined an extended system of which the outputs are the observation errors (the first three outputs) and the tracking error (the last one, defined by the difference between the real torque and the desired one). We have (there are three states in each of x and $\hat{x}$) :

$$\begin{pmatrix} \dot{x} \\ \dot{\hat{x}} \end{pmatrix} = \begin{pmatrix} A & BF_a \\ LC & A - LC + BF_a \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix}$$

$$y = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & \alpha & 0 & -\beta\frac{K_e^2}{R} & -\alpha \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix}$$

All the four outputs should converge to zero, because they are the errors. Simulation using MATLAB gave the following results :

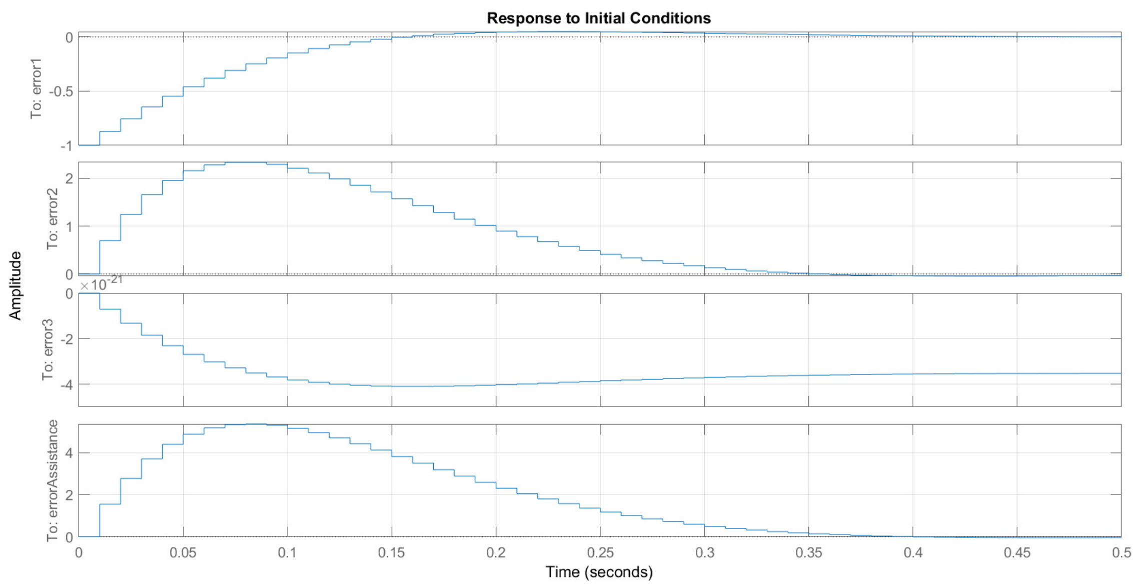

FIGURE 5: Simulation of the electrical assistance system.

And according to this simulation, all the output converged to zero (except the third one which converged to a very small non-zero value of order 10^{-21}, which is basically zero), given non-zero initial conditions. We must also note that there are constraints on the values of α and β. Indeed, β has to stay between 0 and 1, and α has to stay between 0 and the max value for which the voltage stays between -5V and +5V.

Coding and testing of the electrical assistance system

```
angposNew = ticks*0.0099 ; //angle mesur en ticks et puis convertit en radians

//Assistance electrique
 float alpha = 0.8; //between zero and the voltage has to stay between -5 and +5V!
 float beta = 0.5; //between zero and one

 V = R*(alpha*Td+beta*Ke*Ke*xd2/R)/Ke;

 //Observateur avec commande
 xdd1 = Ad1*xd1 + Ad2*xd2 + Ad3*Td + Bd1*V - Lobs1*xd1 + Lobs1*angposNew;
 xdd2 = Ad4*xd1 + Ad5*xd2 + Ad6*Td + Bd2*V - Lobs2*xd1 + Lobs2*angposNew;
 Tdd = Ad7*xd1 + Ad8*xd2 + Ad9*Td - Lobs3*xd1 + Lobs3*angposNew;

//Incrementation
 xd1 = xdd1;
 xd2 = xdd2;
```

```
 Td = Tdd;

//Torques
 Thuman = Td+Ke*Ke*xd2/R;
 Tvoltage = Ke*V/R;
```

Observations : The model produces a torque in the same direction as the human, which acts as a support to the human induced turning force. We can also observe that in the time between the human stopping to apply force and the end of the rotation, the system keeps providing support for a short amount of time, the time it takes the observer to adapt.

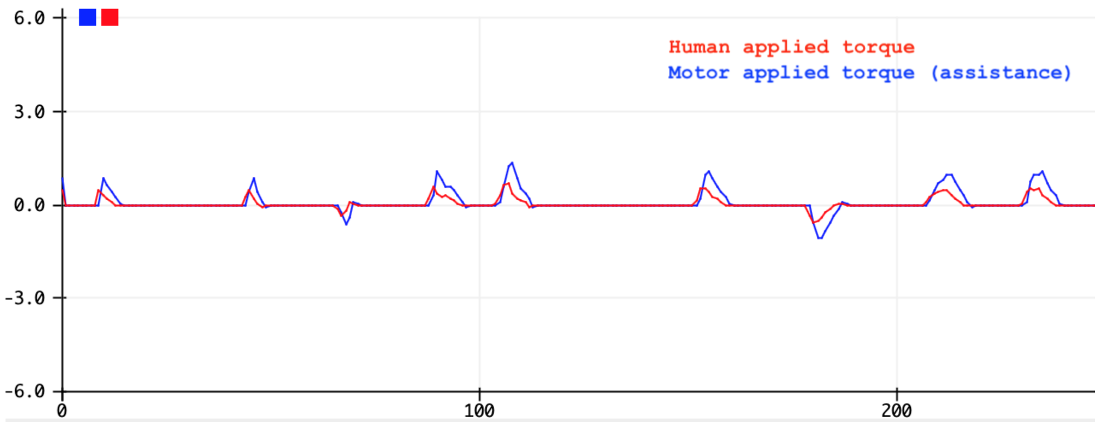

FIGURE 6: Verification on the real mechanical system

Figure 6 shows the torques observed during an experiment on the real mechanical system. The blue curve, which is the total amount of torque the motor produces, includes the human induced torque as well as the torque added by the system. The curve follows the red curve, which indicates the estimated human applied torque. The observed difference between the red and the blue curve is the amount of torque applied to assist the human input.

Limits of CASE 2

— The assistance applied can not be a lot stronger than the torque applied by the user, because then the assistance would be highly sensitive and the system would become unstable.

— The limit of the assistance is reached once the motor control tries to apply more than $+/-5V$ to the motor, which causes an actuator saturation.

Conclusion :

After this practical work session, we understood better the methods of state observation and state-feedback control on a real mechanical system.
We also understood better the advantages of a state observer since we were able to control a steering wheel and add assistance to it using only one sensor (relative position in ticks).