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

Physical Human-Robot Interaction

Report on the assignments given during the 2022/2023 a.y.

Author: Lorenzo Busellato, VR472249
email: lorenzo.busellato_02@studenti.univr.it

UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Contents

1 Assignment 1

1.1 Implement the single-input single-output four-channel bilateral teleoperation architecture

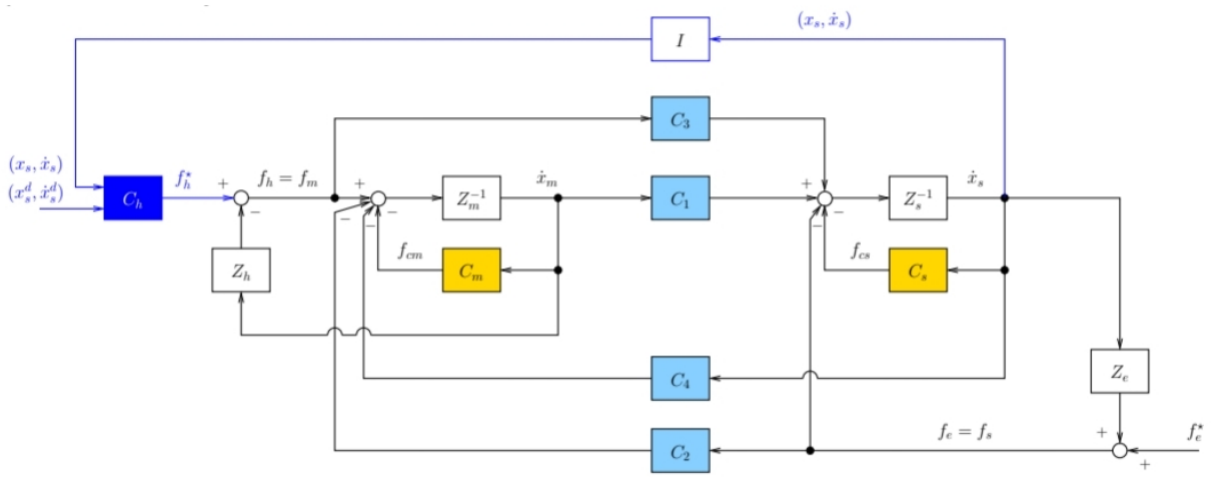

Figure 1: SISO 4ch bilateral teleoperation architecture

The inner controllers for the robots are defined as follows:

$$C_m = B_m + \frac{K_m}{s} \qquad C_s = B_s + \frac{K_s}{s}$$

To achieve perfect transparency, the coordination controllers must satisfy the following conditions:

$$\begin{cases} C_1 &= Z_s + C_s \\ C_2 &= I \\ C_3 &= I \\ C_4 &= -(Z_m + C_m) \end{cases} \qquad \text{with } Z_m = M_m s + D_m,\ Z_s = M_s s + D_s$$

with $M_m = 0.5, M_s = 2, D_m = D_s = 0$.

The human is modelled with damping $B_h = 1$ and stiffness $K_h = 0$. The human intention is modelled with a PD controller with $K_P = 2000$ and $K_D = 50$. The master robot is controlled with a PD controller with $K_P = 20$ and $K_D = 10$. The slave robot is controlled with a PD controller with $K_P = 4000$ and $K_D = 500$.

The environment is modelled as a spring with inertia $B_e = 100$ and stiffness $K_e = 200$.

The architecture is tested with a low-pass filtered step signal ($f_{lp} = 0.5$Hz, $A = 1$) and with a sinusoidal signal ($A = 1$, $f = 1$Hz), both in free motion ($x_e = 1.5$) and in contact ($x_e = 0.75$). The architecture is tested both with $D_m = D_s = 0$ and $D_m = 5, D_s = 10$.

The architecture is modeled in simulink as follows:

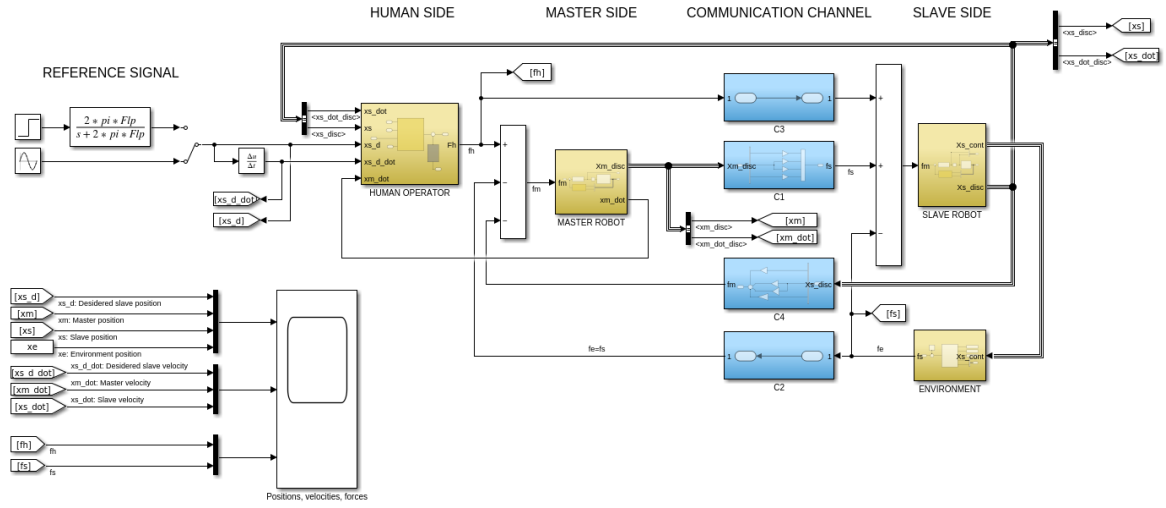

Figure 2: SISO 4ch bilateral teleoperation architecture - SIMULINK model

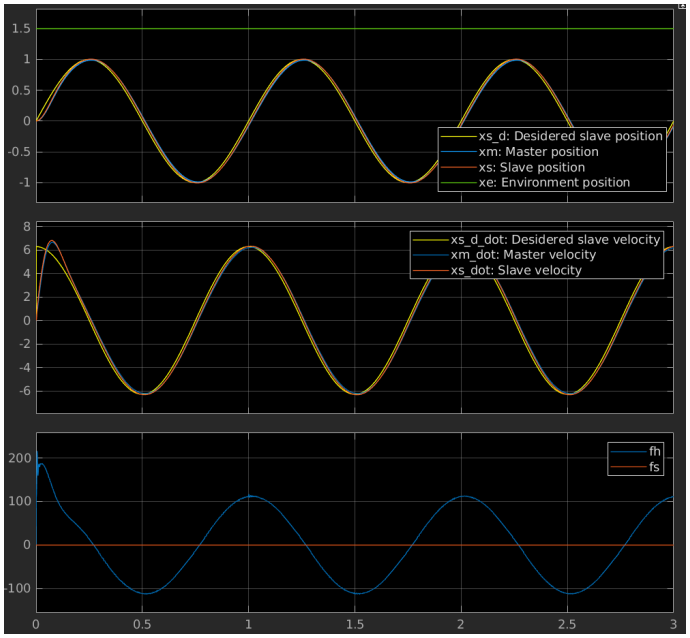

Figure 3: Sinusoidal response in free motion with $Dm = Ds = 0$

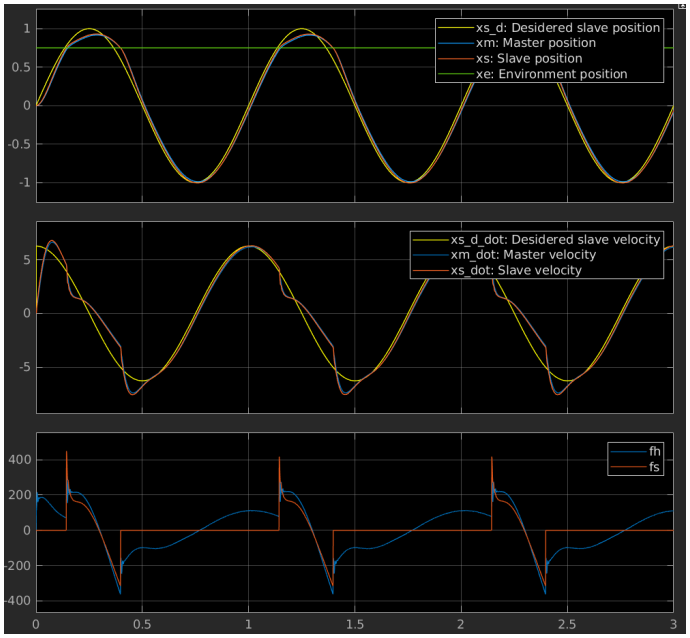

Figure 5: Sinusoidal response in contact with $Dm = Ds = 0$

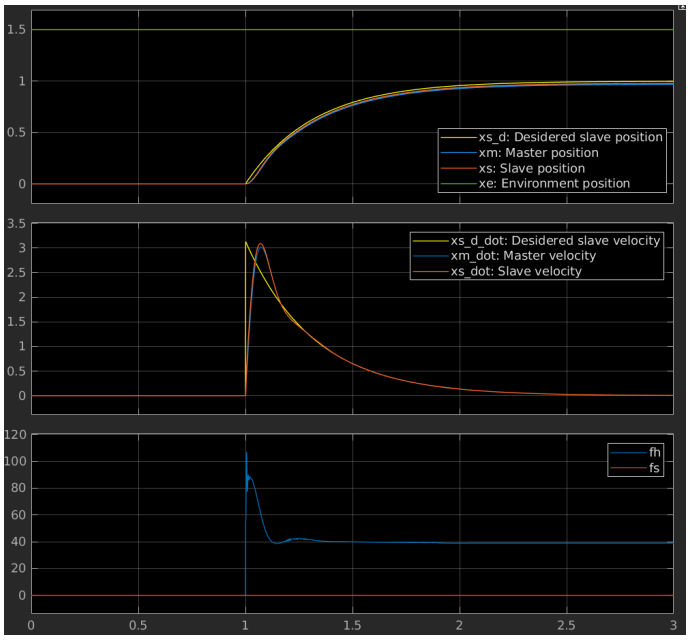

Figure 4: Step response in free motion with $Dm = Ds = 0$

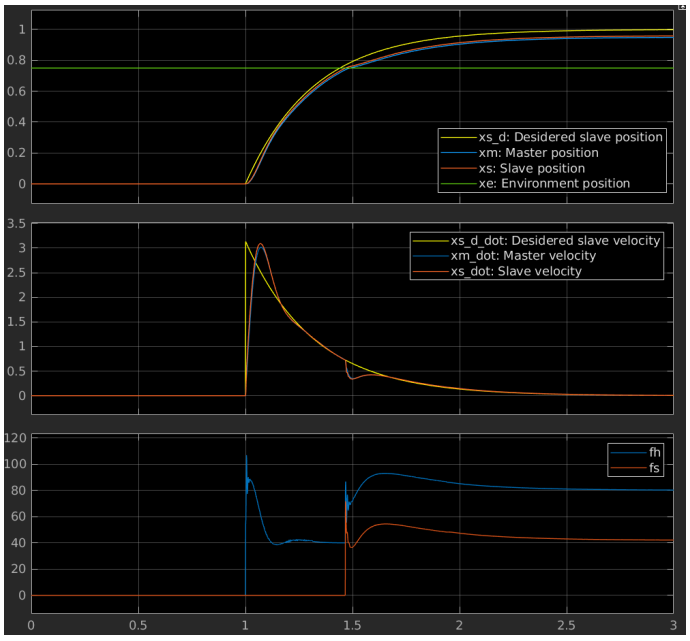

Figure 6: Step response in contact with $Dm = Ds = 0$

In both the sinusoidal and step reference cases, the trajectory is followed nicely in free motion (figures 3 and 4). In contact (figures 5 and 6), once the slave reaches the environment there are sharp spikes in the force of both slave and master, and the trajectory is no longer followed during contact.

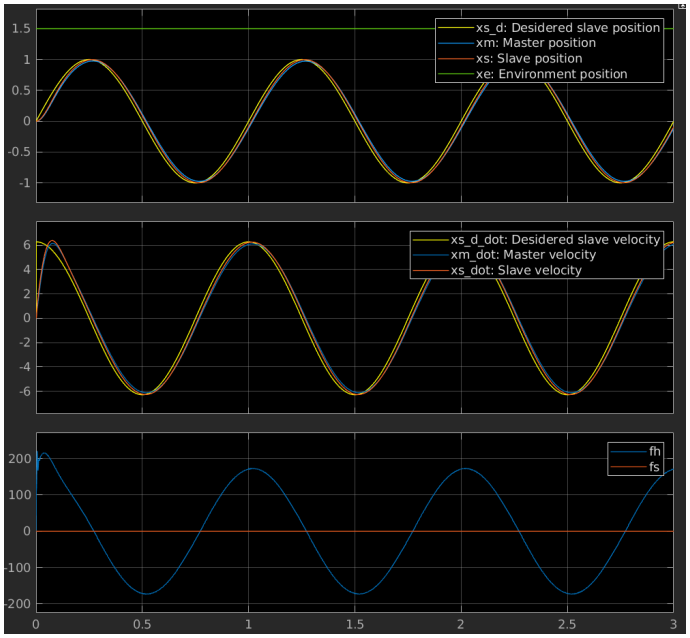

Figure 7: Sinusoidal response in free motion with $Dm = 5, Ds = 10$

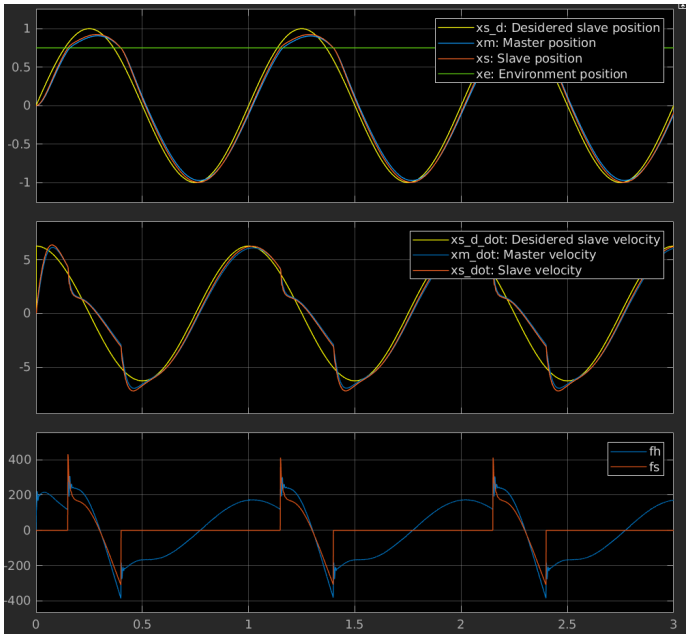

Figure 9: Sinusoidal response in contact with $Dm = 5, Ds = 10$

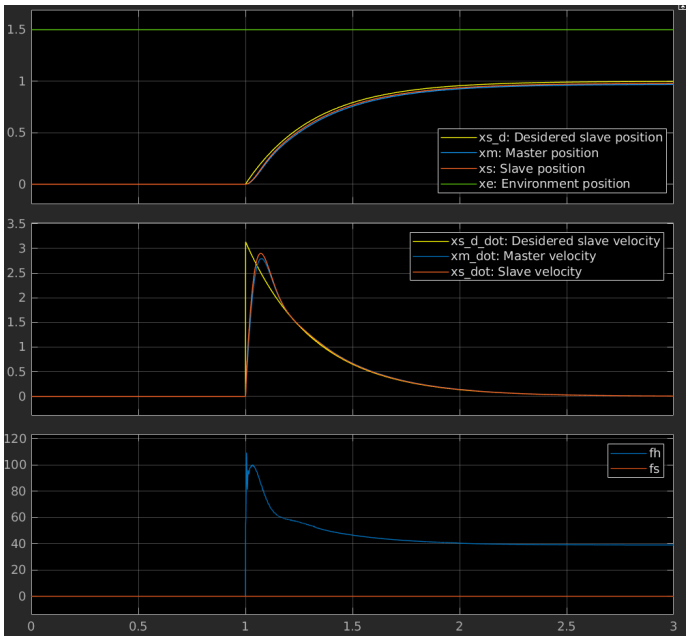

Figure 8: Step response in free motion with $Dm = 5, Ds = 10$

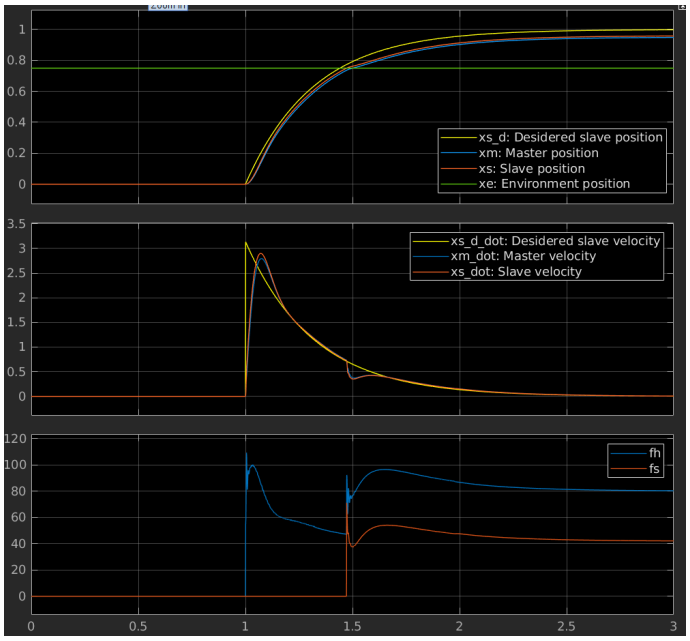

Figure 10: Step response in contact with $Dm = 5, Ds = 10$

Using damping values of $D_m = 5, D_s = 10$ doesn't seem to produce a noticeable difference in the execution of the trajectory, both in the free motion (figures 7 and 8) and contact (figures 9 and 10) cases.

2 Assignment 2

2.1 Implement the three two-channel bilateral teleoperation architectures and the three-channel bilateral teleoperation architecture

The three two-channel architectures and the three channel architecture are obtained by removing some of the coordinating controllers:

- 2-channel position-position (P-P): $C_2 = C_3 = 0$
- 2-channel force-position (F-P): $C_3 = C_4 = 0$
- 2-channel force-force (F-F): $C_1 = C_4 = 0$
- 3-channel position and force-position (P,F-P): $C_3 = 0$

The four architectures are modelled with the same SIMULINK model, which is based on the 4-channel architecture, with the added possibility of individually cutting off the coordinating controllers:

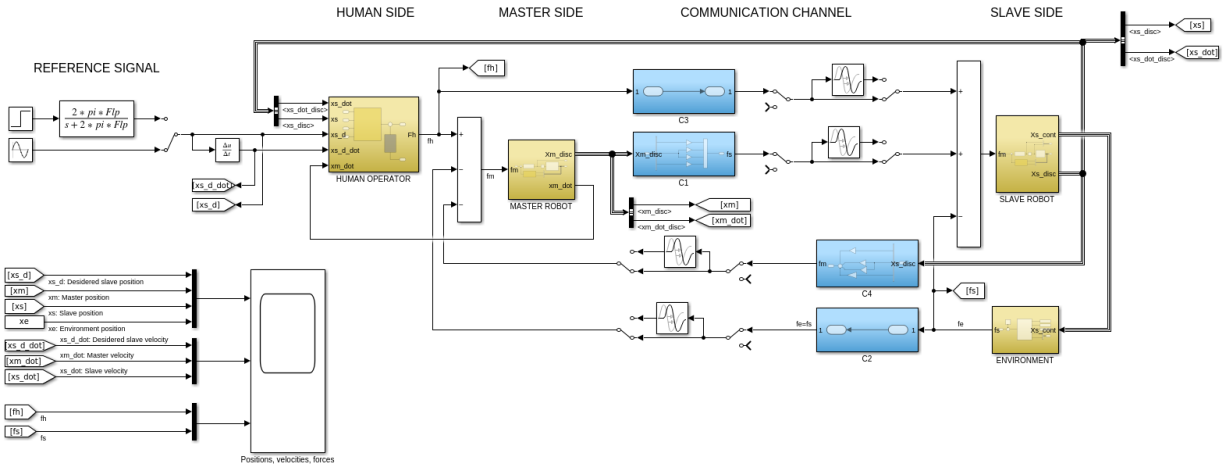

Figure 11: 2-3 channel teleoperation architecture with transport delays - SIMULINK model

All architectures are tested with a sinusoidal position reference signal, both in the free motion and the contact cases.

All the architectures follow the reference trajectory without issues in free motion (figures 12 through 15). The resulting forces seem to be the smallest in the F-P architecture and the greatest in the F-F architecture, while they are about the same in the P-P and P,F-P architectures.

During contact (figures 16 through 19), all the architectures lose the reference trajectory, with the presence of sharp force spikes at the moments of contact and release.

2.2 What happens if transportation delays are added in series to the coordinating controllers?

The same SIMULINK model has the capability of adding time delays to the communication channel. All the architectures have been tested with varying amounts of delay. In figures 20 through 23 are reported the position, velocity and forces plots with a time delay $d = 20$ms. It is observed that by increasing the delay results in an increasingly unstable system behaviour. All architectures' position, velocity and force signals start to oscillate around the respective reference signals with amplitude and frequency that increase as the time delay is increased, until around $d = 20$ms, after which the signals start to diverge.

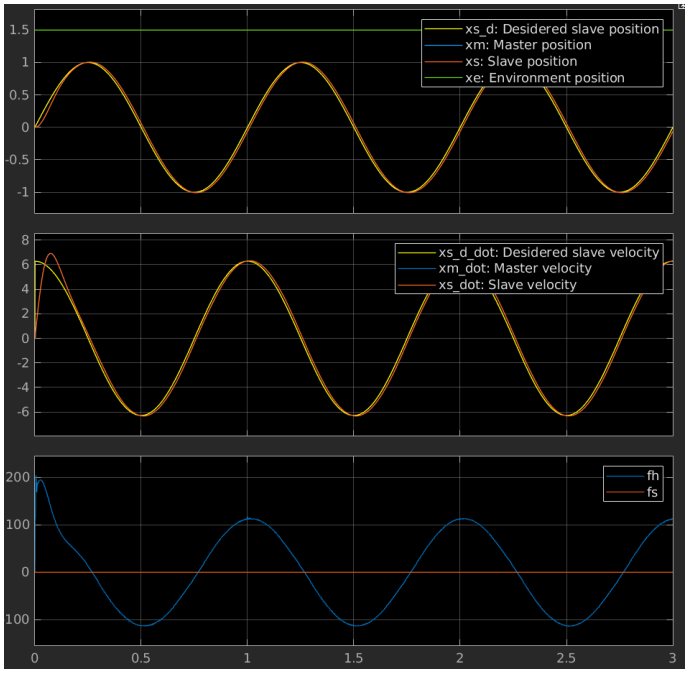

Figure 12: 2-channel P-P in free motion.

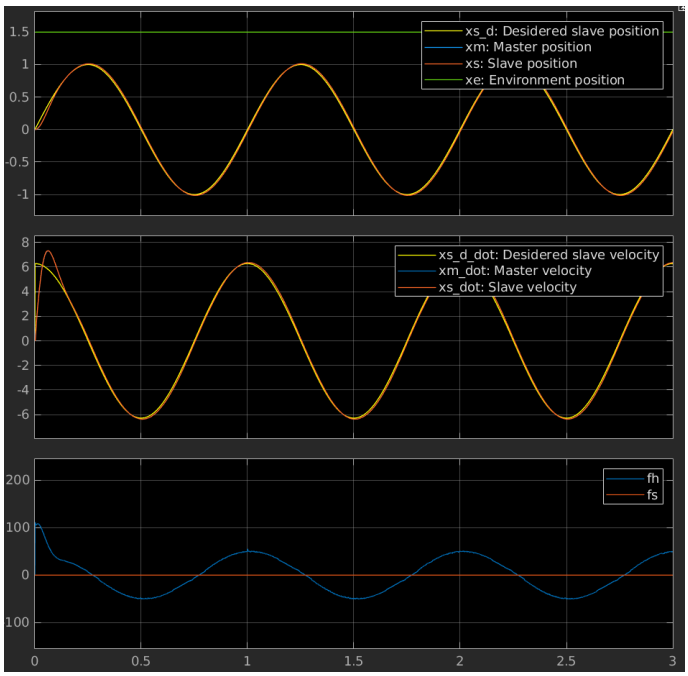

Figure 13: 2-channel F-P in free motion.

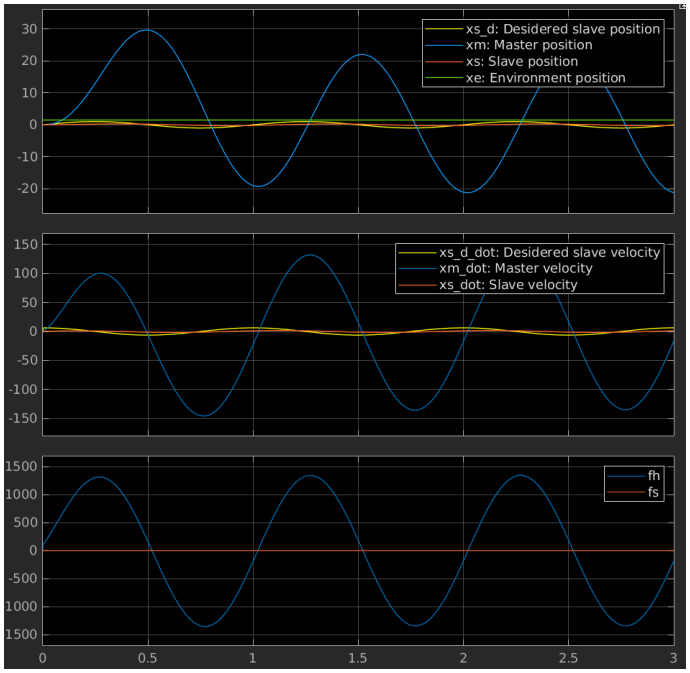

Figure 14: 2-channel F-F in free motion.

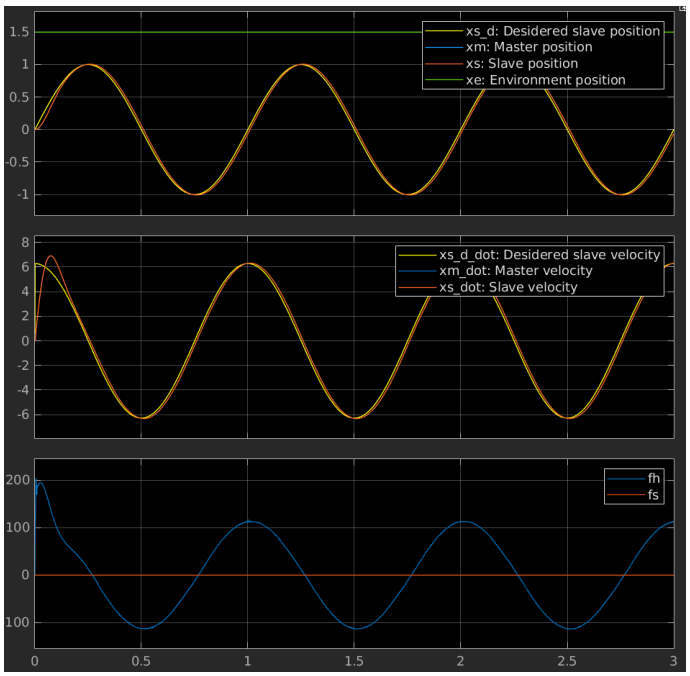

Figure 15: 3-channel P,F-P in free motion.

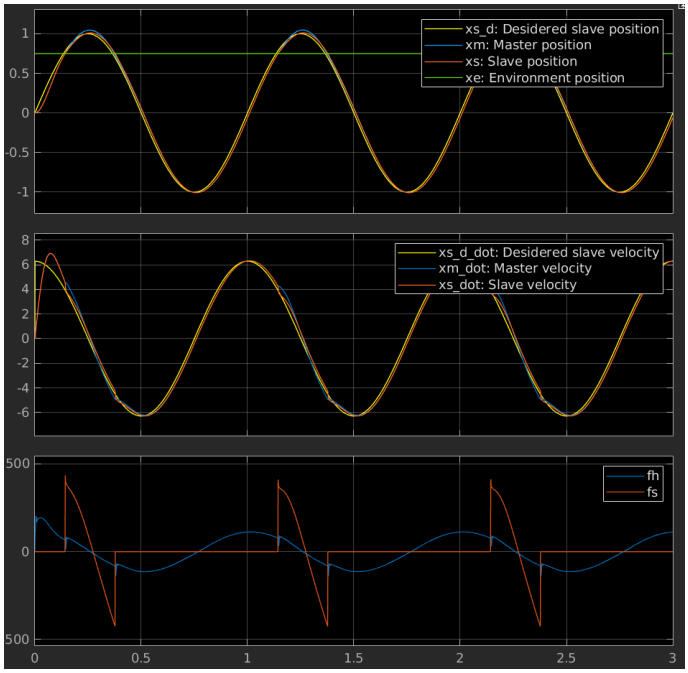

Figure 16: 2-channel P-P in contact.

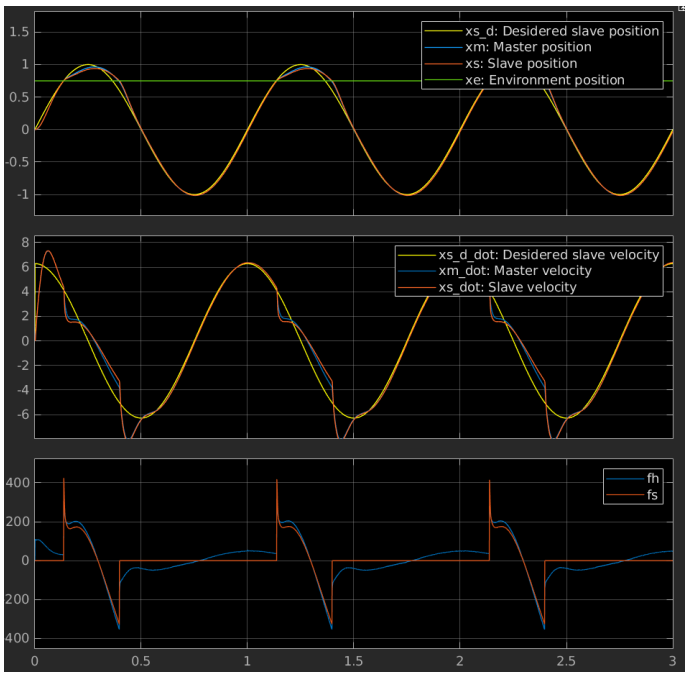

Figure 17: 2-channel F-P in contact.

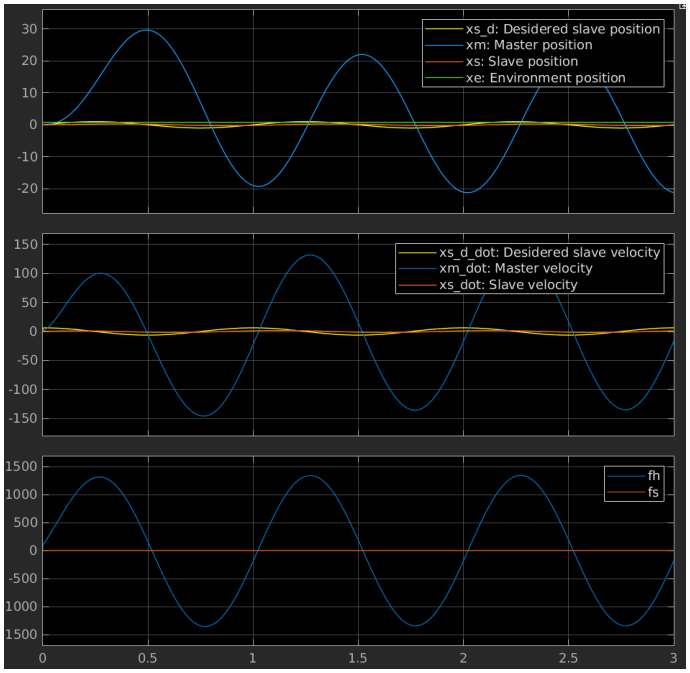

Figure 18: 2-channel F-F in contact.

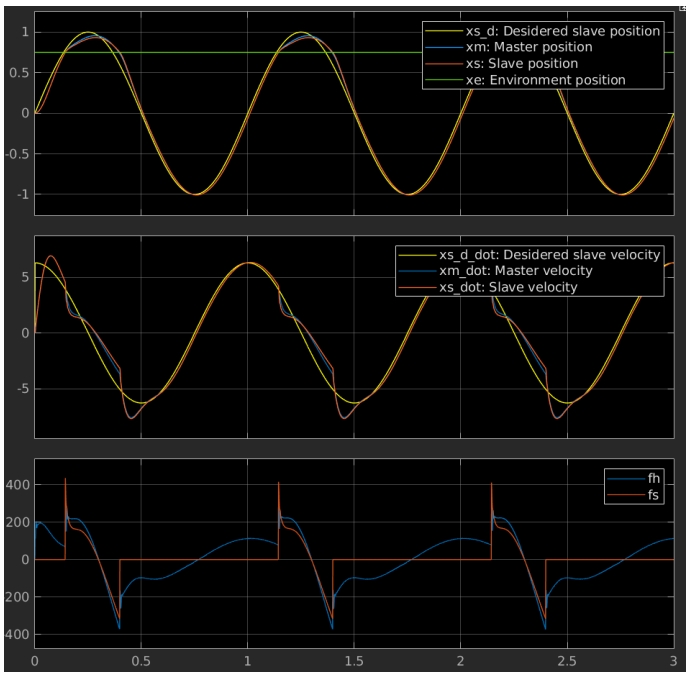

Figure 19: 3-channel P,F-P in contact.

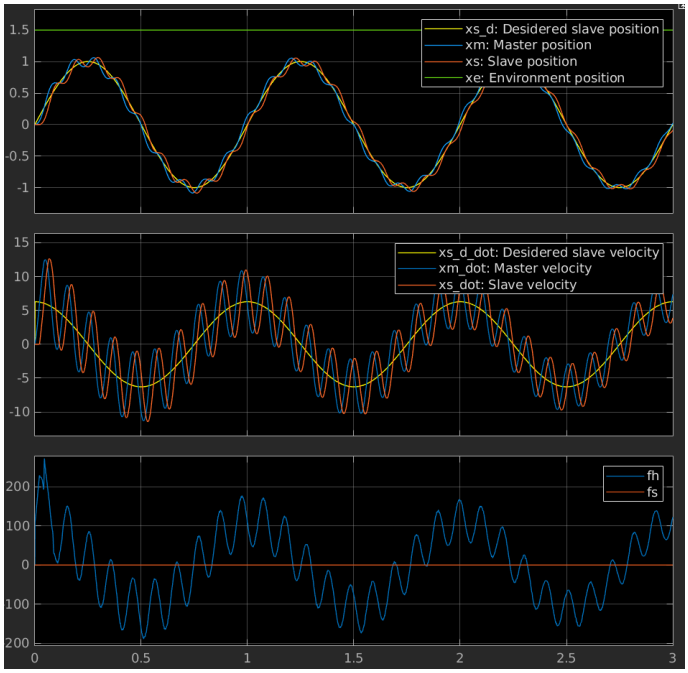

Figure 20: 2-channel P-P in free motion with transportation delays.

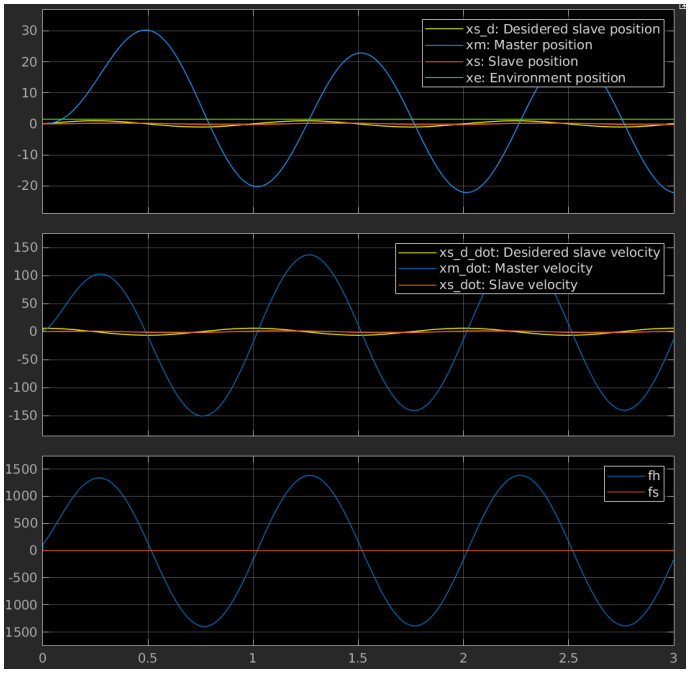

Figure 22: 2-channel F-F in free motion with transportation delays.

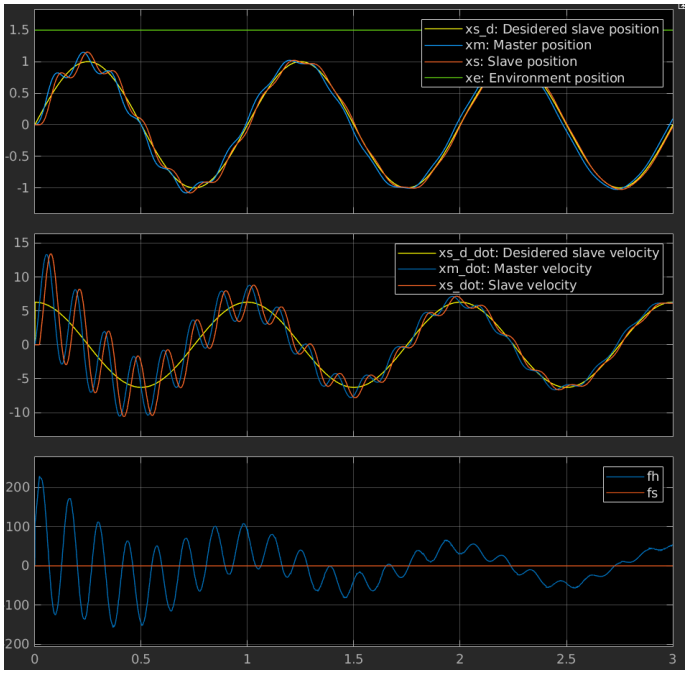

Figure 21: 2-channel F-P in free motion with transportation delays.

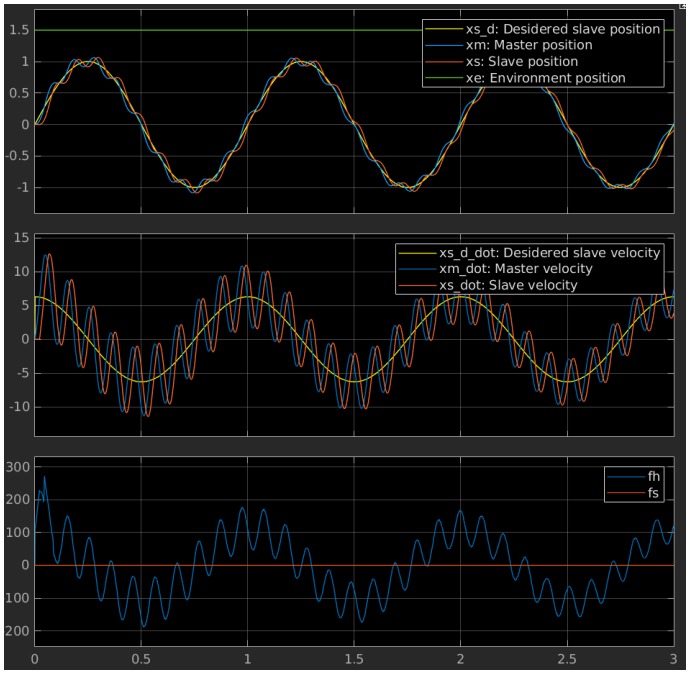

Figure 23: 3-channel P,F-P in free motion with transportation delays.

3 Assignment 3

3.1 Implement the Kalman filter/predictor and estimate the velocity and acceleration from noisy position measurements

Both the Kalman filter and the Kalman predictor are based on the stochastic LTI system:

$$\begin{cases} \mathbf{x}_{k+1} = A\mathbf{x}_k + Bw_k \\ y_k = C\mathbf{x}_k + v_k \end{cases} \qquad \mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad A = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \frac{T_s^3}{6} \\ \frac{T_2}{2} \\ T_s \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

and w_k, v_k zero mean uncorrelated Gaussian random variables describing respectively model and noise variances:

$$w_k = \mathcal{N}(0, Q) \quad v_k = \mathcal{N}(0, R)$$

with $Q = qBB^T$, $R = var(x)$, $q = 10^7$ and initial state $x_0 \approx \mathcal{N}(\bar{x}_0, P_0)$, $\bar{x}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, $P_0 = \mathbb{I}_3 \cdot 1 \cdot 10^{-5}$.
The Kalman filter is recursively defined as follows:

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

where P is the variance of the estimation error and the Kalman gain (the matrix that maps the output estimation error into the correction of the state) defined as:

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

The Kalman predictor is recursively defined as follows:

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

with the Kalman gain defined as:

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

The difference between filter and predictor is that the former estimates the state $\hat{x}$ at time $k+1$ given measurements until time $k+1$, while the latter estimates the state $\hat{x}$ at time $k+1$ given measurements until time k (1-step ahead prediction). Therefore the resulting estimations will differ only in the first state. Both filter and predictor are applied to a position signal with Gaussian noise and a signal with quantization noise. The results are shown in figures 24 and 25.

3.2 Implement the steady-state Kalman filter/predictor and estimate the velocity and acceleration from noisy position measurements

The steady state Kalman filter and predictor are obtained by computing a fixed solution for the estimation error variance P_∞ and for the Kalman gain, K_∞. The estimation error variance for $k \to \infty$ is obtained by solving the Algebraic Riccati Equation:

$$P_\infty = AP\infty A^T - AP_\infty C^T(CP_\infty C^T + R)^{-1}CP_\infty A^T + Q$$

Then, the Kalman gains for the filter and predictor are defined as:

$$K_\infty = P_\infty C^T(CP_\infty C^T + R)^{-1} \quad \bar{K}_\infty = AP_\infty C^T(CP_\infty C^T + R)^{-1}$$

Finally, the steady state Kalman filter is defined as:

$$\hat{x}_{k+1|k+1} = A\hat{x}_{k|k} + K_\infty(y_{k+1} - CA\hat{x}_{k|k})$$

while the steady state Kalman predictor is defined as:

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + \bar{K}_\infty(y_k - C\hat{x}_{k|k-1})$$

Both filter and predictor are applied to a position signal with Gaussian noise and a signal with quantization noise. The results are shown in figures 24 and 25.

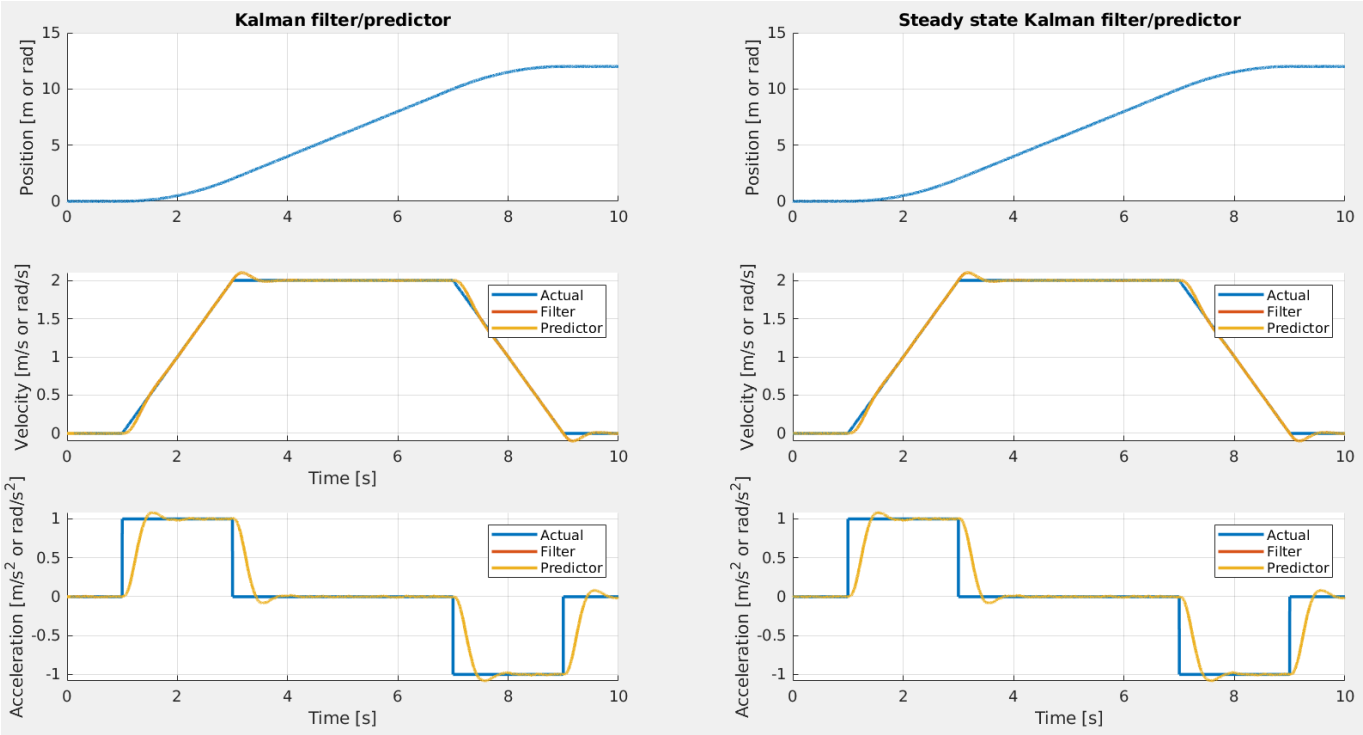

Figure 24: Kalman filter/predictor applied to a position signal with Gaussian noise

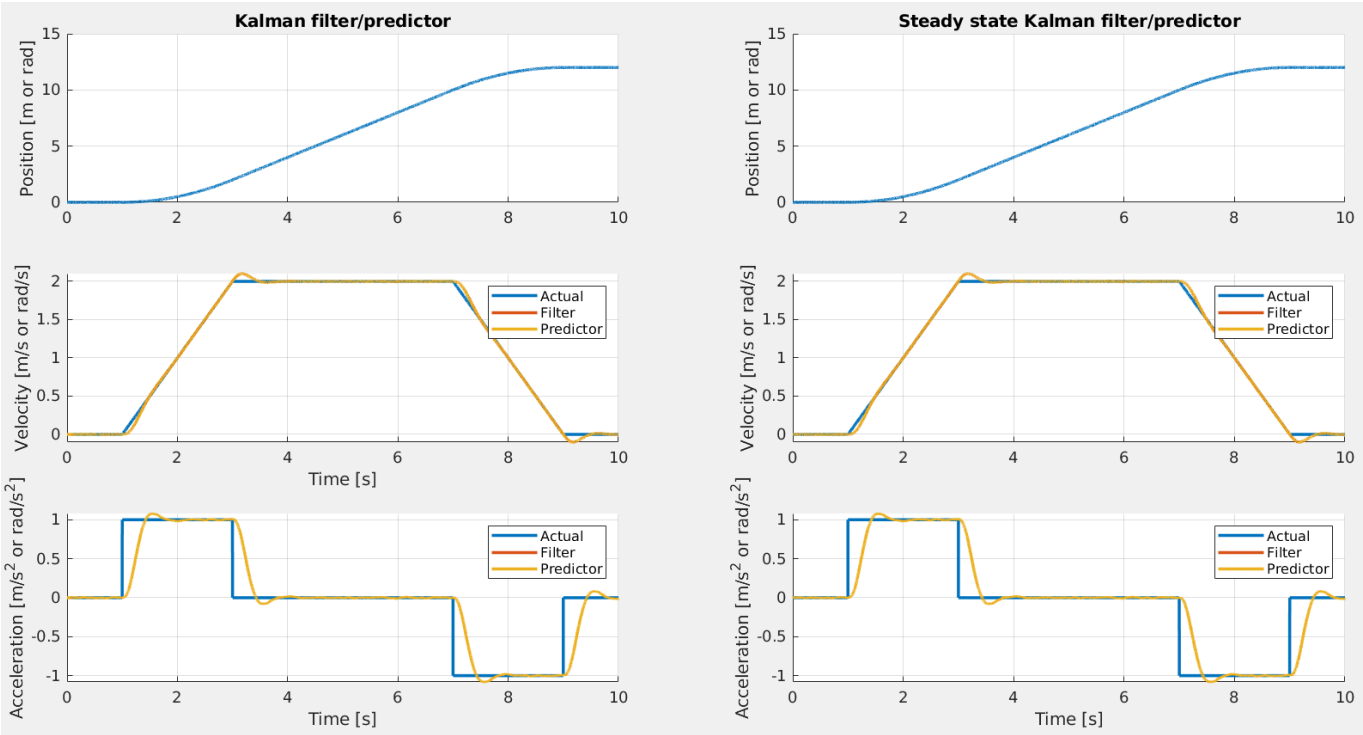

Figure 25: Kalman filter/predictor applied to a position signal with quantization noise

As expected, both filter and predictor produce the same estimations, and there's no difference with their respective steady-state implementations. Both the Gaussian and quantization noise cases produce very similar estimates.

4 Assignment 4

4.1 Implement the Kalman smoother and estimate the velocity and acceleration from noisy position measurements

Starting with the same model described in assignment 3, the Kalman smoother is defined in two parts. a forward step consisting of standard Kalman filtering and a backward step in which the smoothing takes place.

In the forward step, a standard Kalman filtering procedure is carried out:

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

Then in the backward step, smoothing is carried out as follows:

$$\begin{aligned}
\hat{x}^s_{k|N} &= \hat{x}^f_{k|k} + \breve{K}_k(\hat{x}^s_{k+1|N} - \hat{x}^f_{k+1|k}) \\
\hat{x}^s_{N|N} &= \hat{x}^f_{N|N}
\end{aligned}$$

where:

$$\breve{K}_k = P^f_{k|k}A^T(P^f_{k+1|k})$$

and the conditional covariance matrix $P_{k|N}$ satisfies:

$$\begin{aligned}
P_{k|N} &= P^f_{k|k} + \breve{K} - k(P_{k+1|N} - P^f_{k+1|k}) \\
P_{N|N} &= P^f_{N|N}s
\end{aligned}$$

The smoother is applied to a position signal with Gaussian noise and a signal with quantization noise, and compared to the estimation given by the Kalman filter. The results are shown in figures 26 and 27.

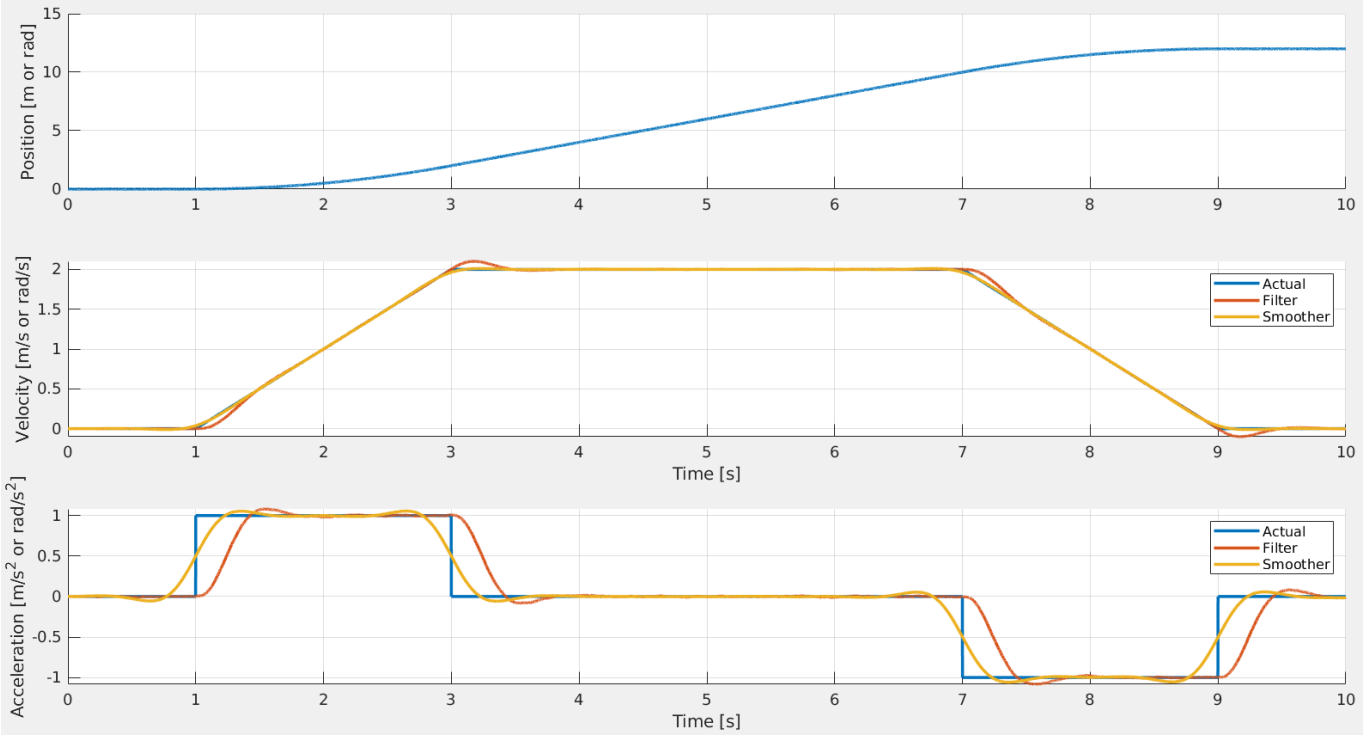

Figure 26: Kalman smoother vs Kalman filter applied to a position signal with Gaussian noise

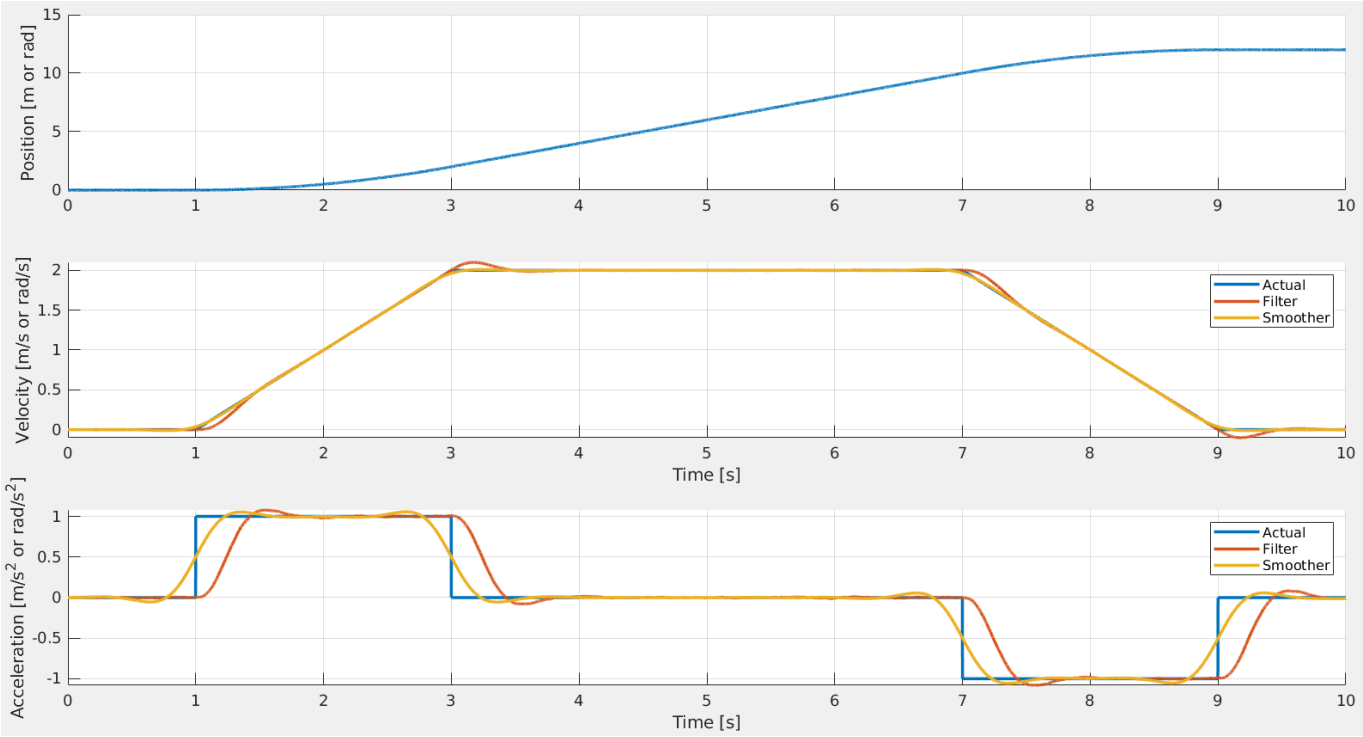

Figure 27: Kalman smoother vs Kalman filter applied to a position signal with quantization noise

The smoother effectively improves the estimation of the velocity, especially near the discontinuity points, and of the acceleration, making the shape of the estimation more closely match the "actual" acceleration.

5 Assignment 5

5.1 Identify the parameters J and D using the LS and the RLS on the DC motors data

6 Assignment 6

6.1 Implement the Scattering-based bilateral teleoperation architecture for the F-P and P-P cases

6.2 Compare positions, velocities, forces, commands in free motion and in contact

6.3 Create another simulink model and (a) add the measurement noise to the position/force signals, and (b) estimate velocities from positions

7 Assignment 7

7.1 Implement the Tank-based bilateral teleoperation architecture for the F-P and P-P cases

7.2 Compare positions, velocities, forces, commands in free motion and in contact

7.3 Create another simulink model and (a) add the measurement noise to the position/force signals, and (b) estimate velocities from positions