Physical Human Robot Interaction

Filippo Grotto VR460638

November 19, 2021

Contents

1 Four channel bilateral teleoperation architecture

1.1 HW1: Continuous and discretized implementation

Implement the SISO Four-channel bilateral teleoperation architecture with

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

$$Z_m^{-1} = \frac{1}{M_m s + D_m} \quad Z_s^{-1} = \frac{1}{M_s s + D_s}$$

where $Mm = 0.5$, $M_s = 2$. Moreover $D_s = 10$ and $D_m = 5$ or both zero in the initial case. In Fig 1 a simple plot of the positions and velocities (slave and master) are reported for proper selected tuning parameters of the related master and slave controller. In order to properly tune the controllers the following closed-loop systems were considered:

$$G_m = \frac{1}{M_m s^2 + B_m s + K_m} \quad G_s = \frac{1}{M_s s^2 + B_s s + K_s}$$

The proper parameters were selected considering the step reponse of the two second order systems. For the human intention controller the parameters were selected by comparing the reference position with the master/slave position and perform proper tuning (an analytical closed-loop system might also be considered for this analysis).

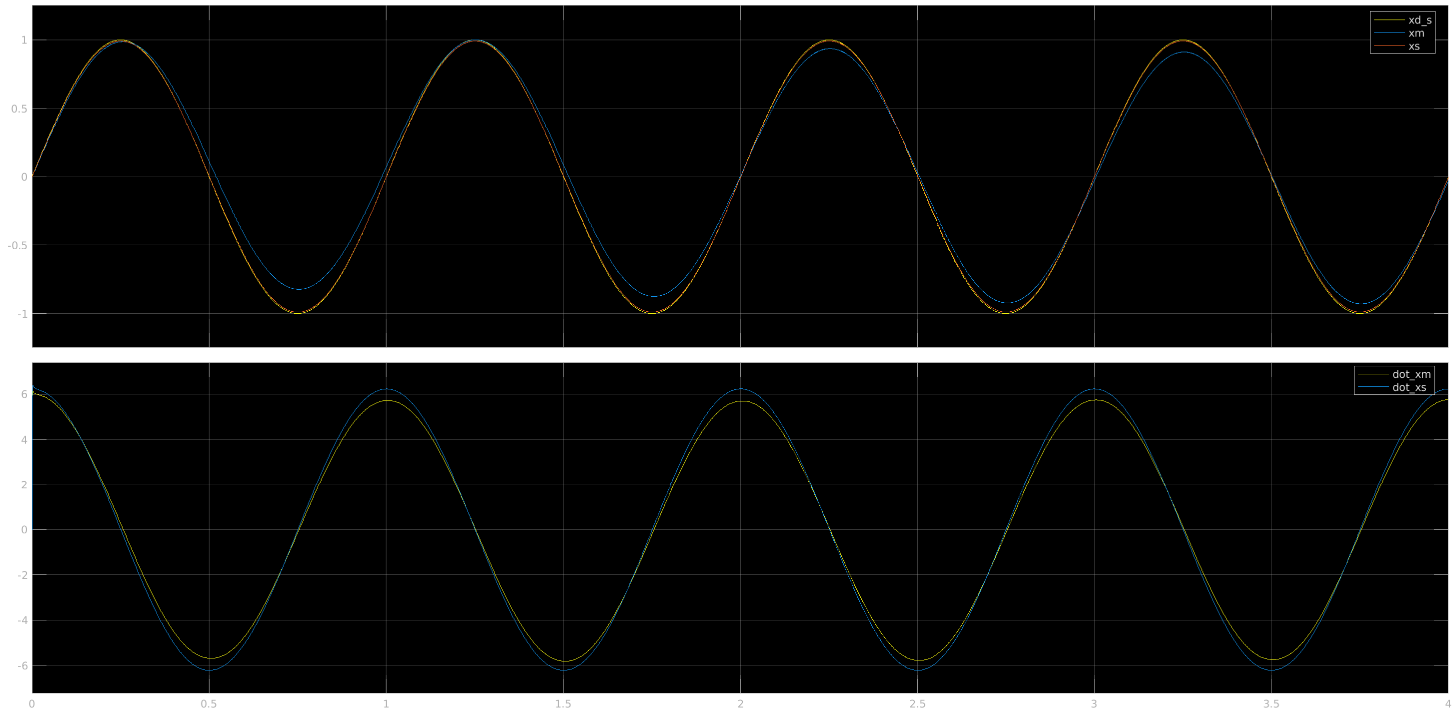

Figure 1: Reference slave and master position are depicted in the first plot. Slave and Master velocities are reported in the second plot

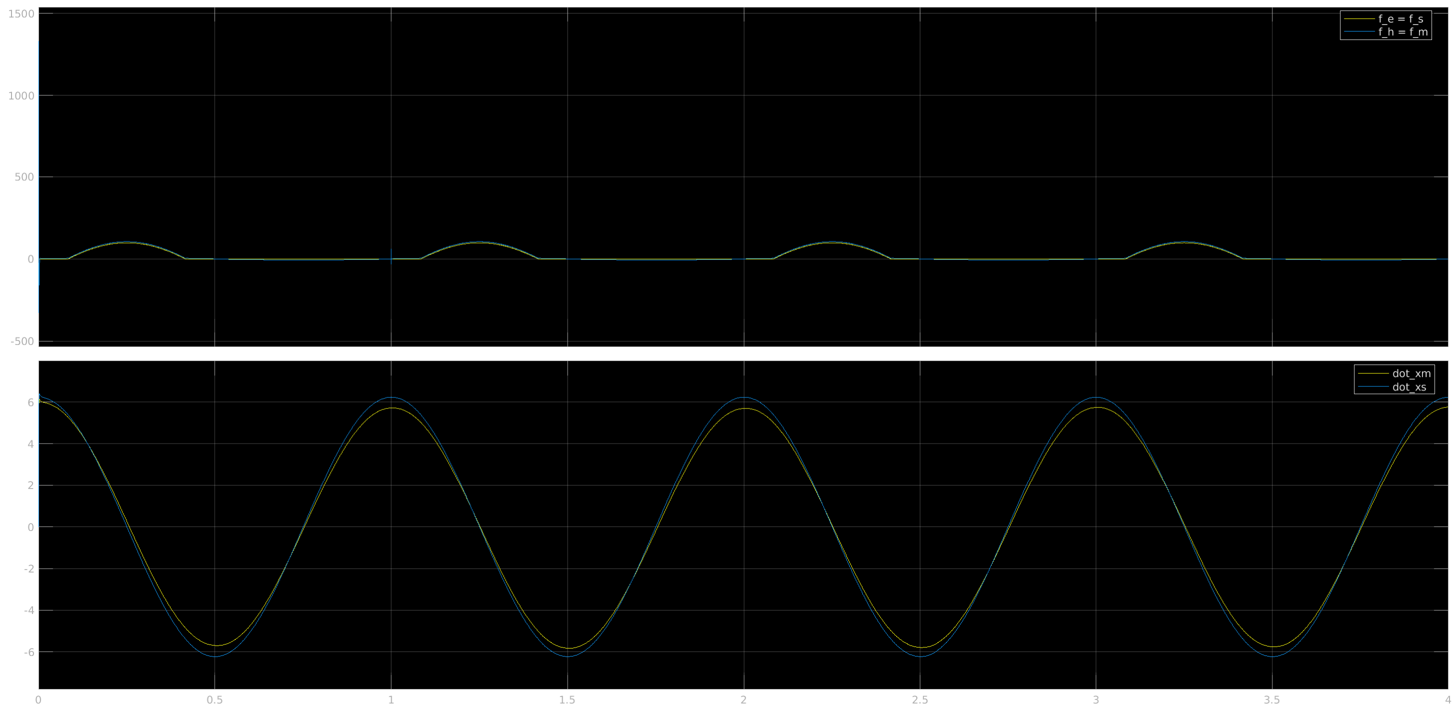

Figure 2: Master and slave velocity and forces compared when the environment is not attached to the end effector of the robot. The environment force has to be zero when not in contact

Finally the entire architecture was translated in the related discretized version according to our specification in terms of encoders and the related derivative were performed using the simulink block. In the following section a proper estimation tool will be analysed.

1.2 HW2: Derive hybrid matrix

Derive the hybrid matrix for the four channel bilateral teleoperation considering the inner force loop at the master and slave side. We will consider $Z_{cm} = C_6$ and $Z_{sm} = C_5$.

$$\begin{bmatrix} f_m \\ -v_s \end{bmatrix} = \begin{bmatrix} \overline{H_{11}} & \overline{H_{12}} \\ \overline{H_{21}} & \overline{H_{22}} \end{bmatrix} = \begin{bmatrix} v_m \\ f_s \end{bmatrix}$$

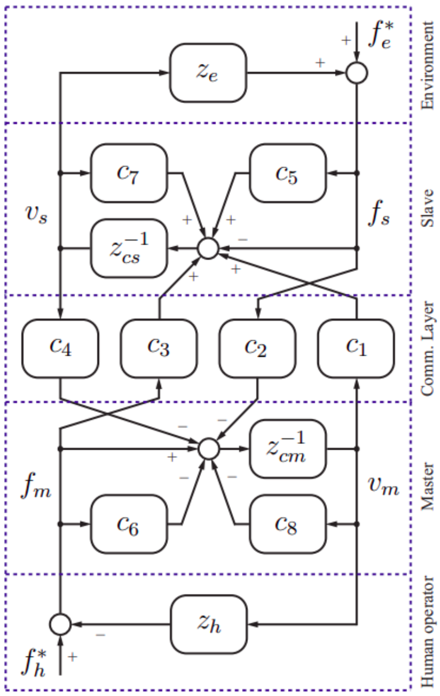

Figure 3: Extended Lawrence Architecture taken from *Global Transparency Analysis of the Lawrence Teleoperator Architecture* Edvard Naerum and Blake Hannaford. In our case C_7 and C_8 are zero

Let's start by defining:

$$v_m = Z_{cm}^{-1}(f_m - C_2 f_s - C_4 v_s + C_6 f_m) \tag{1}$$

$$v_s = Z_{cs}^{-1}(-f_s + C_5 f_s + C_1 v_m + C_3 f_m) \tag{2}$$

Finally let's compute the 4 components of the hybrid matrix considering one component to zero at each step.

$$\overline{H_{11}} : f_m \to v_m \qquad f_s = 0$$

$$Z_{cm}v_m = f_m - C_4Z_{cs}^{-1}C_1v_m - C_4Z_{cs}^{-1}C_3f_m + C_6f_m$$
$$v_m(Z_{cm} + C_4Z_{cs}^{-1}C_1) = f_m(1 - C_4Z_{cs}^{-1}C_3 + C_6)$$
$$\overline{H_{11}} = \frac{Z_{cm}Z_{cs} + C_1C_4}{(1 + C_6)Z_{cs} - C_3C_4}$$

$$\overline{H_{12}} : f_m \to f_s \qquad v_m = 0$$
$$0 = f_m - C_2f_s - C_4Z_{cs}^{-1}C_5f_s + C_4Z_{cs}^{-1}f_s - C_6f_m$$
$$f_m(1 - C_6 - C_4Z_{cs}^{-1}C_3) = f_s(C_2 + C_4Z_{cs}^{-1}C_5 - C_4Z_{cs}^{-1})$$
$$\overline{H_{12}} = \frac{C_2Z_{cs} - C_4(1 + C_5)}{(1 + C_6)Z_{cs} - C_3C_4}$$

$$\overline{H_{21}} : -v_s \to v_m \qquad f_s = 0$$
$$v_sZ_{cs} = -(C_1v_m + C_3f_m)$$
$$v_s(Z_{cs} - \frac{C_3C_4}{1 + C_6} = -v_m(C_1 + \frac{C_3Z_{cm}}{1 + C_6})$$
$$\overline{H_{21}} = -\frac{C_1(1 + C_6) + C_3Z_{cm}}{(1 + C_6)Z_{cs} - C_3C_4}$$

$$\overline{H_{22}} : -v_s \to f_s \qquad v_m = 0$$
$$v_sZ_{cs} = -(C_5f_s - f_s + C_5f_s)$$
$$v_s(Z_{cs} - \frac{C_3C_4}{1 + C_6}) = -f_s(C_5 - 1 + \frac{C_3C_2}{1 + C_6})$$
$$\overline{H_{22}} = \frac{(1 - C_5)(1 + C_6) - C_2C_3}{(1 + C_6)Z_{cs} - C_3C_4}$$

1.3 HW3: Kalman Filter and Predictor

The aim is to estimate the velocities and accelerations from noisy position measurements. Some samples were provided for the task and the kalman filter and predictor were implemented as well as the steady state versions. In Fig 4 and in Fig 5 the estimations of velocities and accelerations are reported. The equations for the kalman filter and predictor are not reported. Let's derive the discretized model for the position, velocities and accelerations to use in the kalman filter:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \qquad y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t) + v(t) \tag{3}$$

Let's discretize the continuous-time model:

$$A_d = e^{AT_s} = e^{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} T_s} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$B_d = \left(\int_0^{T_s} e^{\begin{bmatrix} 1 & \tau & \tau^2/2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}} d\tau \right) = \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix}$$

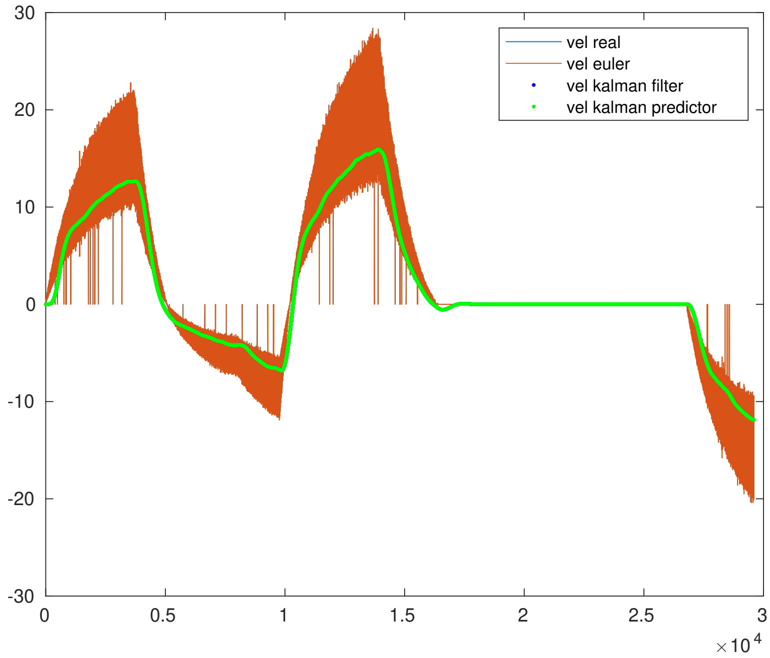

Figure 4: Velocities estimations using kalman filter, kalman predictor and euler against the real values. As expected the euler approximation is much more noisy

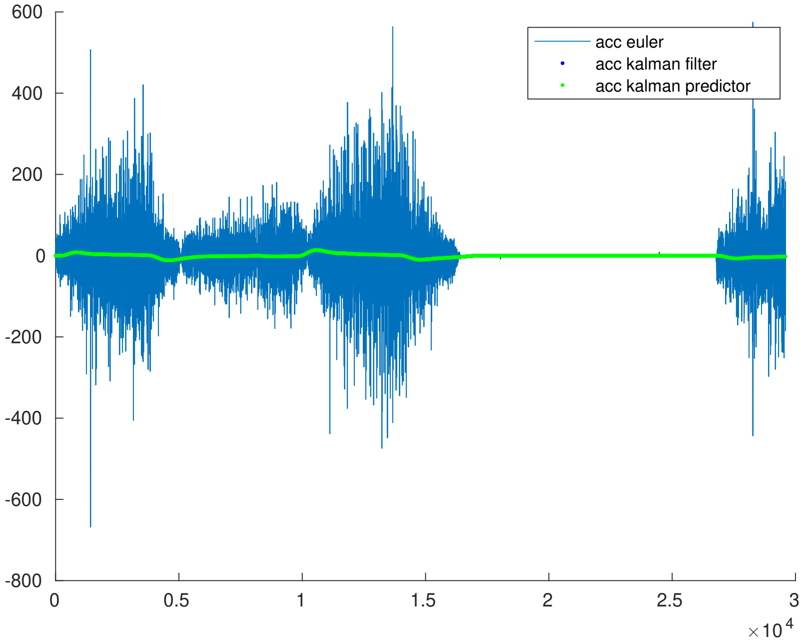

Figure 5: Accelerations estimations using kalman filter, kalman predictor and euler against the real values. As expected the euler approximation is much more noisy

Finally the results obtained using the kalman smoother (offline estimations) for velocities and accelerations:

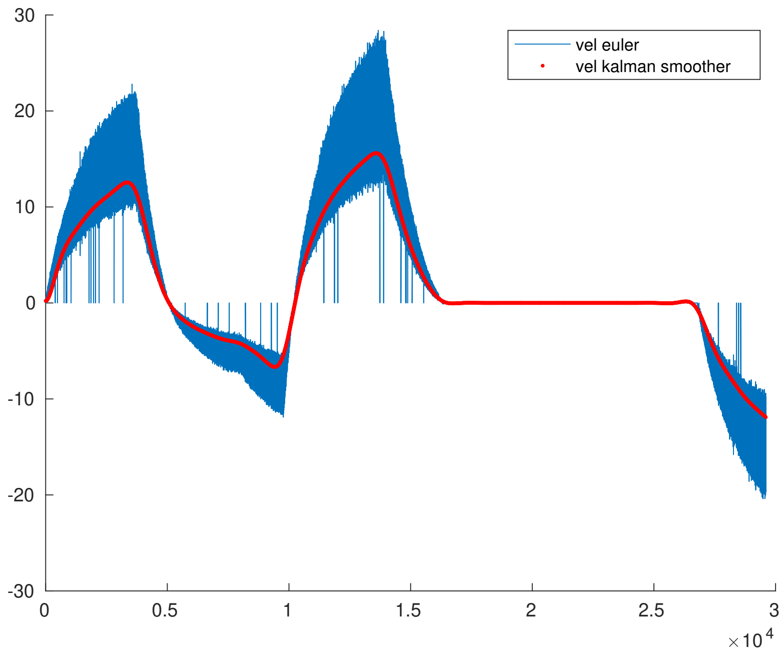

Figure 6: Velocities estimations using kalman smoother

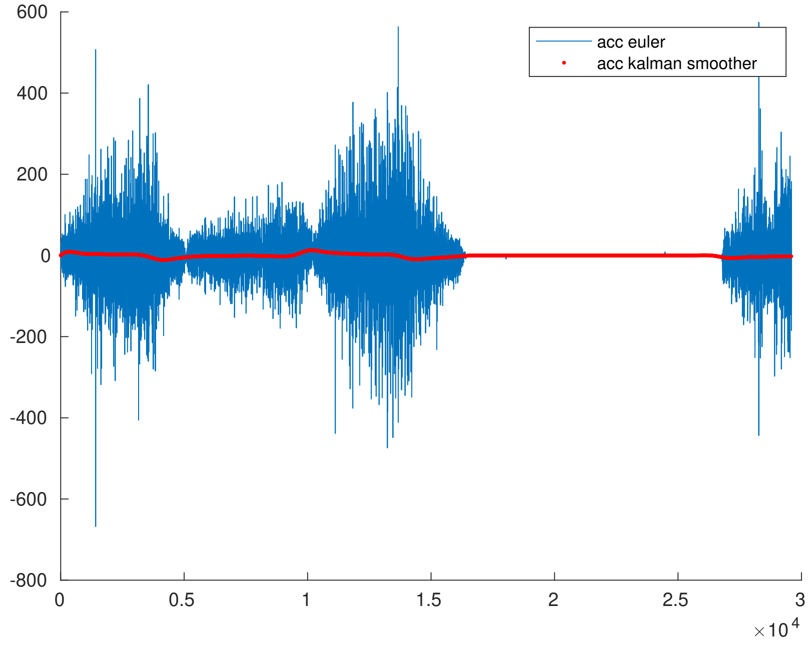

Figure 7: Accelerations estimations using kalman smoother