Physical Human-Robot Interaction Project

Master's degree in Computer Engineering for Robotics and Smart Industry

ACADEMIC YEAR: 2020-2021

Edoardo Fiorini

March 23, 2022

Summary

1 Four-channel bilateral teleoperation architecture

In the first assignment we implemented the Single-Input Single-Output Four-channel bilateral teleoperation architecture. In this scenario we have the following assumptions:

- A1: No communication delays
- A2: Perfect knowledge of the master/slave robot dynamics
- A3: Force and velocity (position) measurements are available

In the schema, m is for the master and s for the slave. Everything has been implemented in Matlab/Simulink, here we have reported only the schema, parameters and achieved results. Moreover, we added to the plant Z_m^{-1} and Z_s^{-1} a damping, but we did not see particular difference. The behavior should be more rigid, due to the parameters we cannot get this point. The blocks have the following shape:

$$
\begin{aligned}
C_m &= B_m + \frac{K_m}{s} & \qquad C_s &= B_s + \frac{K_s}{s} \\
Z_m^{-1} &= \frac{1}{M_m s} & \qquad Z_s^{-1} &= \frac{1}{M_s s} \\
C_1 &= Z_s + C_s & \qquad C_3 &= I \\
C_2 &= I & \qquad C_4 &= -(Z_m + C_m)
\end{aligned}
$$

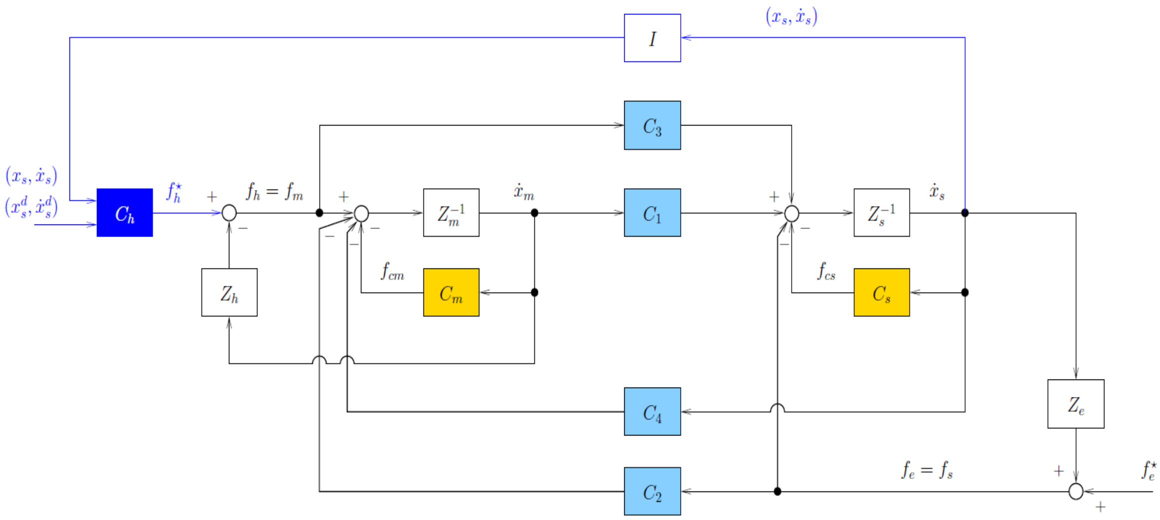

Figure 1: Four-channel bilateral teleoperation architecture scheme

$F_c = 2\pi$	$P_h = 13000$	$B_m = 100$	$B_s = 100$	$J_h = 1$	$J_e = 0$	$D_m = 0$
$A = 1$	$D_h = 700$	$k_m = 10$	$k_s = 10$	$B_h = 70$	$B_e = 0$	$D_s = 0$
$env = 0.1$		$M_m = 0.5$	$M_s = 2$	$K_h = 2000$	$K_e = 200$	

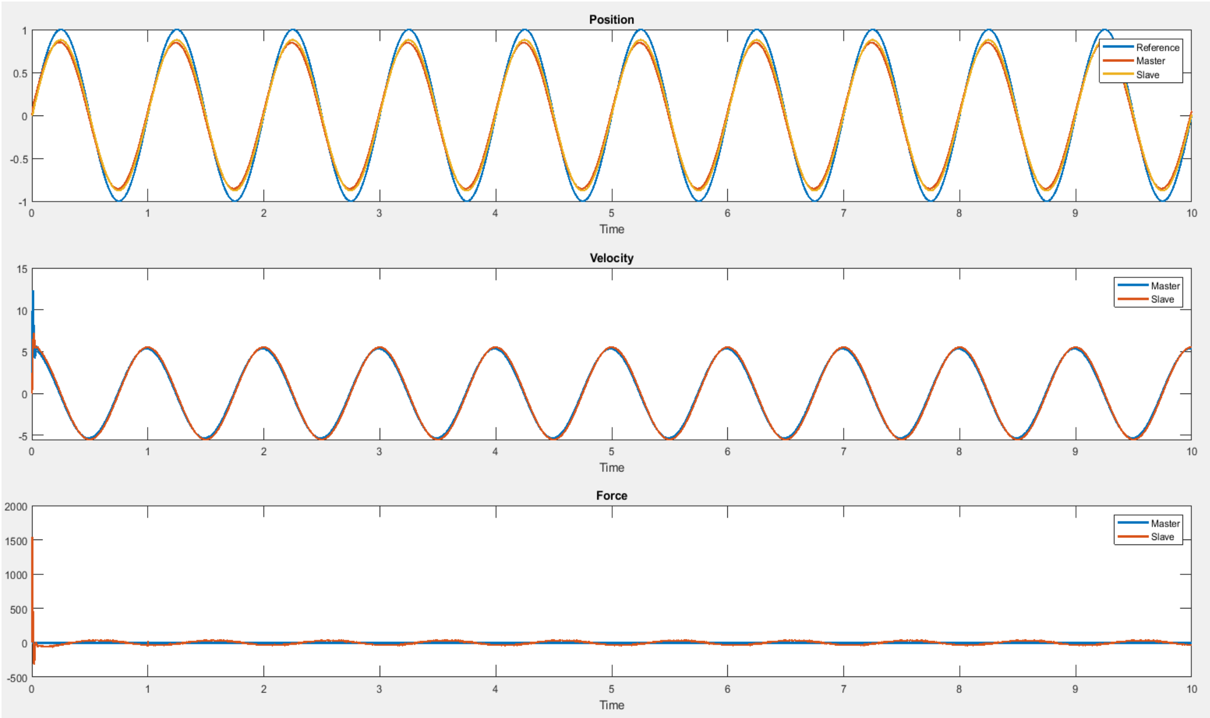

Figure 2: Four-channel bilateral teleoperation architecture results in free motion

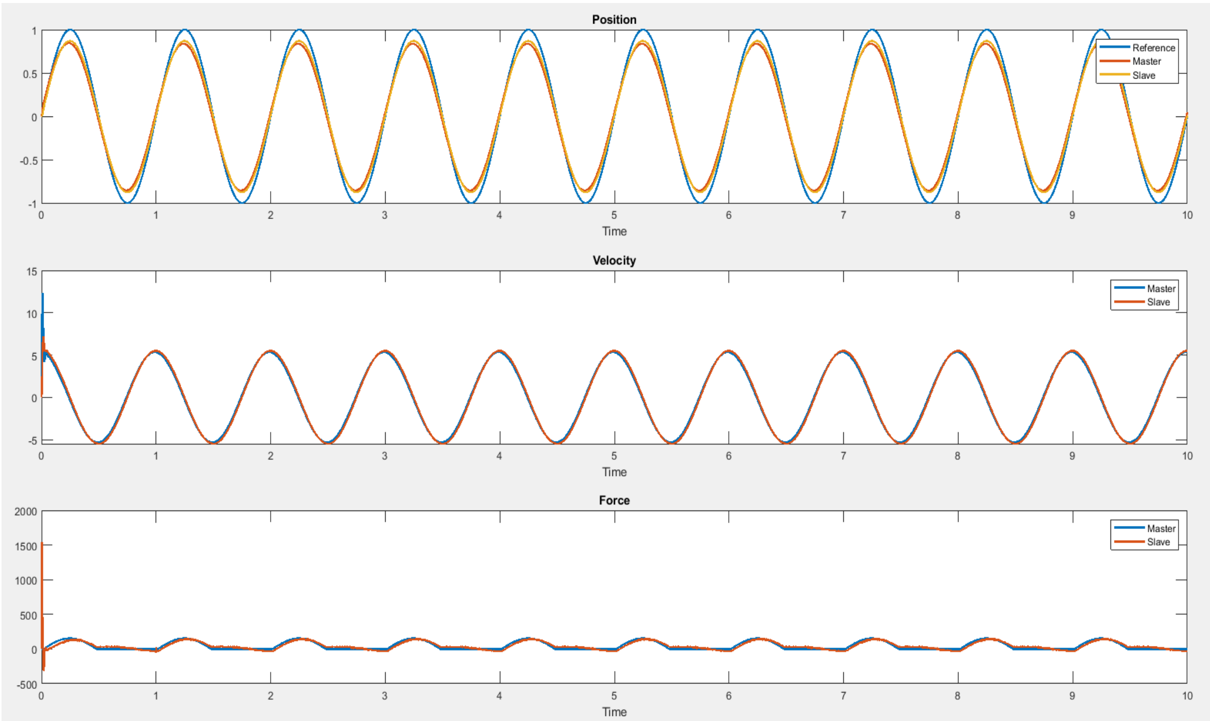

Figure 3: Four-channel bilateral teleoperation architecture results in contact

2 Four-Channel bilateral teleoperation architecture with local force feedback

In the second assignment we have to implement the Four-Channel bilateral teleoperation architecture with local force feedback, the schema is shown in 4 . Same parameters as in the previous section, but C_{mf} and C_{sf} are chosen as constants. The main difference from the previous architecture is the possibility to focus better on position or force in relation of the stiffness of the environment. Everything has been implemented in Matlab/Simulink, here we have reported only the schema, parameters and achieved results.

$$
\begin{aligned}
C_m &= B_m + \frac{K_m}{s} & \qquad C_s &= B_s + \frac{K_s}{s} \\
Z_m^{-1} &= \frac{1}{M_m s} & \qquad Z_s^{-1} &= \frac{1}{M_s s} \\
C_1 &= Z_s + C_s & \qquad C_3 &= 1 + C_{sf} \\
C_2 &= 1 + C_{mf} & \qquad C_4 &= -(Z_m + C_m)
\end{aligned}
$$

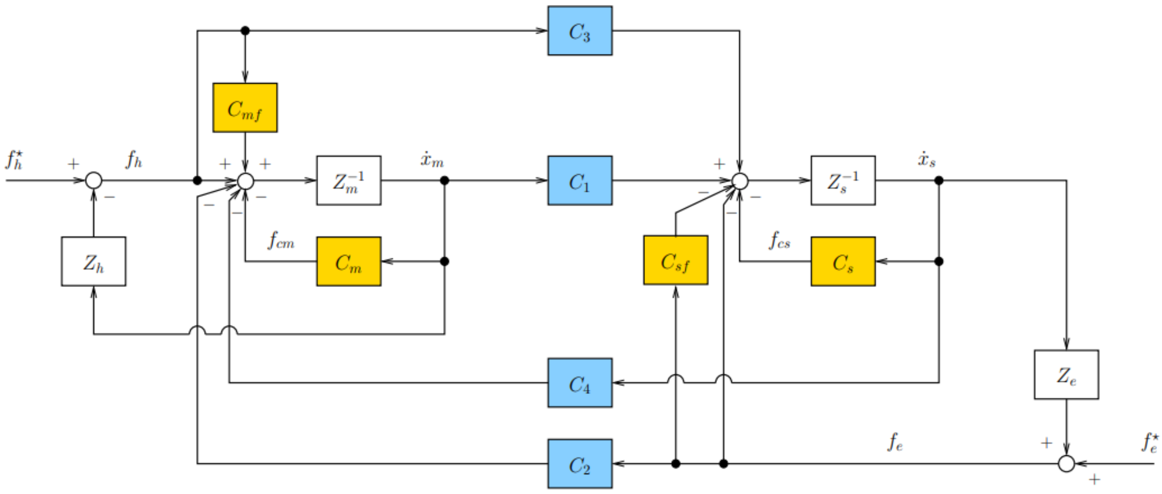

Figure 4: Four-channel bilateral teleoperation architecture with local force feedback scheme

$$
\begin{array}{ccccccc}
\begin{gathered} F_c = 2\pi \\ A = 1 \\ env = 0.1 \end{gathered} &
\begin{gathered} P_h = 20000 \\ D_h = 100 \end{gathered} &
\begin{gathered} B_m = 144 \\ k_m = 180 \\ M_m = 0.5 \\ C_{mf} = 0.6 \end{gathered} &
\begin{gathered} B_s = 6.4 \\ k_s = 8 \\ M_s = 2 \\ C_{ms} = 0.6 \end{gathered} &
\begin{gathered} J_h = 0.5 \\ B_h = 70 \\ K_h = 2000 \end{gathered} &
\begin{gathered} J_e = 0 \\ B_e = 10 \\ K_e = 20 \end{gathered} &
\begin{gathered} D_m = 0 \\ D_s = 0 \end{gathered}
\end{array}
$$

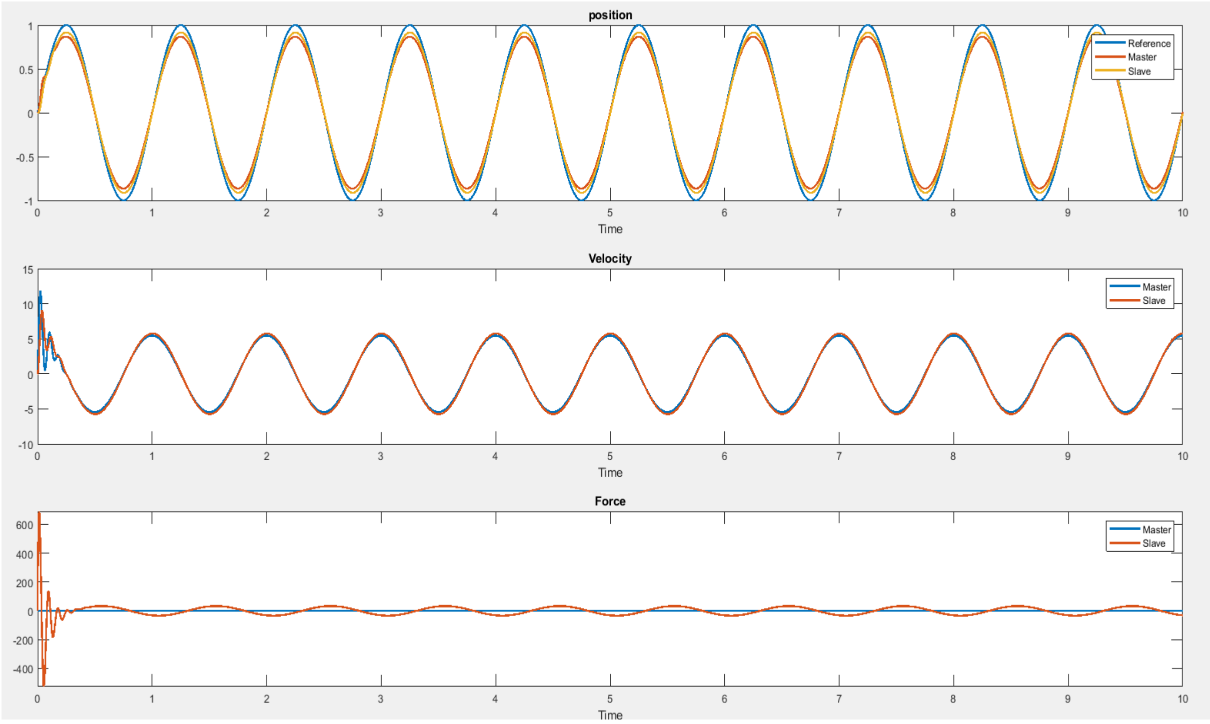

Figure 5: Four-channel bilateral teleoperation architecture results in free motion

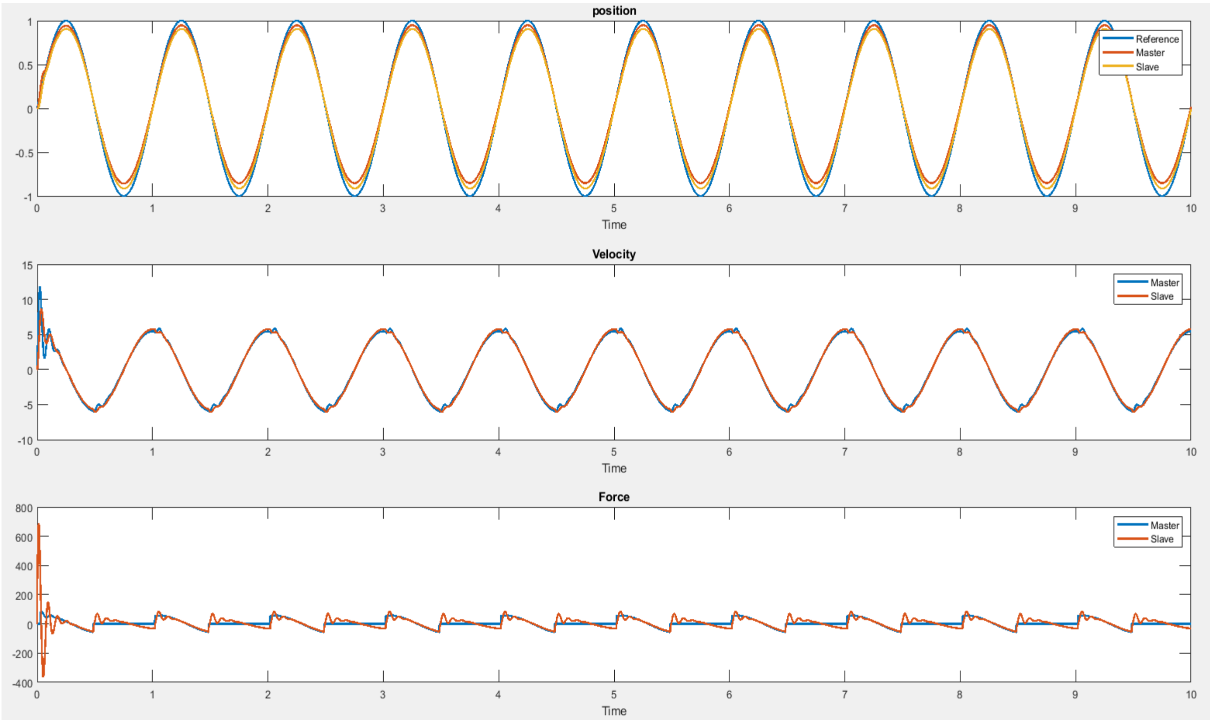

Figure 6: Four-channel bilateral teleoperation architecture results in contact

Moreover, we have to derive the expression for the hybrid matrix H which guarantee the transparency. The shape is the following one:

$$\begin{bmatrix} f_m \\ \dot{x}_m \end{bmatrix} = \begin{bmatrix} H_{11} H_{12} \\ H_{21} H_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_s \\ -f_s \end{bmatrix}$$

First of all we define the equivalence at sum blocks:

$$f_m(1 + C_{mf}) - C_4\dot{x}_s - C_2 f_s - \dot{x}_m C_m = \dot{x}_m Z_m \tag{1}$$

$$(f_m C_3 + \dot{x}_m C_1 - f_s(C_{sf} + 1) - \dot{x}_s C_s) = \dot{x}_s Z_s \tag{2}$$

1.

$$f_m = H_{11}\dot{x}_s \qquad f_s = 0$$

$$\downarrow$$

Keep $f_s = 0$ in Eq. 1

$$\downarrow$$

$$\dot{x}_m = \frac{f_m(1 + C_{mf}) - C_4\dot{x}_s}{Z_{cm}} \qquad where \qquad Z_{cs} = Z_s + C_s, \qquad Z_{cm} = Z_m + C_m$$

$$\downarrow$$

Put $\dot{x}_m$ in Eq. 2

$$\downarrow$$

$$(C_4C_1 + Z_{cm}Z_{cs})\dot{x}_s = f_m(C_1 + C_1C_{mf} + C_3Z_{cm})$$

$$\downarrow$$

$$H_{11} = \frac{C_4C_1 + Z_{cm}Z_{cs}}{C_1(1 + C_{mf}) + C_3Z_{cm}}$$

2.

$$f_m = H_{12}(-f_s) \qquad \dot{x}_s = 0$$

$$\downarrow$$

Keep $\dot{x}_s = 0$ in Eq. 1

$$\downarrow$$

$$\dot{x}_m = \frac{f_m(1 + C_{mf}) - C_2 f_s}{Z_{cm}} \qquad where \qquad Z_{cs} = Z_s + C_s, \qquad Z_{cm} = Z_m + C_m$$

$$\downarrow$$

Put $\dot{x}_m$ in Eq. 2

$$\downarrow$$

$$f_m = \frac{(1+C_{sf})Z_{cm} + C_1 C_2 f_s}{C_1(1+C_{mf}) + C_3 Z_{cm}}$$

$$\downarrow$$

$$H_{12} = \frac{-(1+C_{sf})Z_{cm} - C_1 C_2}{C_1(1+C_{mf}) + C_3 Z_{cm}}$$

3.

$$\dot{x}_m = H_{21}(\dot{x}_s) \qquad f_s = 0$$

$$\downarrow$$

Keep $f_s = 0$ in Eq. 1

$$\downarrow$$

$$f_m = \frac{\dot{x}_m Z_{cm} + C_4 \dot{x}_s}{(1+C_{mf})} \qquad where \qquad Z_{cs} = Z_s + C_s, \qquad Z_{cm} = Z_m + C_m$$

$$\downarrow$$

Put f_m in Eq. 2

$$\downarrow$$

$$\dot{x}_m(C_1 + C_1 C_{mf} + C_3 Z_{cm}) = \dot{x}_s(Z_{cs} + C_{mf} Z_{cs}) - C_3 C_4$$

$$\downarrow$$

$$H_{21} = \frac{Z_{cs}(1+C_{mf}) - C_3 C_4}{C_1(1+C_{mf}) + C_3 Z_{cm}}$$

4.

$$\dot{x}_m = H_{22}(-f_s) \qquad \dot{x}_s = 0$$

$$\downarrow$$

Keep $\dot{x}_s = 0$ in Eq. 1

$$\downarrow$$

$$f_m = \frac{\dot{x}_m Z_{cm} + C_2 f_s}{1+C_{mf}} \qquad where \qquad Z_{cs} = Z_s + C_s, \qquad Z_{cm} = Z_m + C_m$$

$$\downarrow$$

Put f_m in Eq. 2

$$\downarrow$$

$$\dot{x}_m(C_3 Z_{cm} + C_1(1+C_{mf})) = -C_2 f_s C_3 + (1+C_{mf})(1+C_{sf})$$

$$\downarrow$$

$$H_{22} = \frac{C_2 C_3 - (1+C_{mf})(1+C_{sf})}{C_3 Z_{cm} + C_1(1+C_{mf})}$$

3 Kalman filtering

In the third assignment we have to implement the Kalman filter/predictor and the steady-state Kalman filter/predictor to estimate the velocity and acceleration from noisy position measurements of a motor. Everything has been implemented in Matlab, here we have reported only the formulas, parameters and achieved results. Moreover, we have tried to estimate the velocity and acceleration using Euler Approximation with a low-pass filter.

For the **Kalman Filter** we have:

$$\begin{aligned}\hat{X}_{k+1|k+1} &= A\hat{X}_{k|k} + K_{k+1}(y_{k+1} - CA\hat{X}_{k|k}) \\ P_{k+1|k} &= AP_{k|k-1}A^T - AP_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1}A^T + Q\end{aligned}$$

where

$$K_{k+1} = P_{k+1}|KC^T(CP_{k+1}|KC^T + R)^{-1}$$

For the **Kalman Predictor** we have:

$$\begin{aligned}\hat{X}_{k+1|k} &= A\hat{X}_{k|k-1} + \bar{K}_{k+1}(y_k - CA\hat{X}_{k|k-1}) \\ P_{k+1|k} &= AP_{k|k-1}A^T - AP_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1}A^T + Q\end{aligned}$$

where

$$\bar{K}_k = AP_{k+1}|KC^T(CP_{k|}K_{-1}C^T + R)^{-1}$$

For the **Steady-State Kalman Filter** we have:

$$\begin{aligned}\hat{X}_{k+1|k+1} &= A\hat{X}_{k|k} + K_\infty(y_{k+1} - CA\hat{X}_{k|k}) \\ P &= APA^T - APC^T(CPC^T + R)^{-1}CPA^T + Q\end{aligned}$$

where

$$K_\infty = PC^T(CPC^T + R)^{-1}$$

For the **Steady-State Kalman Predictor** we have:

$$\begin{aligned}\hat{X}_{k+1|k} &= A\hat{X}_{k|k-1} + \bar{K}_\infty(y_k - CA\hat{X}_{k|k-1}) \\ P &= APA^T - APC^T(CPC^T + R)^{-1}CPA^T + Q\end{aligned}$$

where

$$\bar{K}_\infty = APC^T(CPC^T + R)^{-1}$$

The parameters are the following one:

$$A = \begin{bmatrix} 1 & T_s & \frac{Ts^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \qquad P_0 = \begin{bmatrix} 1\exp^{-4} & 0 & 0 \\ 0 & 1\exp^{-4} & 0 \\ 0 & 0 & 1\exp^{-4} \end{bmatrix} \qquad \begin{aligned} &q = 100000 \\ &Q = q\begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix}\begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix}^T \\ &R = 1 \end{aligned} \qquad \begin{aligned} T_s &= 0.001 \\ F_c &= 5 \end{aligned}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad x_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

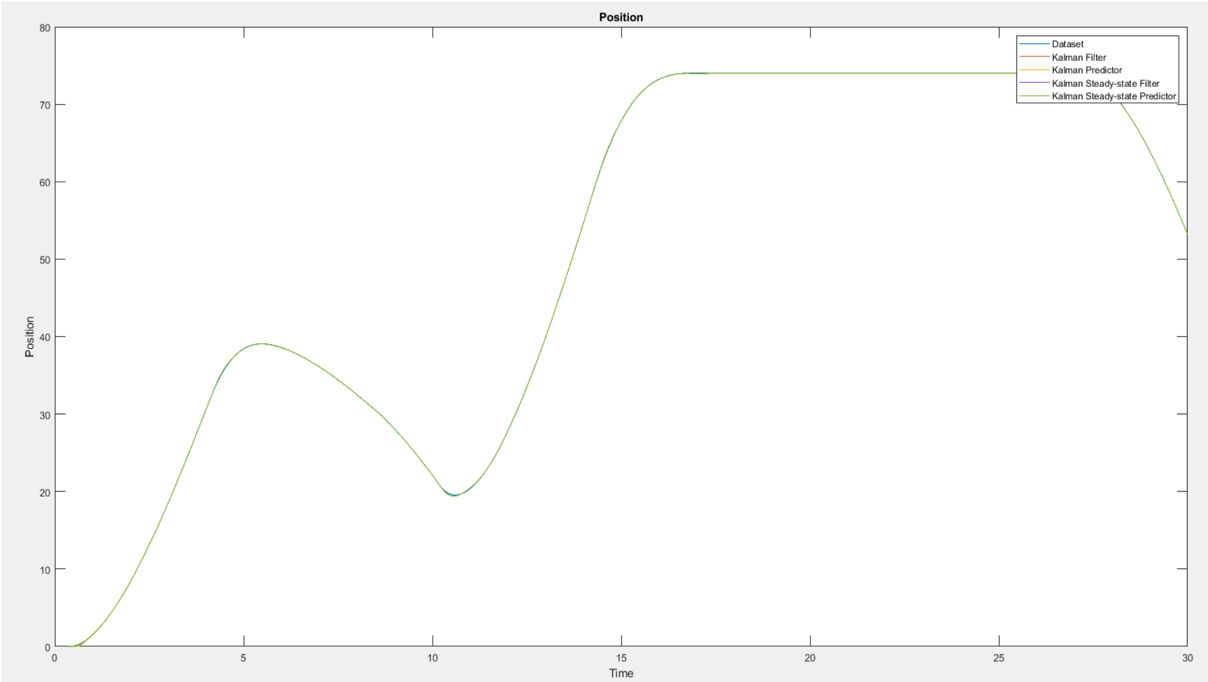

Figure 7: Kalman filter/predictor and the steady-state Kalman filter/predictor for position

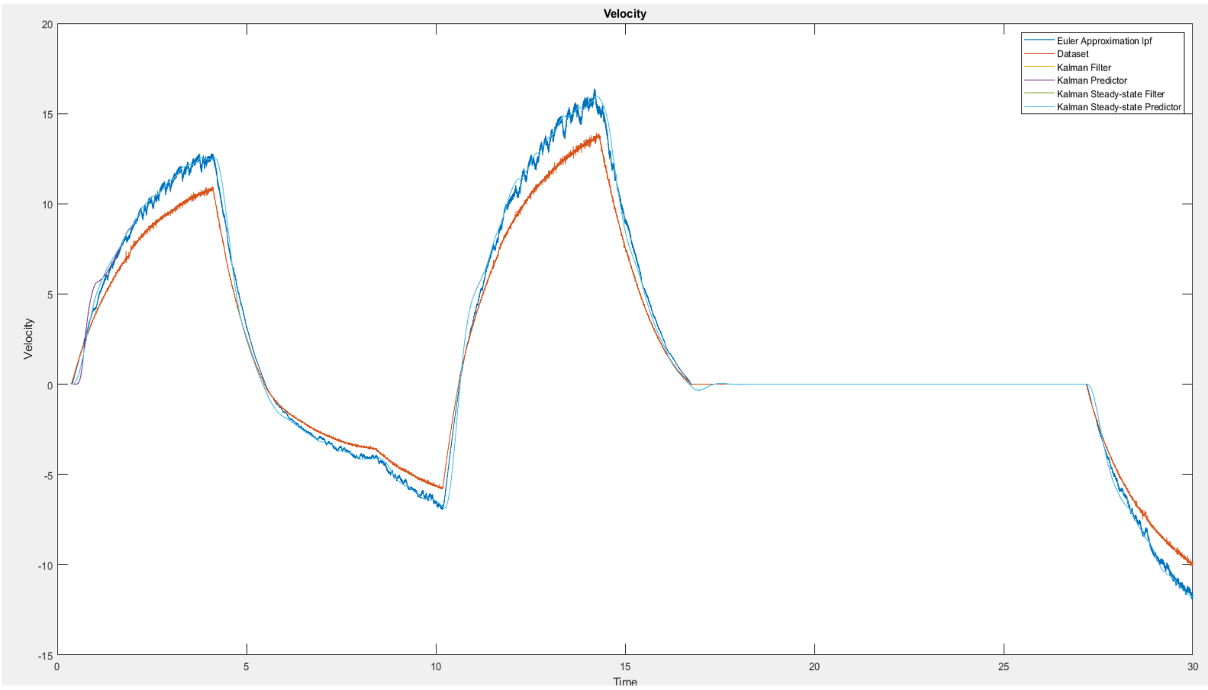

Figure 8: Kalman filter/predictor and the steady-state Kalman filter/predictor for velocity

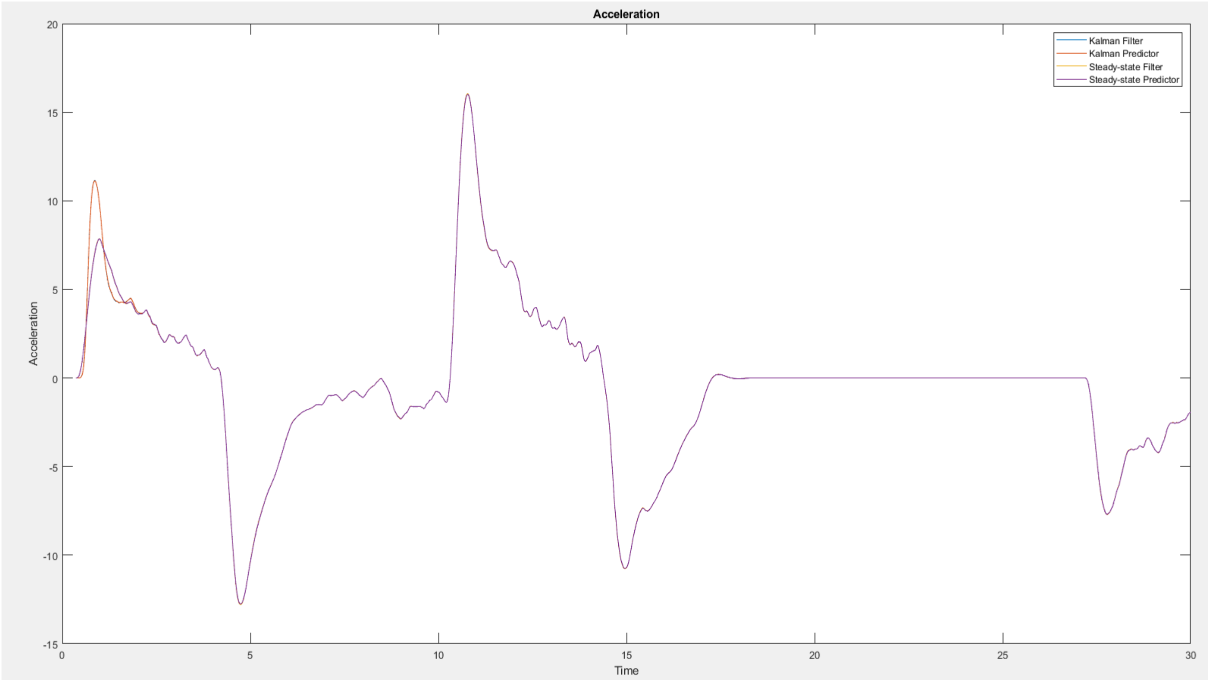

Figure 9: Kalman filter/predictor and the steady-state Kalman filter/predictor for acceleration

4 Kalman Smoother

In the fourth assignment we have to implement the Kalman smoother and estimate the velocity and acceleration from noisy position measurements of a motor. Everything has been implemented in Matlab, here we have reported only the formulas, parameters and achieved results. Moreover, we have made a comparison with the others methods shown in previous section.It is based on two steps:

STEP 1: **Forward step**:

$$
\begin{aligned}
\hat{X}^f_{k+1|k+1} &= A\hat{X}^f_{k|k} + K_{k+1}(y_{k+1} - CA\hat{X}^f_{k|k}) \\
\hat{X}^f_{0|0} &= \bar{x}_0 \\
P^f_{k|k} &= P^f_{k+1|k} - P^f_{k+1|k}C^T(CP^f_{k+1|k}C^T + R)^{-1}CP^f_{k+1|k} \\
P^f_{k+1|k} &= AP^f_{k|k-1}A^T - AP^f_{k|k-1}C^T(CP^f_{k|k-1}C^T + R)^{-1}CP^f_{k|k-1}A^T + Q \\
P^f_{0|0} &= P_0
\end{aligned}
$$

STEP 2: **Backward step**:

$$
\begin{aligned}
\hat{X}^s_{k|k} &= A\hat{X}^f_{k|k} + \bar{K}_k(\hat{X}^s_{k+1|N} - \hat{X}^f_{k+1|k}) \\
\hat{X}^s_{N|N} &= \hat{X}^f_{N|N} \\
\bar{K}_k &= P^f_{k|k}A^T(P^f_{k+1|k})^{-1} \\
P_{N|N} &= P^f_{k|k} + \bar{K}_k(P_{k+1|N} - P^f_{k+1|k})
\end{aligned}
$$

The parameters are the following one:

$$
A = \begin{bmatrix} 1 & T_s & \frac{Ts^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \qquad P_0 = \begin{bmatrix} 1\exp^{-4} & 0 & 0 \\ 0 & 1\exp^{-4} & 0 \\ 0 & 0 & 1\exp^{-4} \end{bmatrix} \qquad q = 100000 \qquad Q = q \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix} \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix}^T \qquad T_s = 0.001 \qquad F_c = 5
$$

$$
C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad x_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \qquad R = 1
$$

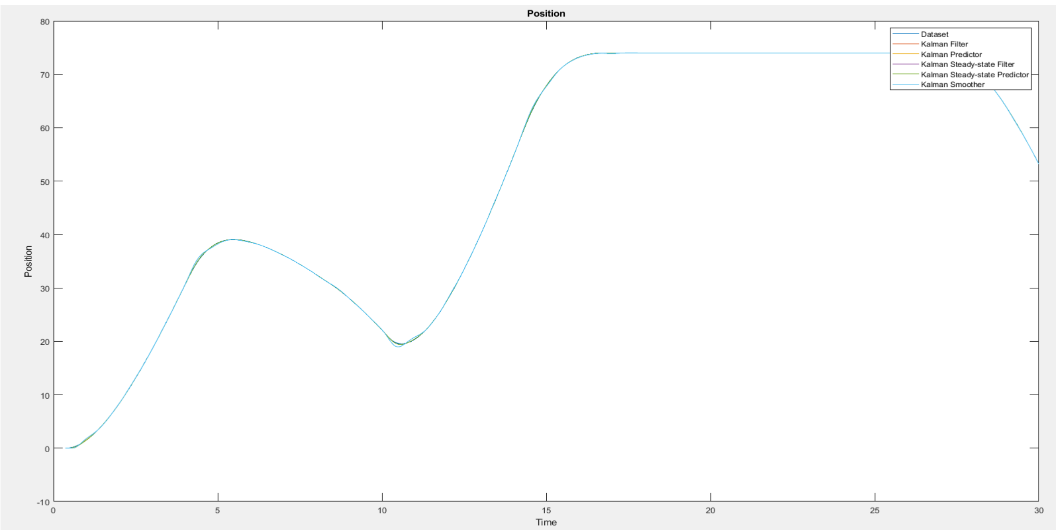

Figure 10: Kalman Smoother for position

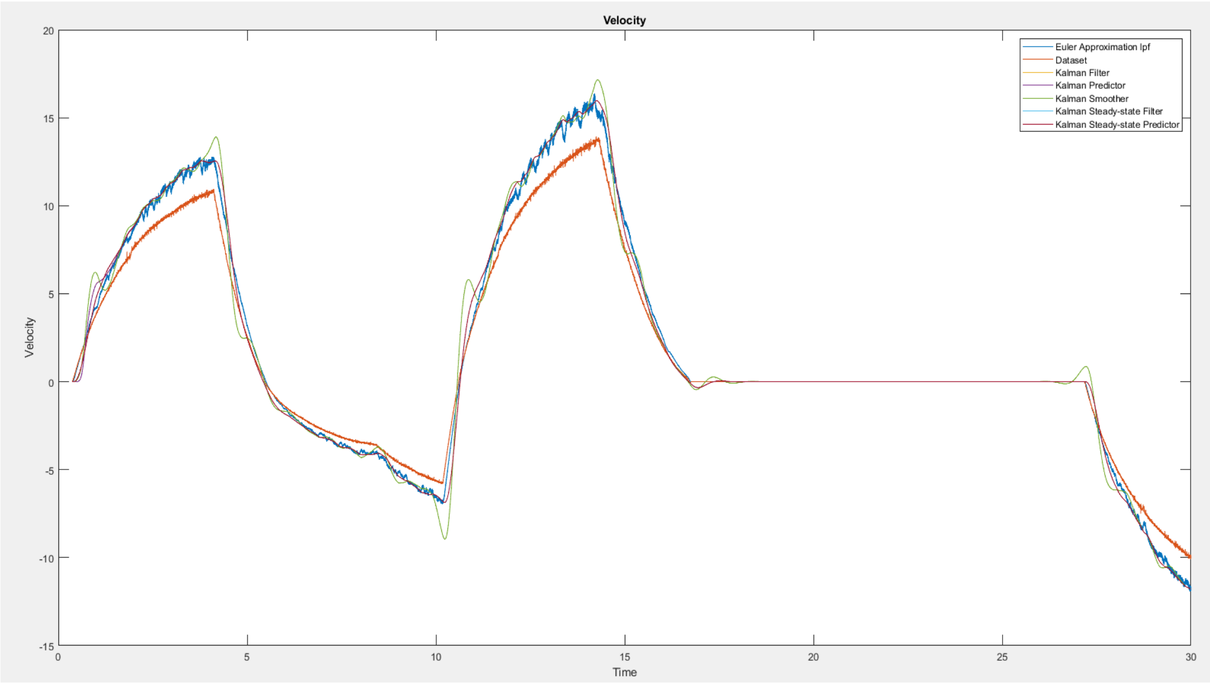

Figure 11: Kalman Smoother for velocity

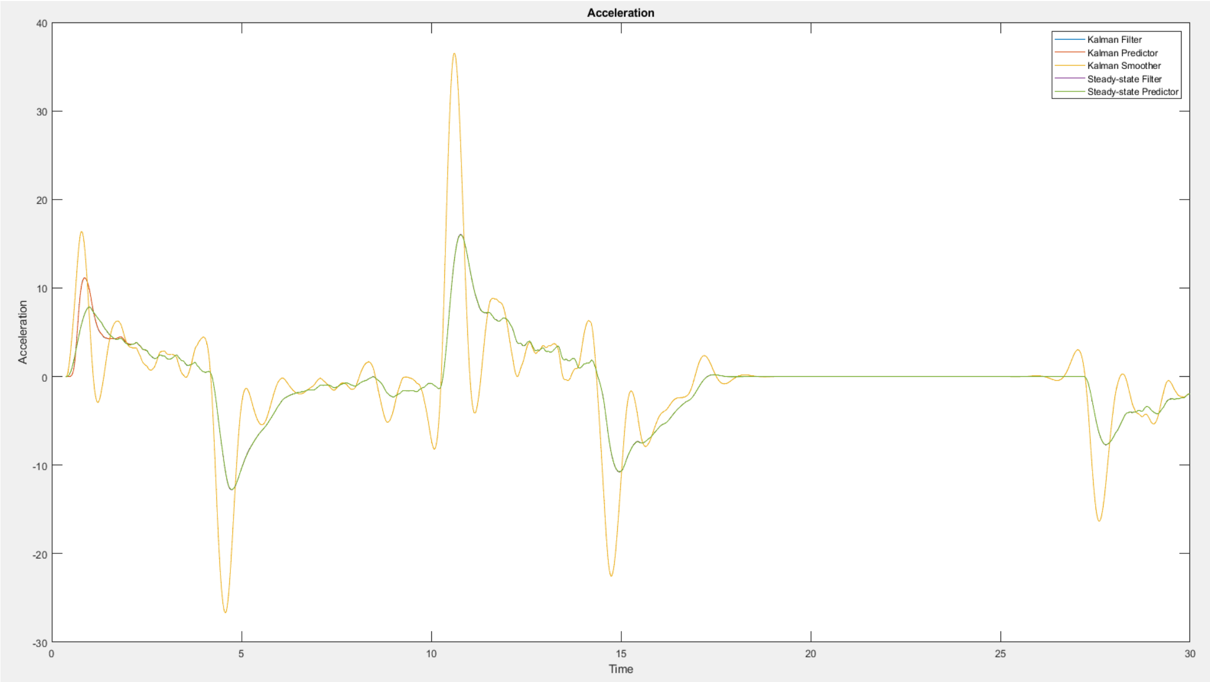

Figure 12: Kalman Smoother for acceleration

5 Linear Methods for Regression

For the fifth assignment we have to identify the parameters k and τ using the Least Square, the Recursive Least Square, and the Adaptive Algorithm on the DC motors data. Everything has been implemented in Matlab, here we report only the results. The velocity and the acceleration has been estimated starting from position measurements, using the kalman filter shown in previous section.

Our **model** is the following one:

$$Y = \begin{bmatrix} V(t_1) \\ V(t_2) \\ \dots \\ V(t_N) \end{bmatrix} \qquad X = \begin{bmatrix} \dot{\omega}(t_1) & \omega(t_1) \\ \dot{\omega}(t_2) & \omega(t_2) \\ \dots \\ \dot{\omega}(t_N) & \omega(t_N) \end{bmatrix} \quad \rightarrow \quad Y = X\beta$$

For the **Least Square** we have:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

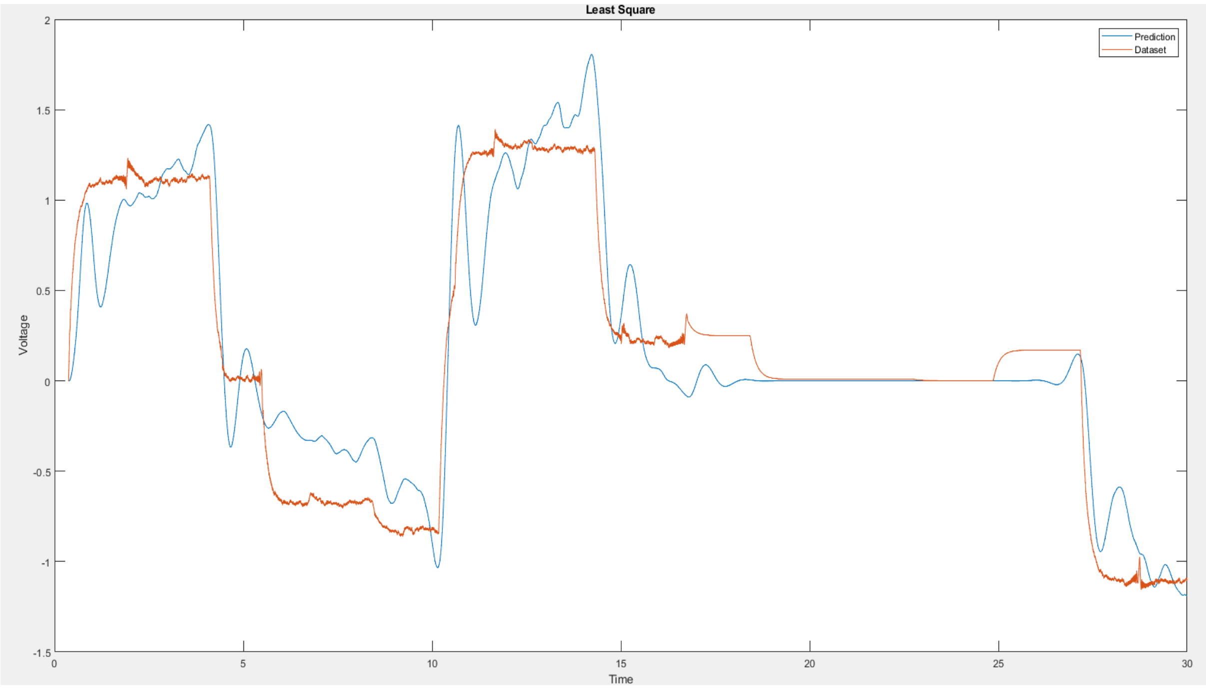

Figure 13: Least square

For the **Recursive Least Square** we have:

$$\begin{aligned} \hat{\beta}(k) &= \hat{\beta}(k-1) + K(k)e(k) \\ K(k) &= P(k)x_k^T \\ e(k) &= y_k - x_k\hat{\beta}(k-1) \\ P(k) &= P(k-1) - \frac{P(k-1)x_k^T x_k P(k-1)}{1 + x_k P(k-1)x_k^T} \end{aligned}$$

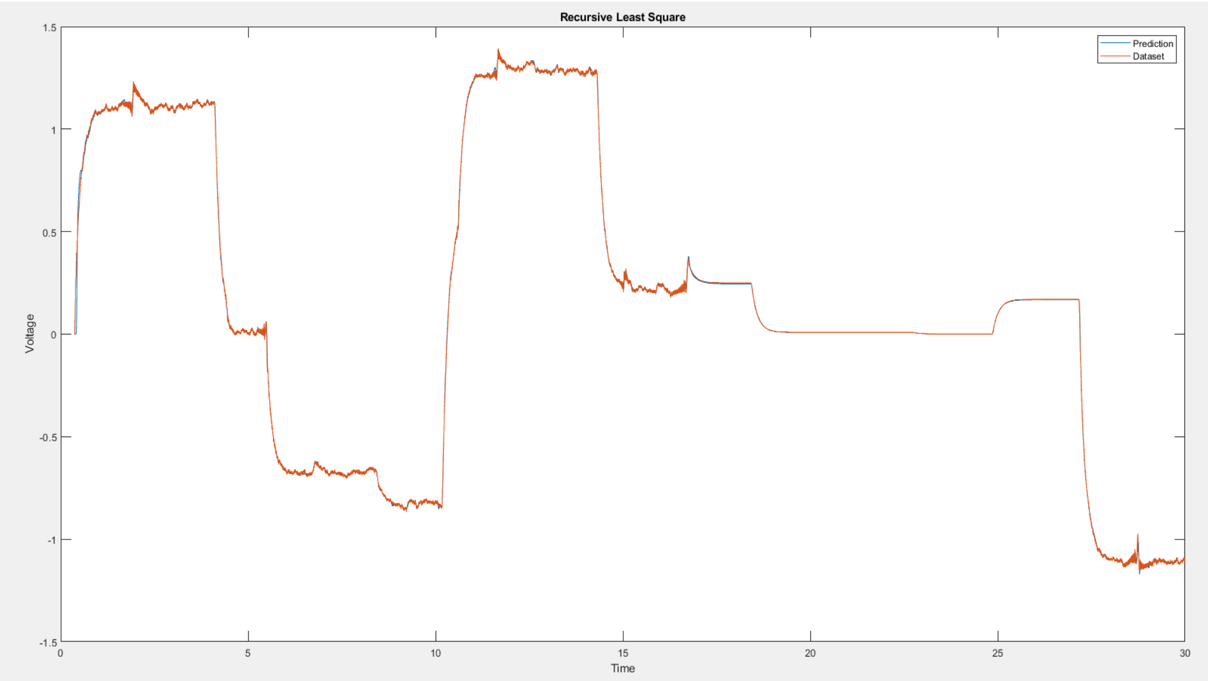

Figure 14: Recursive Least Square

For the **Adaptive Algorithm** we have:

$$\hat{\beta}(k) = \hat{\beta}(k-1) + gT_s x^T(k)e(k) \qquad where \qquad g > 0$$

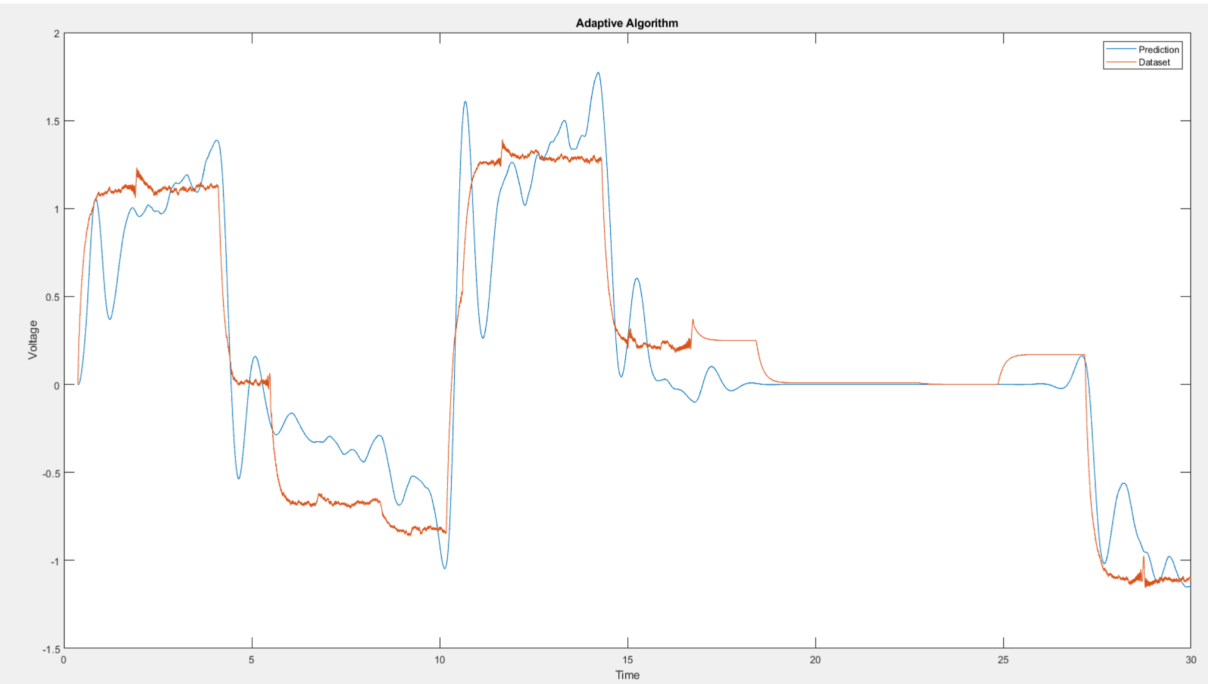

Figure 15: Adaptive ALgorithm

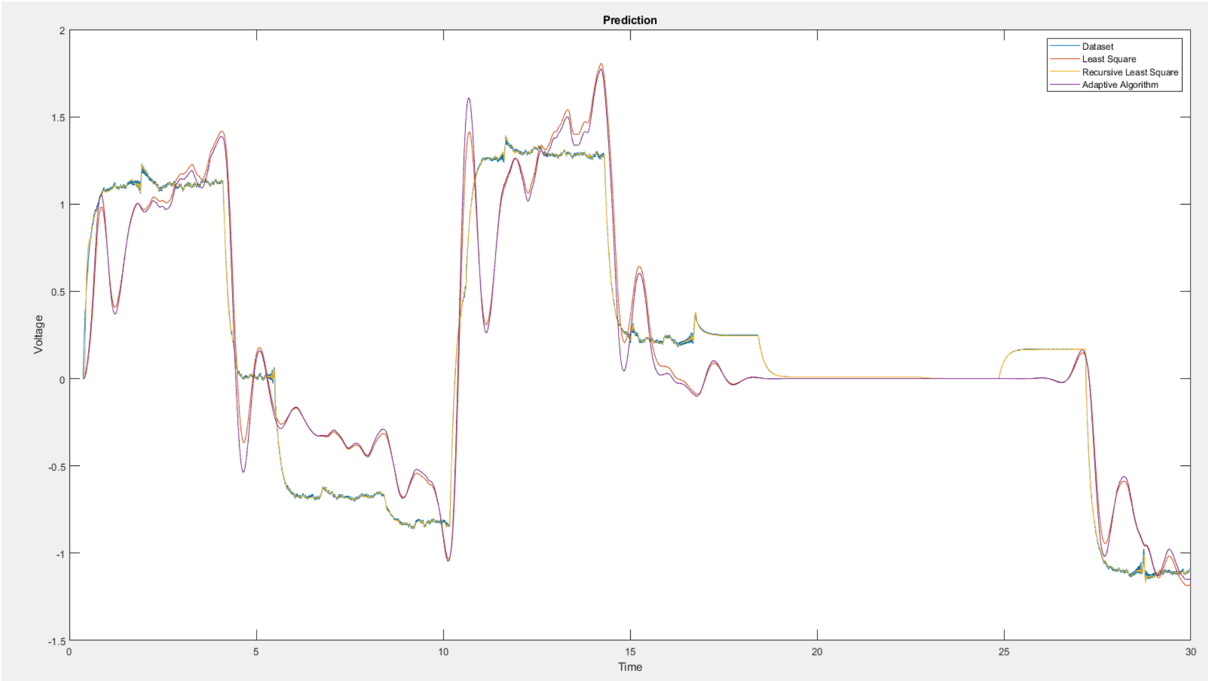

Figure 16: comparison between LS, RLS, AA

6 Scattering-based bilateral teleoperation architecture

In the sixth assignment we have to implement the Scattering-based bilateral teleoperation architecture, designed by G. Niemeyer and J.J. Slotine. We have to consider the Force-Position case and the Position-Position case. Everything has been implemented in Matlab/Simulink, here we have reported only the schema, parameters and achieved results.

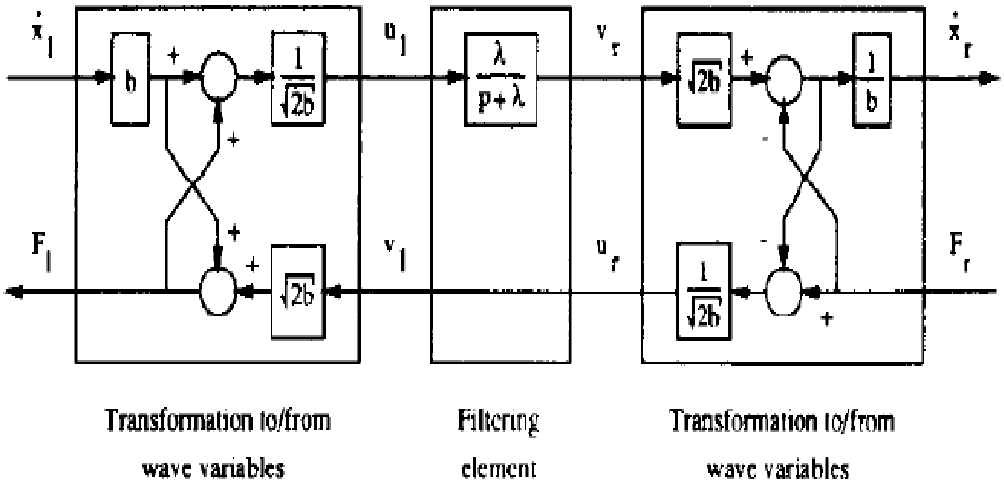

Figure 17: Scattering-based architecture scheme with low-pass filter

Let's start analyzing the Force-Position results.

$F_c = 0.5$	$P_h = 5$	$B_m = 0$	$B_s = 100$	$J_h = 0$	$J_e = 0$	$b = 1$
$A = 1$	$I_h = 2$	$k_m = 0$	$k_s = 80$	$B_h = 1.5$	$B_e = 10$	$delay = 1$
$F_{low} = 100$	$env = 0.1$	$M_m = 0.5$	$M_s = 2$	$K_h = 1$	$K_e = 200$	

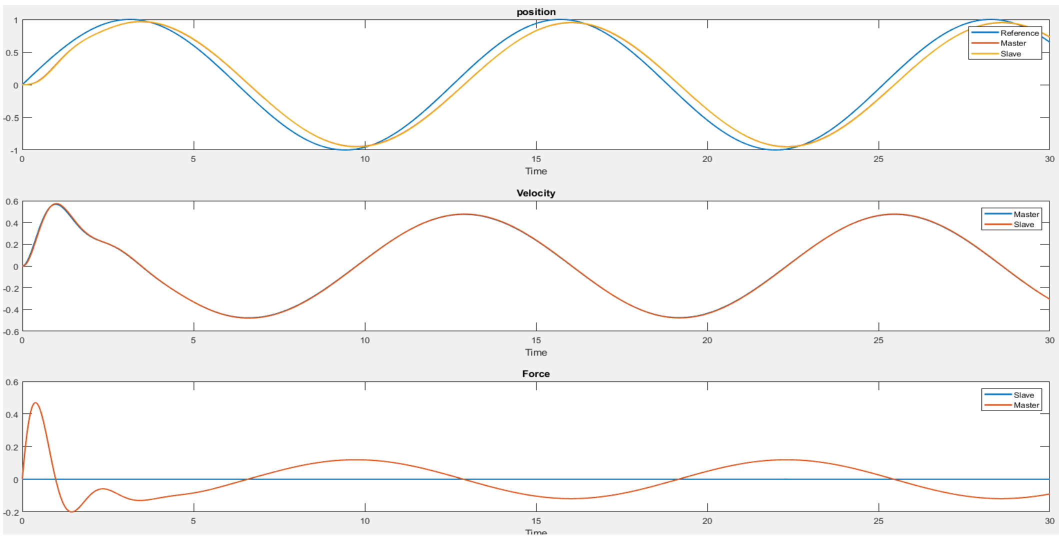

Figure 18: Scattering-based architecture FP in free motion

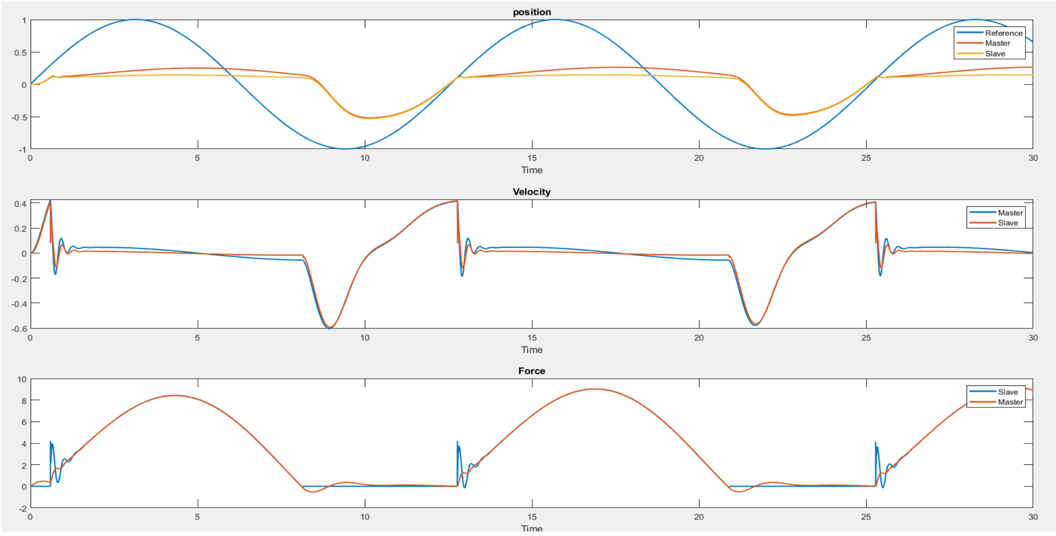

Figure 19: Scattering-based architecture FP in contact

Furthermore, we have evaluated how much it is robust to the delay. So, same parameters but delay = 10.

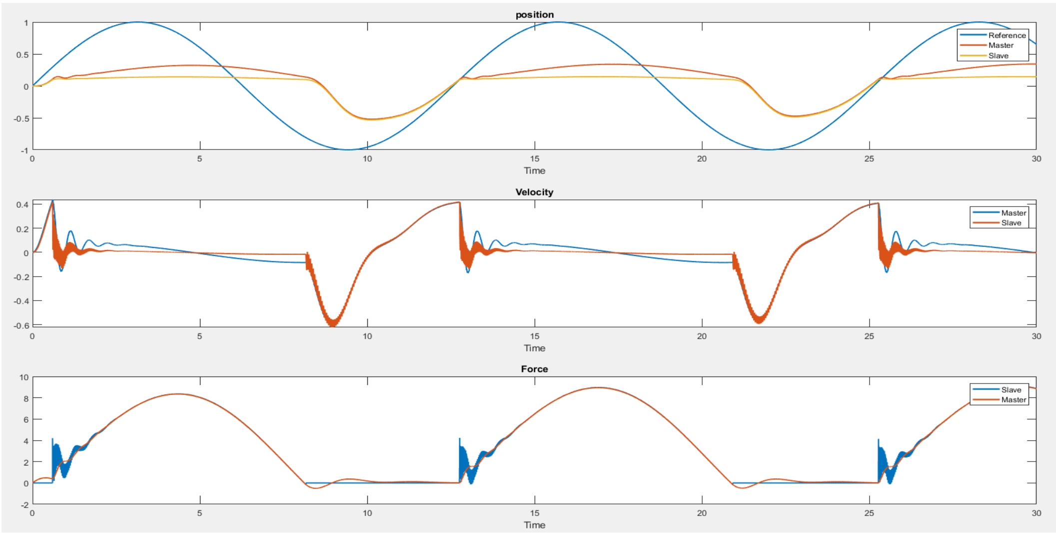

Figure 20: Scattering-based architecture FP in contact with more delay

Moreover, we created another model adding the measurement noise to the position/force signals, and estimate velocities from positions (using the previous methods).

$$\begin{array}{cccccc} F_c = 0.5 & & & & & \\ A = 1 & P_h = 5 & B_m = 0 & B_s = 100 & J_h = 0 & J_e = 0 \\ F_{low} = 100 & I_h = 2 & k_m = 0 & k_s = 80 & B_h = 1.5 & B_e = 10 \\ b = 1 & env = 0.1 & M_m = 0.5 & M_s = 2 & K_h = 1 & K_e = 200 \\ delay = 1 & & & & & \end{array}$$

$$q = 10000000$$

$$Q = q \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix} \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix}^T$$

$$R = 1$$

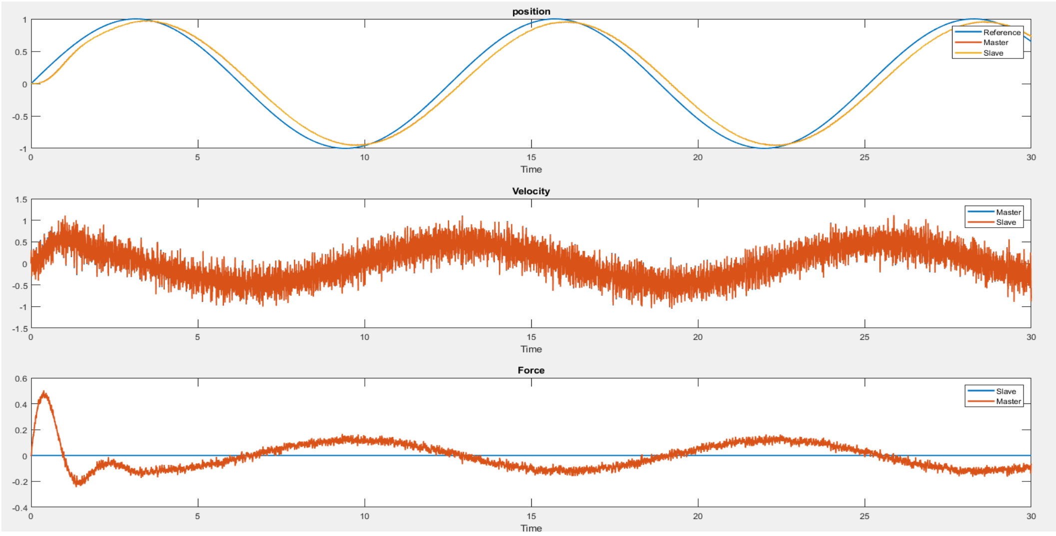

Figure 21: Scattering-based architecture FP in free motion with noise

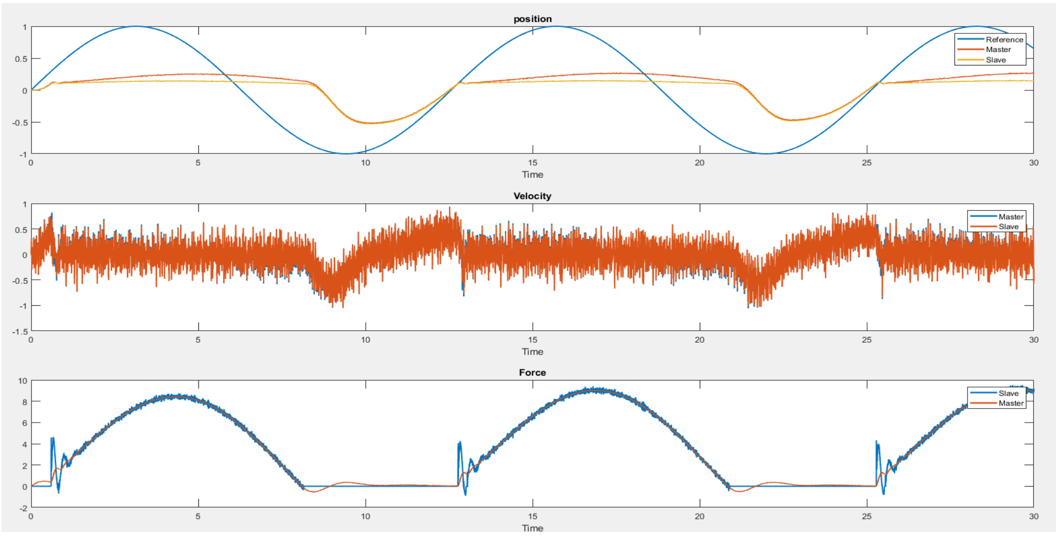

Figure 22: Scattering-based architecture FP in contact with noise

Let's analyze the Position-Position results.

$F_c = 0.5$	$P_h = 5$	$B_m = 100$	$B_s = 100$	$J_h = 0$	$J_e = 0$	$b = 1$
$A = 1$	$I_h = 5$	$k_m = 50$	$k_s = 80$	$B_h = 1.5$	$B_e = 10$	$delay = 1$
$F_{low} = 100$	$env = 0.1$	$M_m = 0.5$	$M_s = 2$	$K_h = 1$	$K_e = 10$	

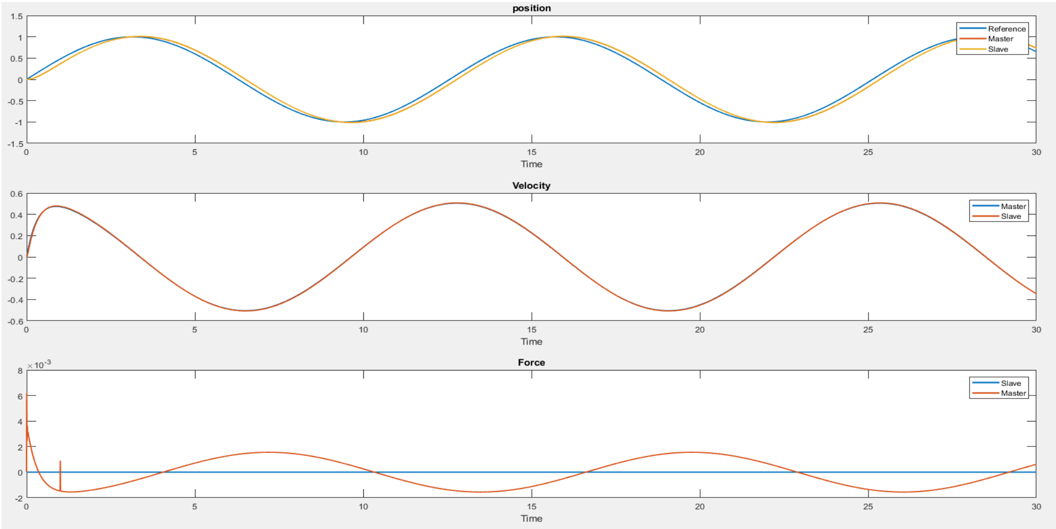

Figure 23: Scattering-based architecture PP in free motion

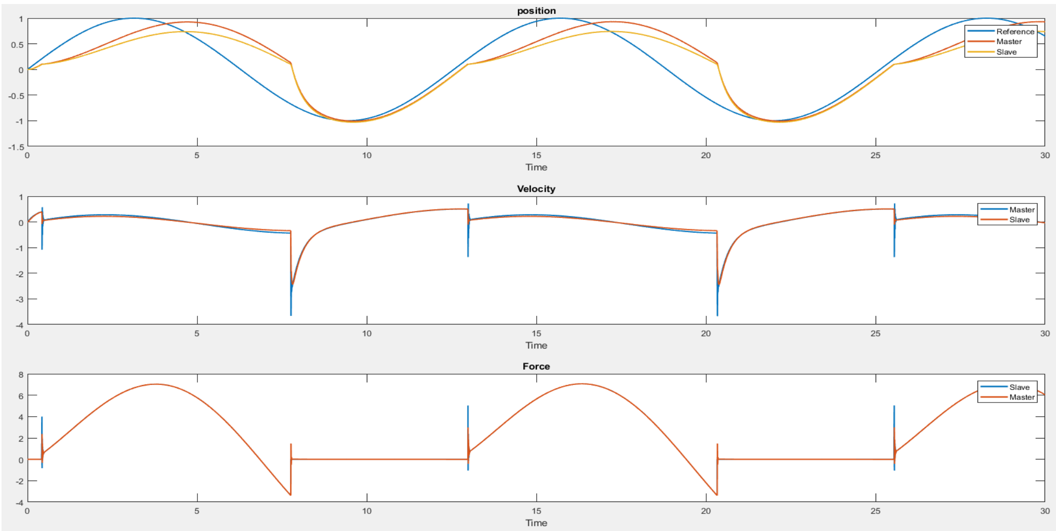

Figure 24: Scattering-based architecture PP in contact

Furthermore, we have evaluated how much it is robust to the delay. So, same parameters but delay = 10.

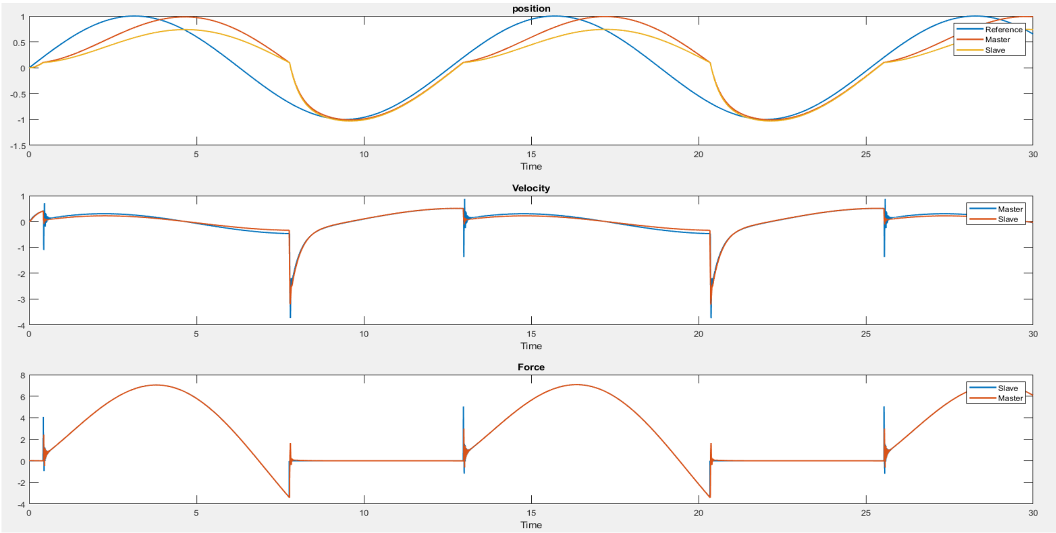

Figure 25: Scattering-based architecture PP in contact with more delay

Moreover, we created another model adding the measurement noise to the position/force signals, and estimate velocities from positions (using the previous methods).

$$
\begin{array}{rrrrrr}
F_c = 0.5 & & & & & \\
A = 1 & P_h = 5 & B_m = 100 & B_s = 100 & J_h = 0 & J_e = 0 \\
F_{low} = 100 & I_h = 5 & k_m = 50 & k_s = 80 & B_h = 1.5 & B_e = 10 \\
b = 1 & env = 0.1 & M_m = 0.5 & M_s = 2 & K_h = 1 & K_e = 10 \\
delay = 1 & & & & &
\end{array}
\qquad
\begin{array}{l}
q = 100000000 \\
Q = q \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix} \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix}^T \\
R = 1
\end{array}
$$

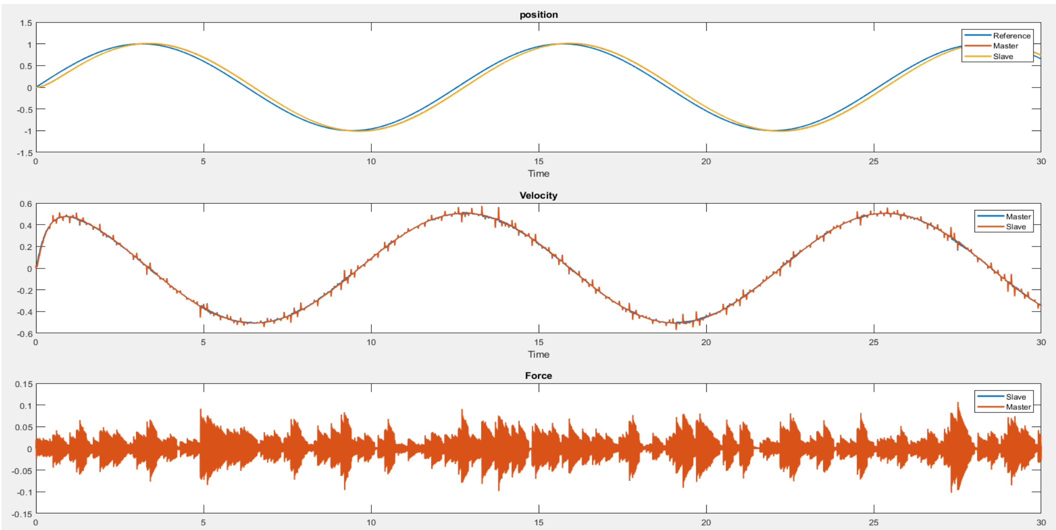

Figure 26: Scattering-based architecture PP in free motion with noise

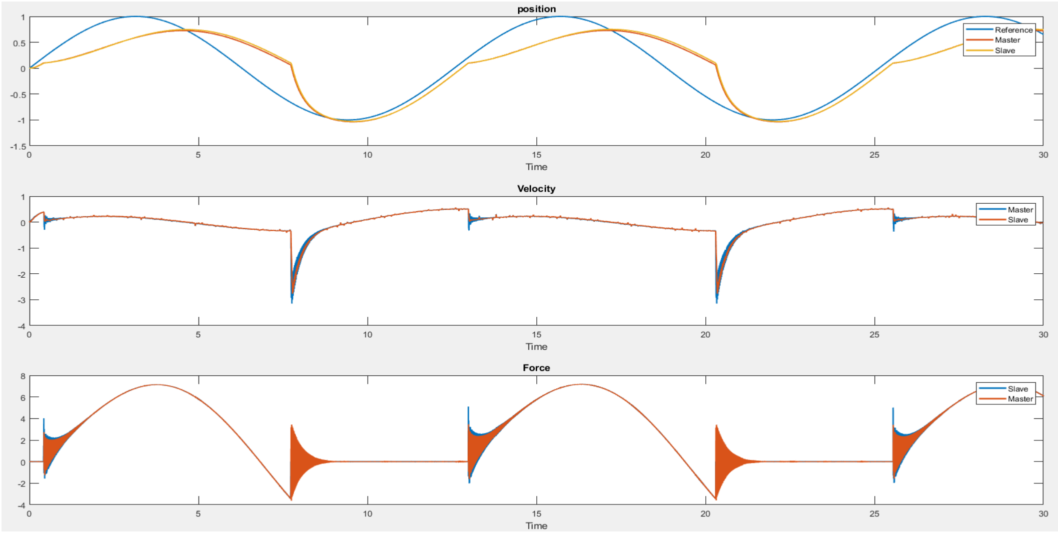

Figure 27: Scattering-based architecture PP in contact with noise

7 Tank-based bilateral teleoperation architecture

In the sixth assignment we have to implement the Tank-based bilateral teleoperation architecture, designed by M. Franken, S. Stramigioli, S. Misra, C. Secchi and A. Machelli, shown in figure 28. We have to consider the Force-Position case and the Position-Position case. Everything has been implemented in Matlab/Simulink, here we have reported only the schema, parameters and achieved results.

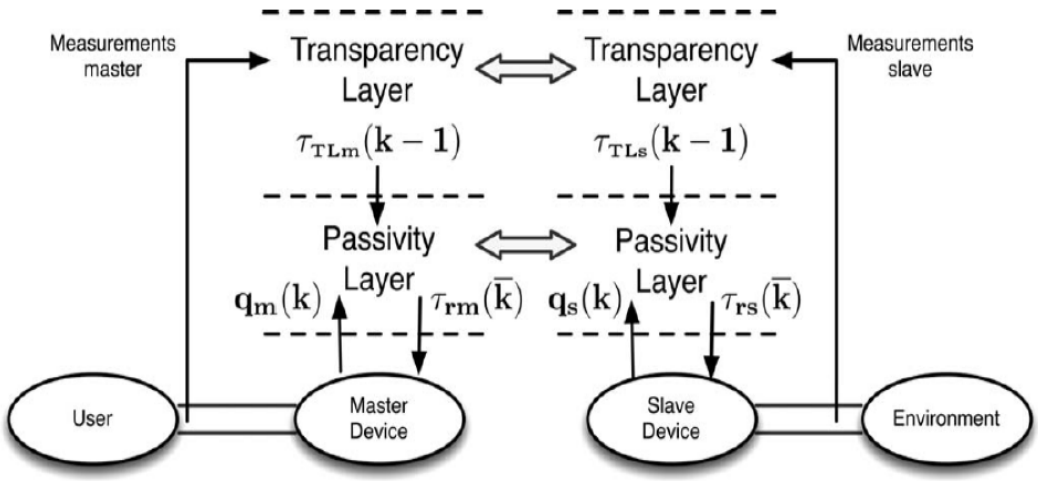

Figure 28: Tank-based architecture scheme

Let's start analyzing the Force-Position results.

$F_c = 0.5$	$P_h = 5$	$B_m = 0$	$B_s = 100$	$J_h = 0$	$J_e = 0$	$beta = 0.1$
$A = 1$	$I_h = 2$	$k_m = 0$	$k_s = 80$	$B_h = 1.5$	$B_e = 10$	$alpha = 0.1$
$H_{initm} = 10$	$env = 0.5$	$M_m = 0.5$	$M_s = 2$	$K_h = 1$	$K_e = 200$	$H_D = 1$
$H_{inits} = 0$						$delay = 1$

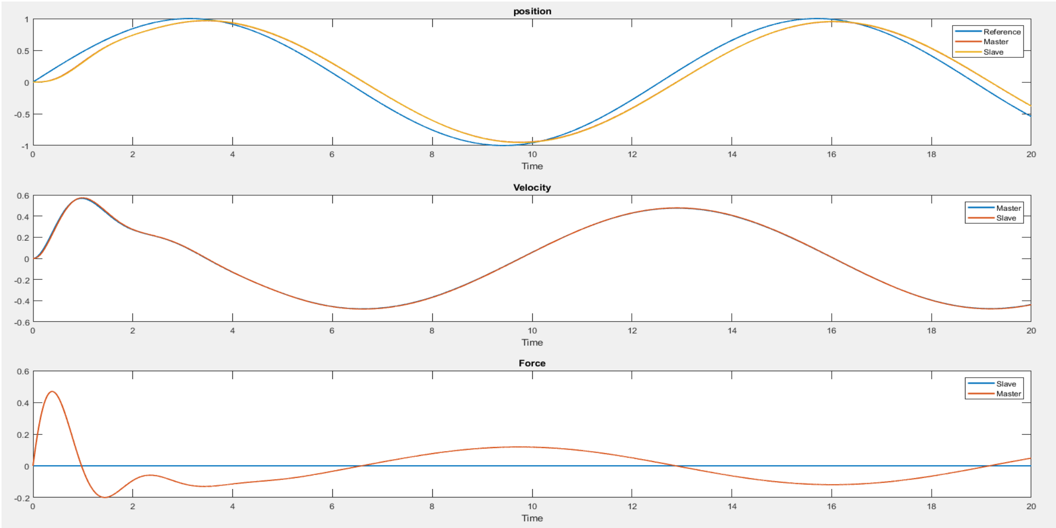

Figure 29: Tank-based architecture FP in free motion

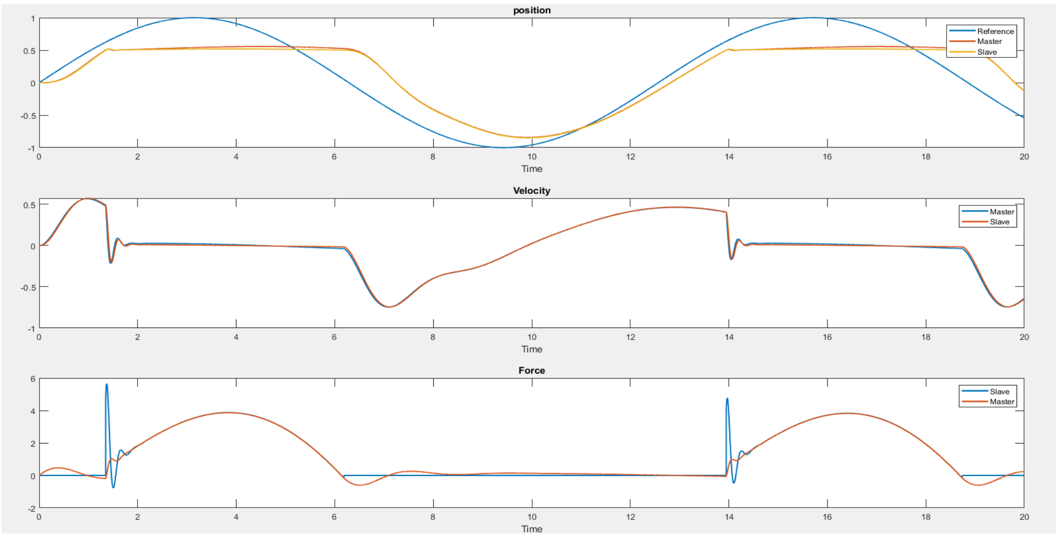

Figure 30: Tank-based architecture FP in contact

$$F_c = 0.5 \quad A = 1 \quad H_{initm} = 0 \quad H_{inits} = 0$$

$$P_h = 5 \quad I_h = 2 \quad env = 0.5$$

$$B_m = 0 \quad k_m = 0 \quad M_m = 0.5$$

$$B_s = 100 \quad k_s = 80 \quad M_s = 2$$

$$J_h = 0 \quad B_h = 1.5 \quad K_h = 1$$

$$J_e = 0 \quad B_e = 10 \quad K_e = 200$$

$$beta = 0.1 \quad alpha = 0.1 \quad H_D = 1 \quad delay = 1$$

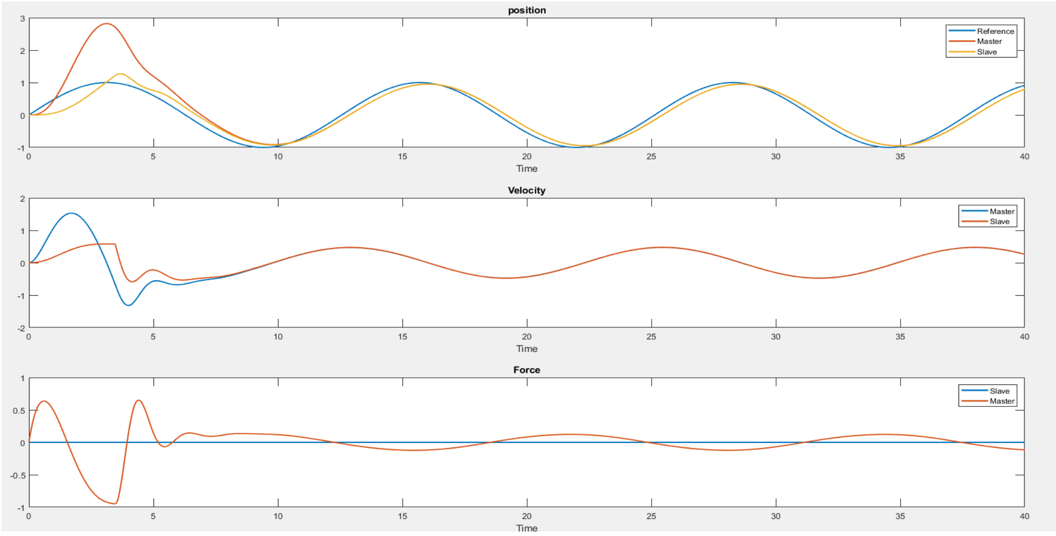

Figure 31: Tank-based architecture FP in free motion with zero initial energy

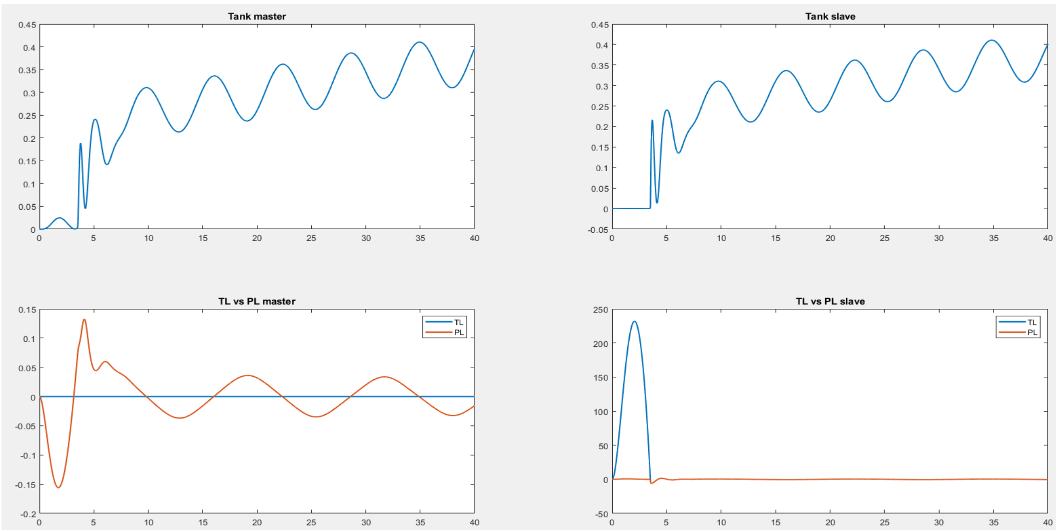

Figure 32: Tank level and applied torque of Tank-based architecture FP in free motion with zero initial energy

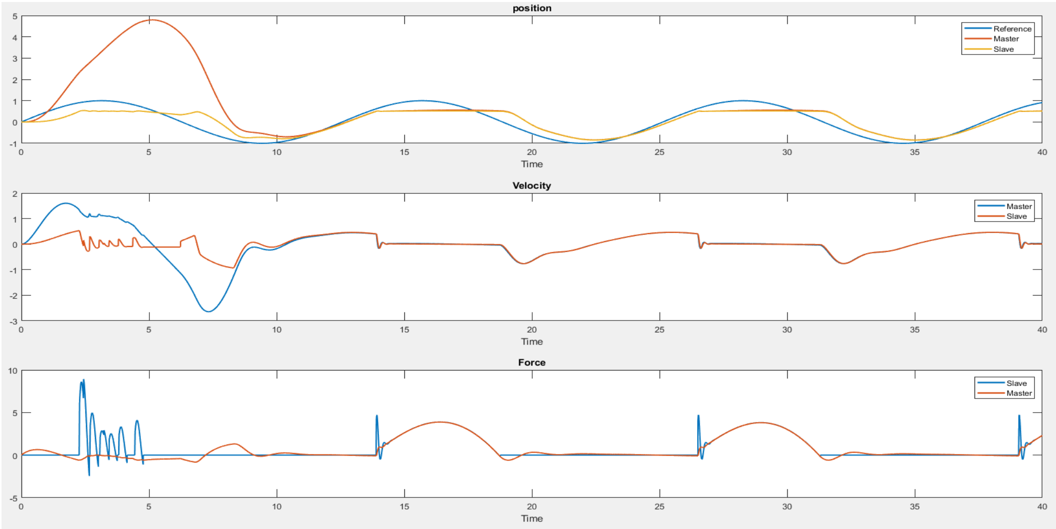

Figure 33: Tank-based architecture FP in contact with zero initial energy

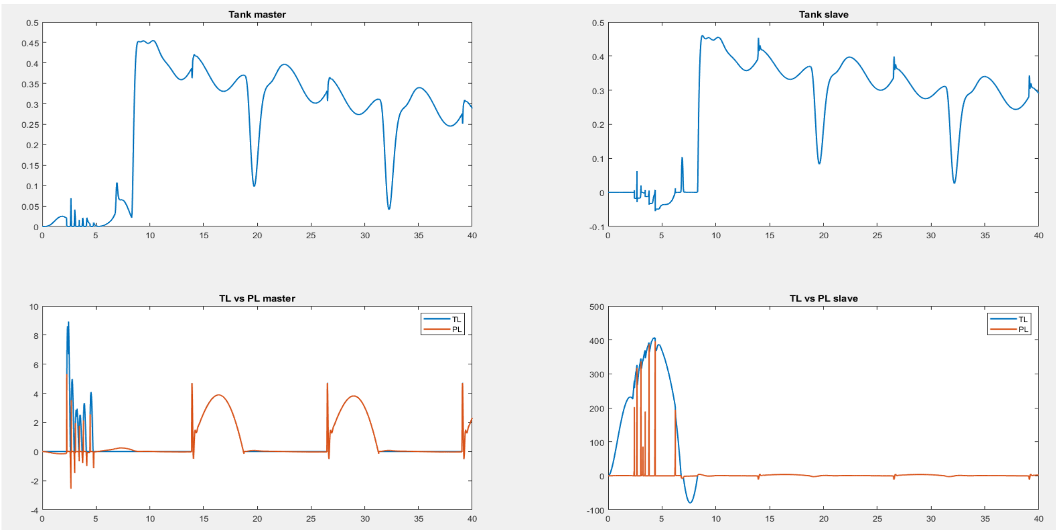

Figure 34: Tank level and applied torque of Tank-based architecture FP in contact with zero initial energy

Furthermore, we have evaluated how much it is robust to the delay. So, same parameters but delay = 10.

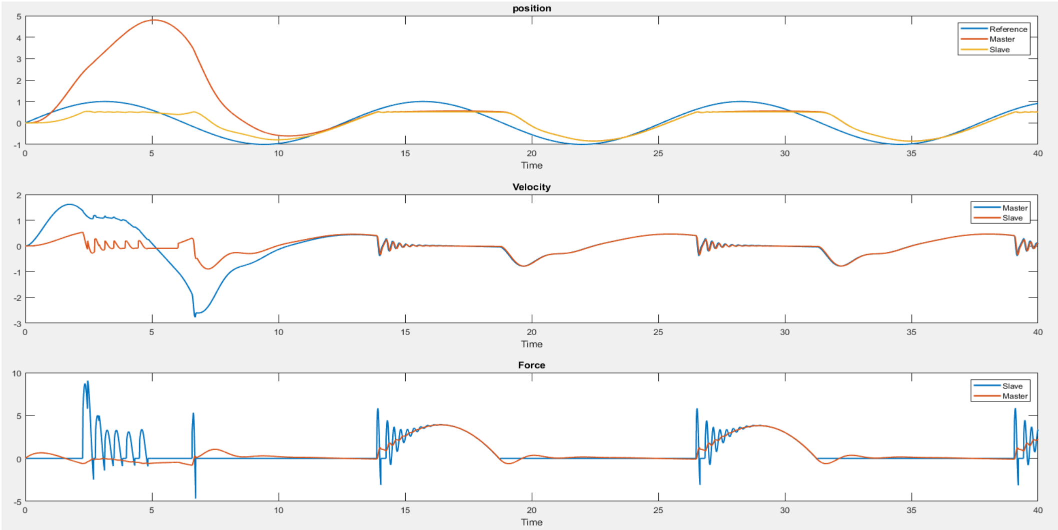

Figure 35: Tank-based architecture FP in contact with zero initial energy and more delay

Moreover, we created another model adding the measurement noise to the position/force signals, and estimate velocities from positions (using the previous methods).

$$
\begin{array}{ccccccc}
F_c = 0.5 & & & & & & q = 100000000 \\
A = 1 & P_h = 5 & B_m = 0 & B_s = 100 & J_h = 0 & J_e = 0 & \\
H_{initm} = 10 & I_h = 2 & k_m = 0 & k_s = 80 & B_h = 1.5 & B_e = 10 & Q = q \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix} \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix}^T \\
H_{inits} = 10 & env = 0.5 & M_m = 0.5 & M_s = 2 & K_h = 1 & K_e = 200 & \\
& & & & & & R = 1
\end{array}
$$

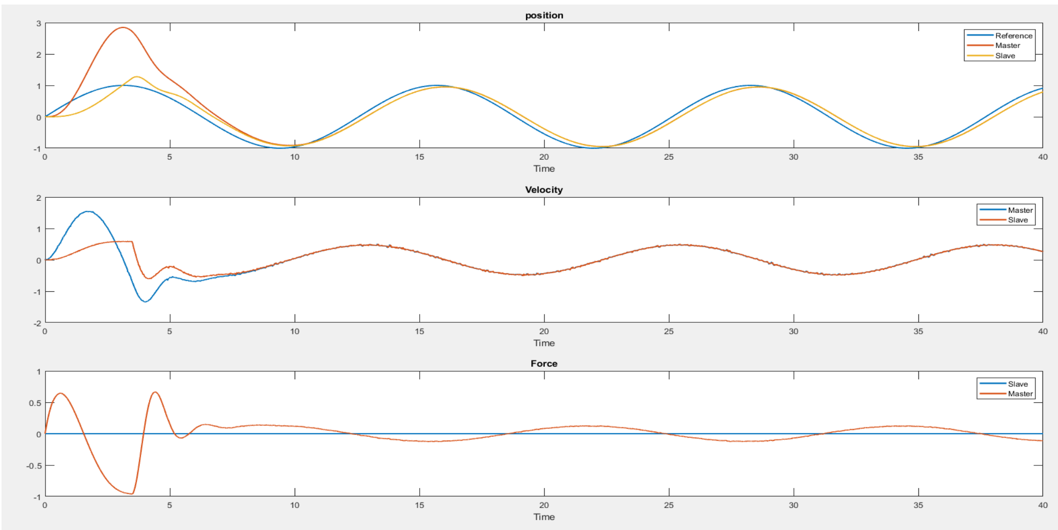

Figure 36: Tank-based architecture FP in free motion with noise

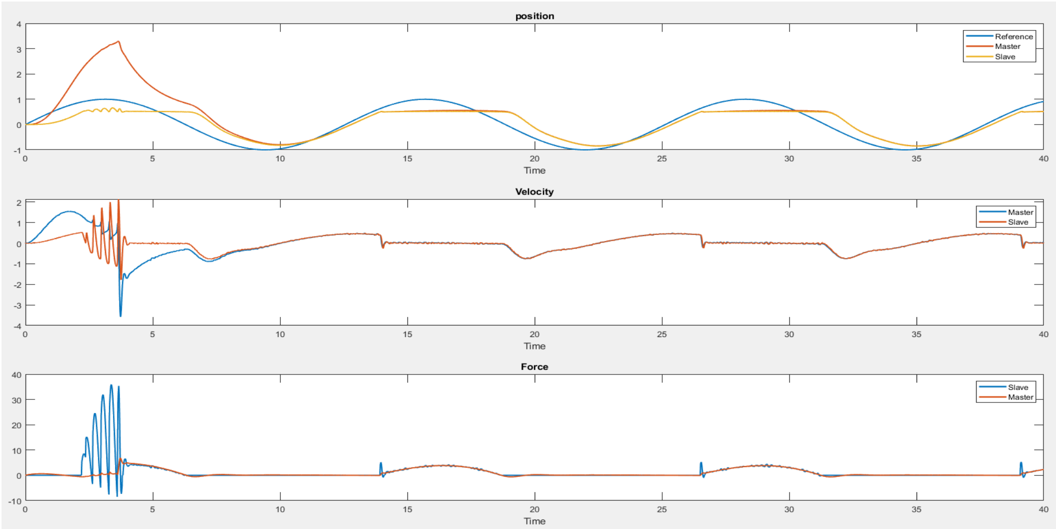

Figure 37: Tank-based architecture FP in contact with noise

Let's analyze the Position-Position results.

$$F_c = 0.5 \quad A = 1 \quad H_{initm} = 10 \quad H_{inits} = 5$$

$$P_h = 5 \quad I_h = 2 \quad env = 0.5$$

$$B_m = 250 \quad k_m = 80 \quad M_m = 0.5$$

$$B_s = 200 \quad k_s = 80 \quad M_s = 2$$

$$J_h = 0 \quad B_h = 1.5 \quad K_h = 1$$

$$J_e = 0 \quad B_e = 10 \quad K_e = 200$$

$$beta = 0.1 \quad alpha = 0.1 \quad H_D = 1 \quad delay = 1$$

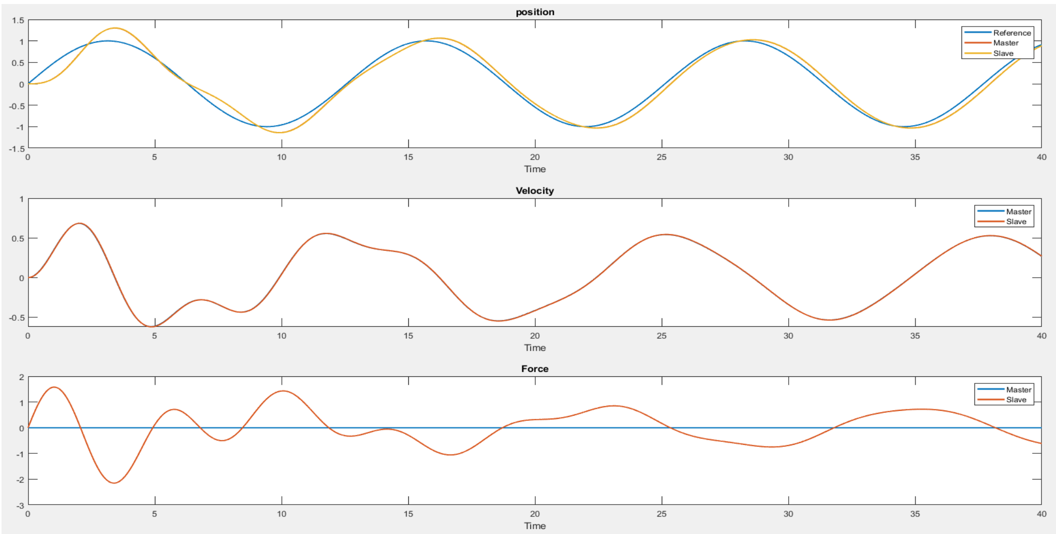

Figure 38: Tank-based architecture PP in free motion

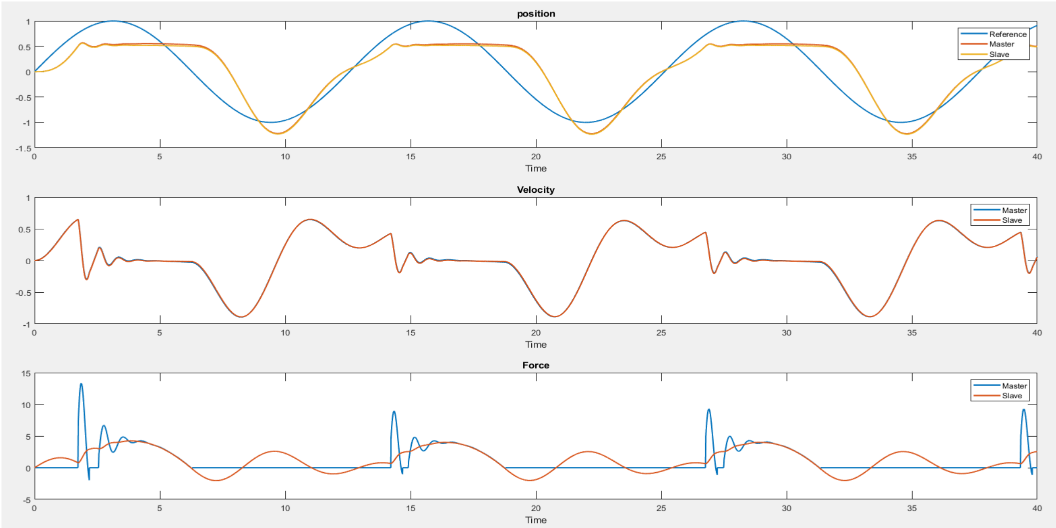

Figure 39: Tank-based architecture PP in contact

$$F_c = 0.5 \quad A = 1 \quad H_{initm} = 3 \quad H_{inits} = 0$$

$$P_h = 5 \quad I_h = 2 \quad env = 0.5$$

$$B_m = 250 \quad k_m = 80 \quad M_m = 0.5$$

$$B_s = 200 \quad k_s = 80 \quad M_s = 2$$

$$J_h = 0 \quad B_h = 1.5 \quad K_h = 1$$

$$J_e = 0 \quad B_e = 10 \quad K_e = 200$$

$$beta = 0.1 \quad alpha = 0.1 \quad H_D = 1 \quad delay = 1$$

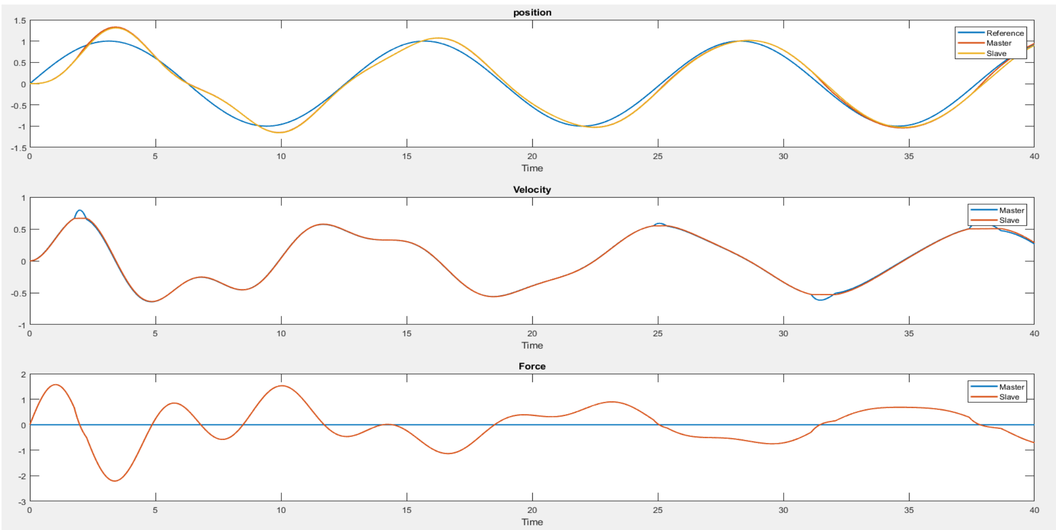

Figure 40: Tank-based architecture PP in free motion with less initial energy

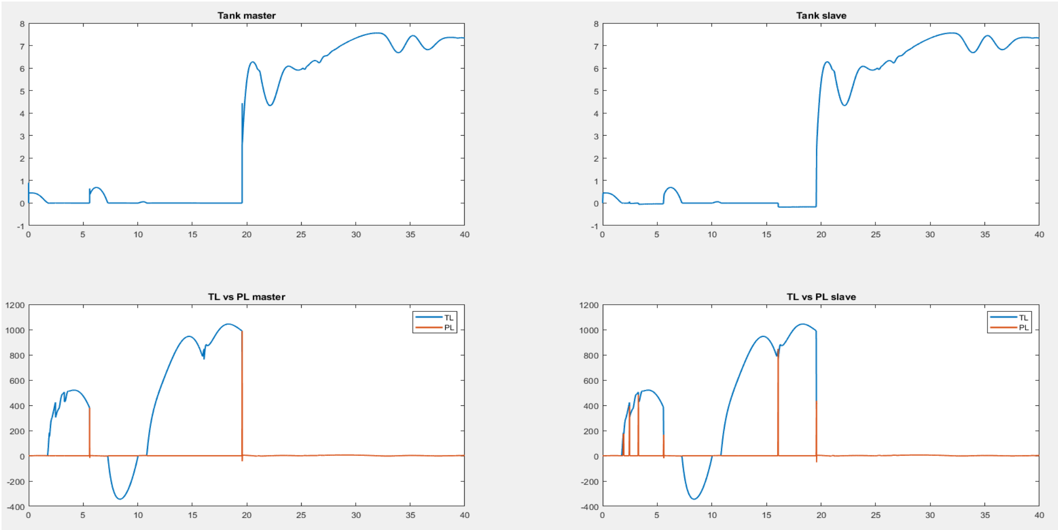

Figure 41: Tank level and applied torque of Tank-based architecture PP in free motion with less initial energy

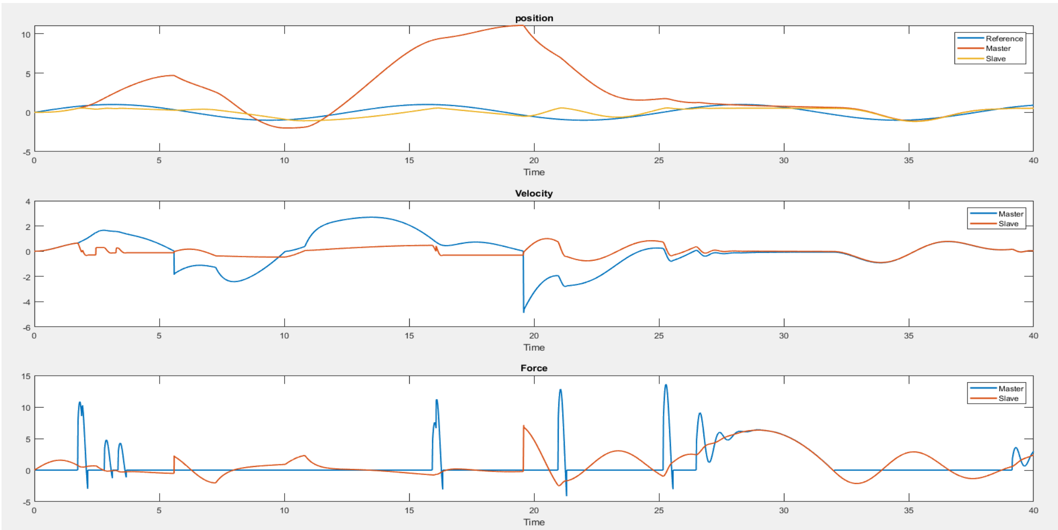

Figure 42: Tank-based architecture PP in contact with less initial energy

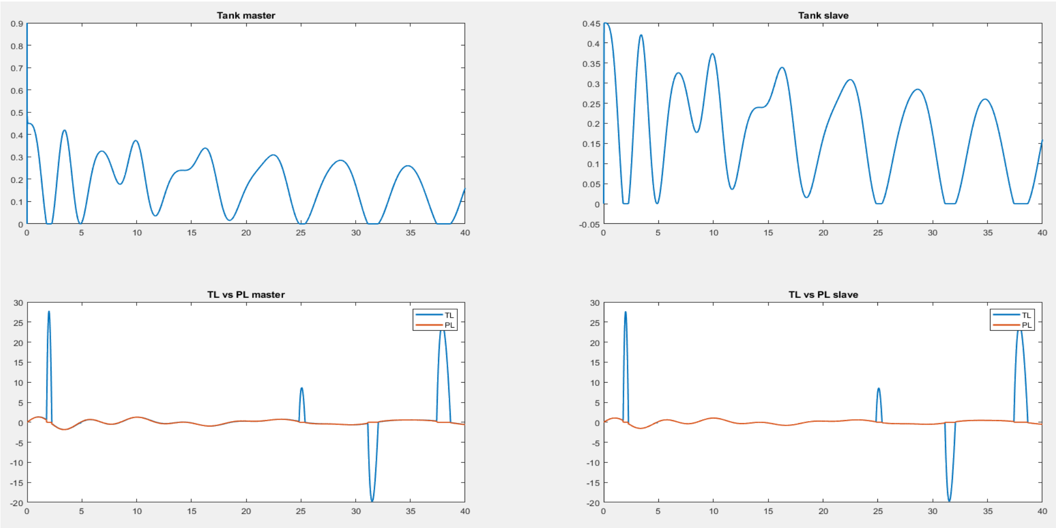

Figure 43: Tank level and applied torque of Tank-based architecture PP in contact with less initial energy

Furthermore, we have evaluated how much it is robust to the delay. So, same parameters but delay = 10.

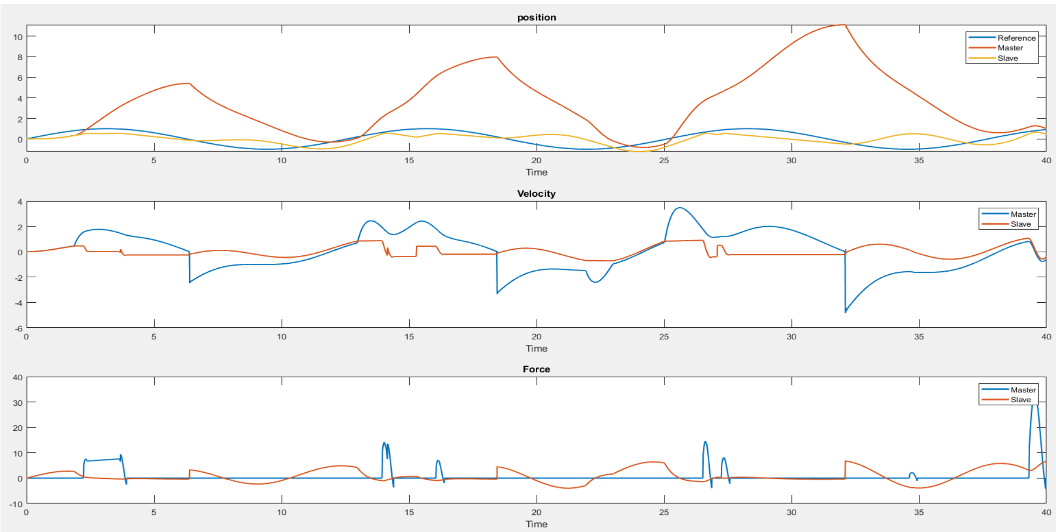

Figure 44: Tank-based architecture PP in contact with less initial energy and more delay

Moreover, we created another model adding the measurement noise to the position/force signals, and estimate velocities from positions (using the previous methods).

$$\begin{array}{c} F_c = 0.5 \\ A = 1 \\ H_{initm} = 10 \\ H_{inits} = 10 \end{array} \quad \begin{array}{c} P_h = 5 \\ I_h = 2 \\ env = 0.5 \end{array} \quad \begin{array}{c} B_m = 250 \\ k_m = 80 \\ M_m = 0.5 \end{array} \quad \begin{array}{c} B_s = 200 \\ k_s = 80 \\ M_s = 2 \end{array} \quad \begin{array}{c} J_h = 0 \\ B_h = 1.5 \\ K_h = 1 \end{array} \quad \begin{array}{c} J_e = 0 \\ B_e = 10 \\ K_e = 200 \end{array} \quad \begin{array}{c} q = 100000000 \\ Q = q \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix} \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix}^T \\ R = 1 \end{array}$$

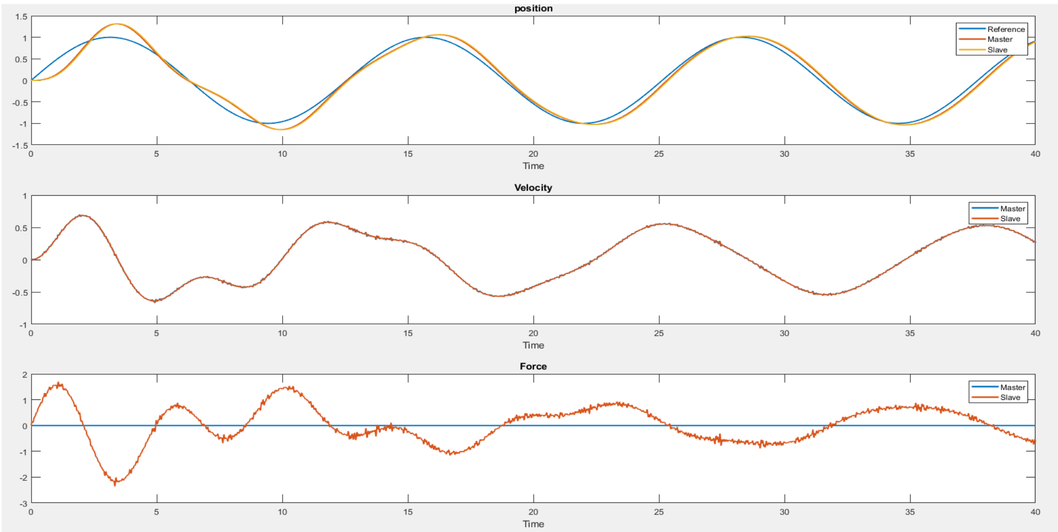

Figure 45: Tank-based architecture PP in free motion with noise only at master side

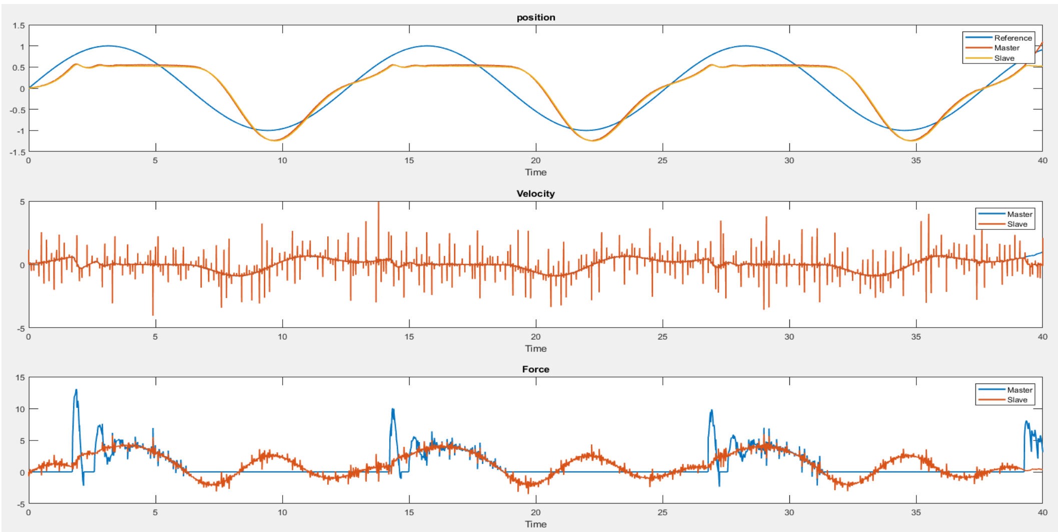

Figure 46: Tank-based architecture PP in contact with noise both master and slave side