Physical Human Robot Interaction

Emanuele Feola, VR474837

March 2023

Contents

1 Homework 1 **3**
- 1.1 Assignment description . . . 3
- 1.2 Assumptions . . . 3
- 1.3 4 channel - schema . . . 4
- 1.4 4 channel - sinusoidal reference . . . 5
- 1.5 4 channel - sinusoidal reference, $D_m = 5$, $D_s = 10$ . . . 6
- 1.6 4 channel - step reference . . . 7

2 Homework 2 **8**
- 2.1 Assignment description . . . 8
- 2.2 Considerations . . . 8
- 2.3 Three-Channel bilateral teleoperation architecture . . . 9
- 2.4 Two-Channel position-position bilateral teleoperation architecture . . . 11
- 2.5 Two-Channel force-position bilateral teleoperation architecture . . . 13
- 2.6 Two-Channel force-force bilateral teleoperation architecture . . . 15

3 Homework 3-4 **16**
- 3.1 Assignment description . . . 16
- 3.2 Considerations . . . 16
  - 3.2.1 Continuous time system . . . 16
  - 3.2.2 Discrete time system . . . 16
- 3.3 Kalman filter and predictor . . . 17
- 3.4 Kalman filter and smoother . . . 18
- 3.5 Euler derivatives, Kalman filter, predictor, smoother . . . 19

4 Homework 5 **20**
- 4.1 Assignment description . . . 20
- 4.2 Considerations . . . 20
- 4.3 Model . . . 20
- 4.4 Goal . . . 20

5 Homework 6 **23**
- 5.1 Assignment description . . . 23
- 5.2 Considerations . . . 23
- 5.3 Scattering-based Position-Position . . . 24
- 5.4 Scattering-based Force-Position . . . 25

6 Homework 7 **26**
- 6.1 Assignment description . . . 26
- 6.2 Considerations . . . 26
- 6.3 Tank-based Position-Position . . . 27
- 6.4 Tank-based Force-Position . . . 31
  - 6.4.1 Sufficient initial energy . . . 31
  - 6.4.2 Not sufficient initial energy . . . 32

6.4.3 Velocity estimation from noisy position measurement 33

1 Homework 1

1.1 Assignment description

- Implement the Single-Input Single-Output Four-channel bilateral teleoperation architecture with

$$C_m = B_m + \frac{K_m}{s}$$

$$C_s = B_s + \frac{K_s}{s}$$

$$Z_m^{-1} = \frac{1}{M_m s}$$

$$Z_s^{-1} = \frac{1}{M_s s}$$

  where $M_m = 0.5$, $M_s = 2$.

- What happens if

$$Z_m^{-1} = \frac{1}{M_m s + D_m}$$

$$Z_s^{-1} = \frac{1}{M_s s + D_s}$$

  with $D_m = 5$ and $D_s = 10$?

- For the reference signal $x^d(t)$ in SISO system, it is possibile to use
  - a sinusoidal signal to test the dynamic response in free motion for different frequencies
  - a step response of a first-order low-pass filter with target value higher than the position of the environment x_e to test the interaction response

1.2 Assumptions

- the network delay is zero
- we know the exact dynamics of both master and slave manipulators
- position and force measurements are available for both master and slave manipulators

1.3 4 channel - schema

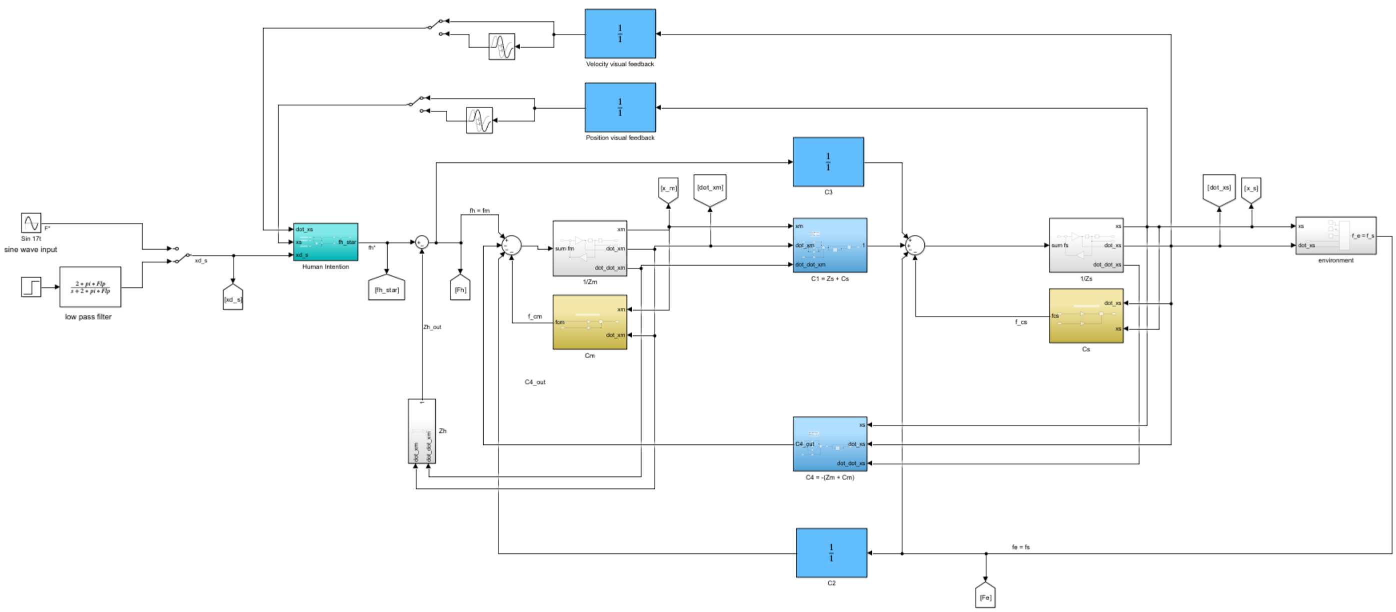

Figure 1: 4 channel architecture - simulink schema

1.4 4 channel - sinusoidal reference

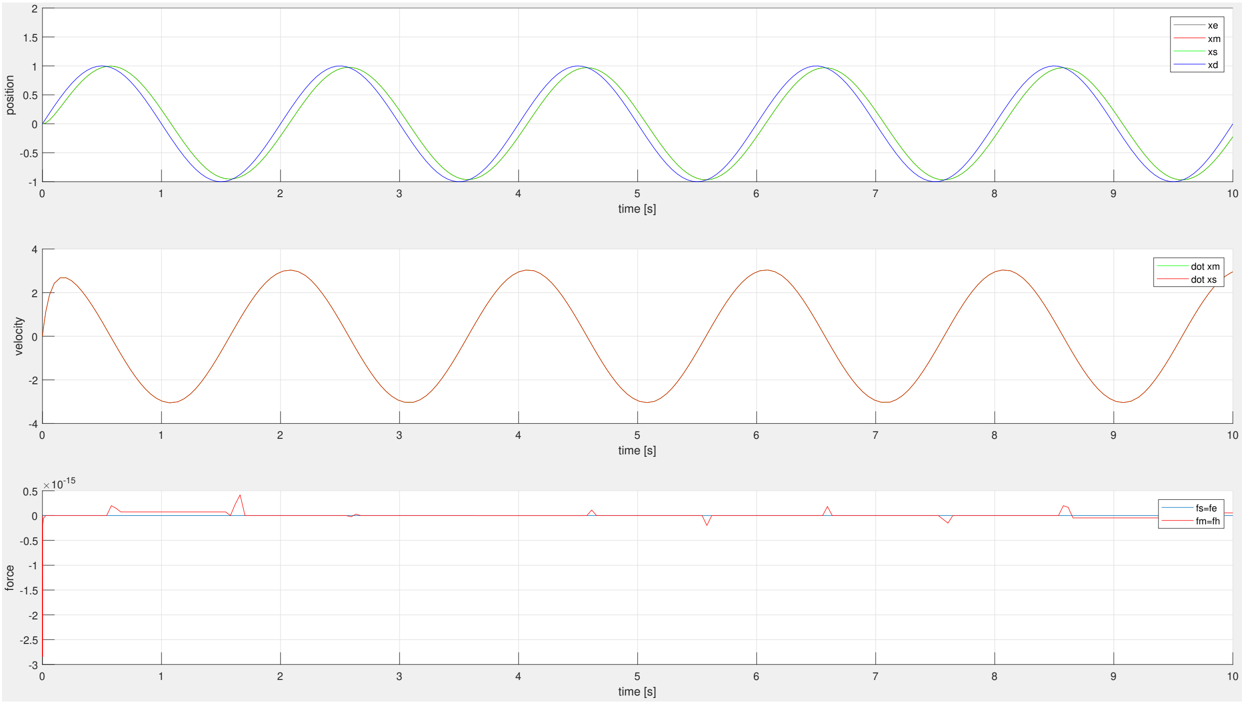

Figure 2: 4 channel architecture - free motion with sinusoidal reference

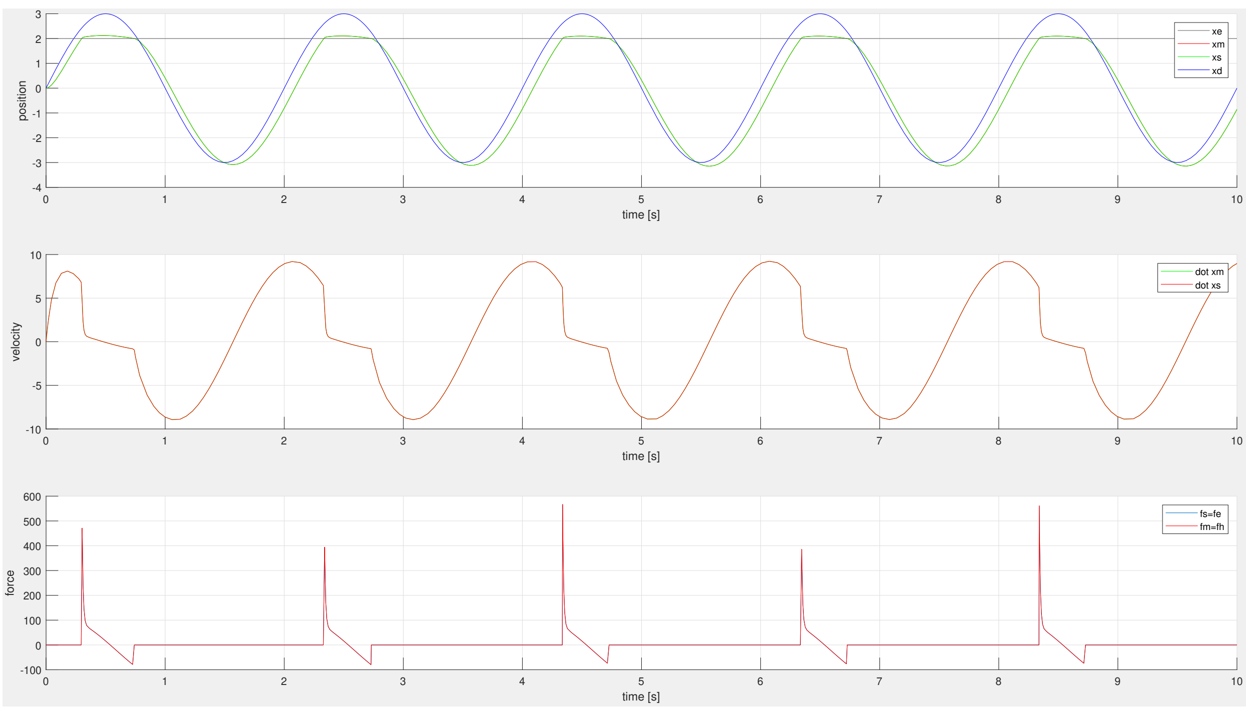

Figure 3: 4 channel architecture - contact with sinusoidal reference

1.5 4 channel - sinusoidal reference, $D_m = 5$, $D_s = 10$

We lose transparency, i.e. $x_m != x_e$, because we are not compensating in C1 and C4 controllers the correct manipulators dynamics. If we add D_m and D_s also in C1 and C4, then we get transparency again.

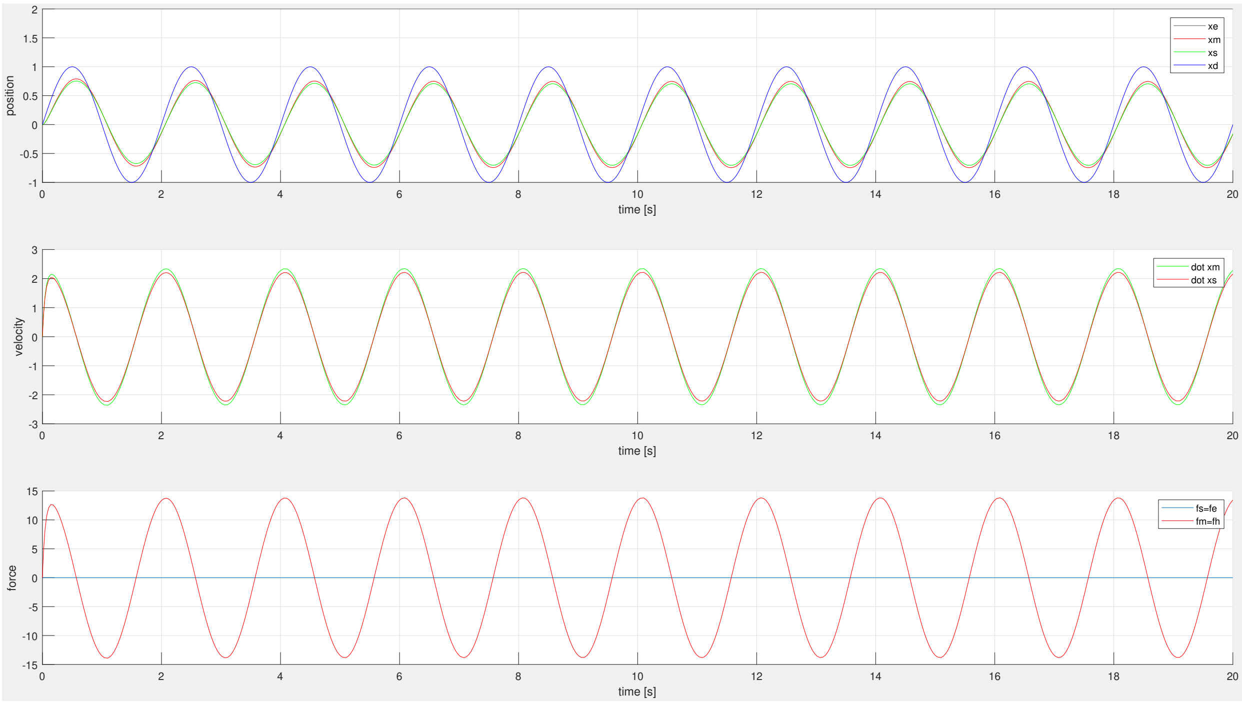

Figure 4: 4 channel architecture - free motion with sinusoidal reference, D_m and D_S not compensated

1.6 4 channel - step reference

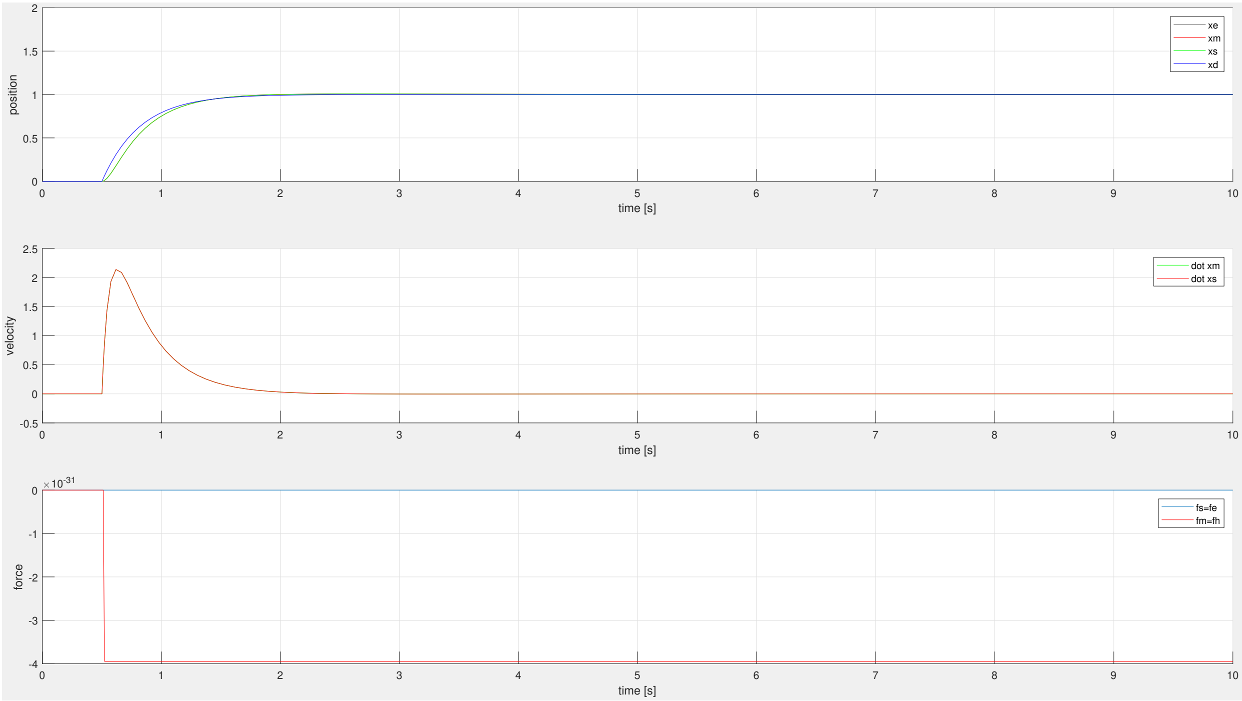

Figure 5: 4 channel architecture - free motion with step reference

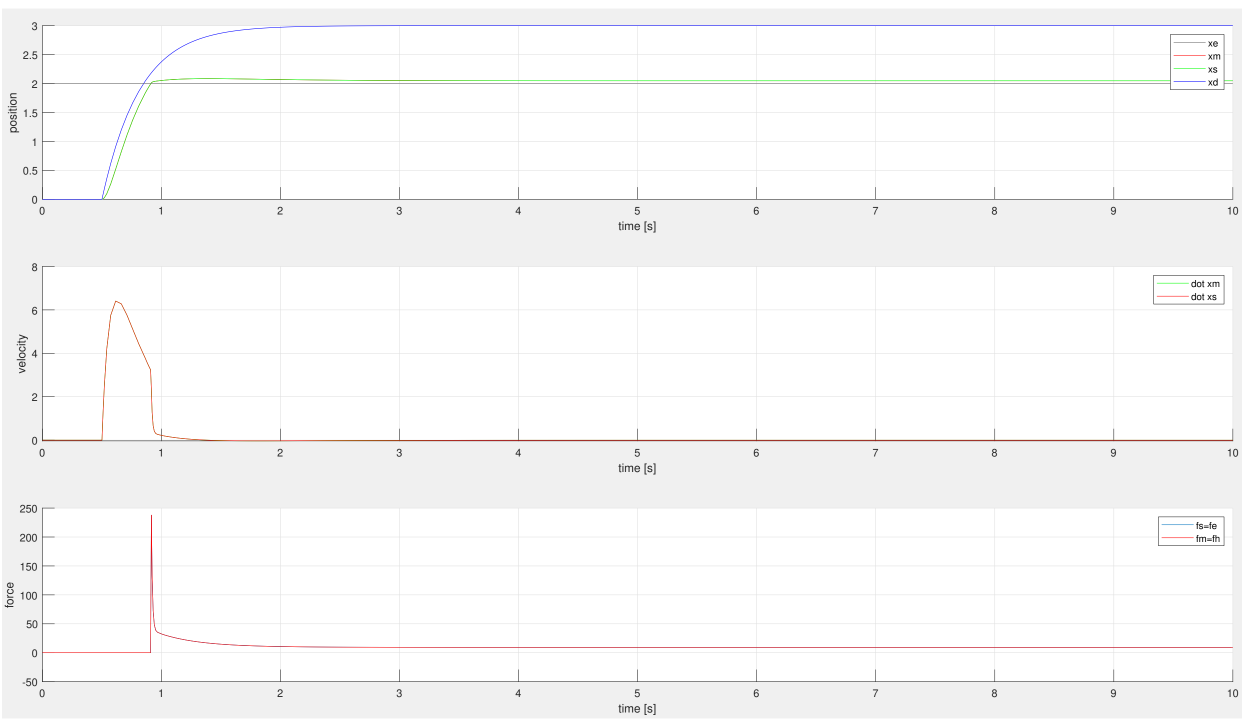

Figure 6: 4 channel architecture - contact with step reference

2 Homework 2

2.1 Assignment description

- Implement the three Two-Channel bilateral teleoperation architectures and the Three-Channel bilateral teleoperation architecture. Same parameters as in HW #1.
- What happens if transportation delays (see Simulink) are added in series at the controllers Ci (light blue blocks) in the different architectures?

2.2 Considerations

- By removing sensors, either force or position sensors, we lose perfect transparency, i.e. $x_m! = x_s$.
- By adding a delay we get closer to instability and system starts to have an oscillatory component like behaviour

Three channel architecture:

- we delete the force channel from master to slave
- we delete the force sensor on the haptic device
- we keep the force sensor on end-effector of the slave

Two channel pos-pos architecture:

- we delete the force channels from master to slave and from slave to master
- to implement this architecture we do not need any force sensors

Two channel force-pos architecture:

- we delete the force channel from master to slave
- we delete the velocity channel from slave to master

Two channel force-force architecture:

- we delete the velocity channel from master to slave
- we delete the velocity channel from slave to master

2.3 Three-Channel bilateral teleoperation architecture

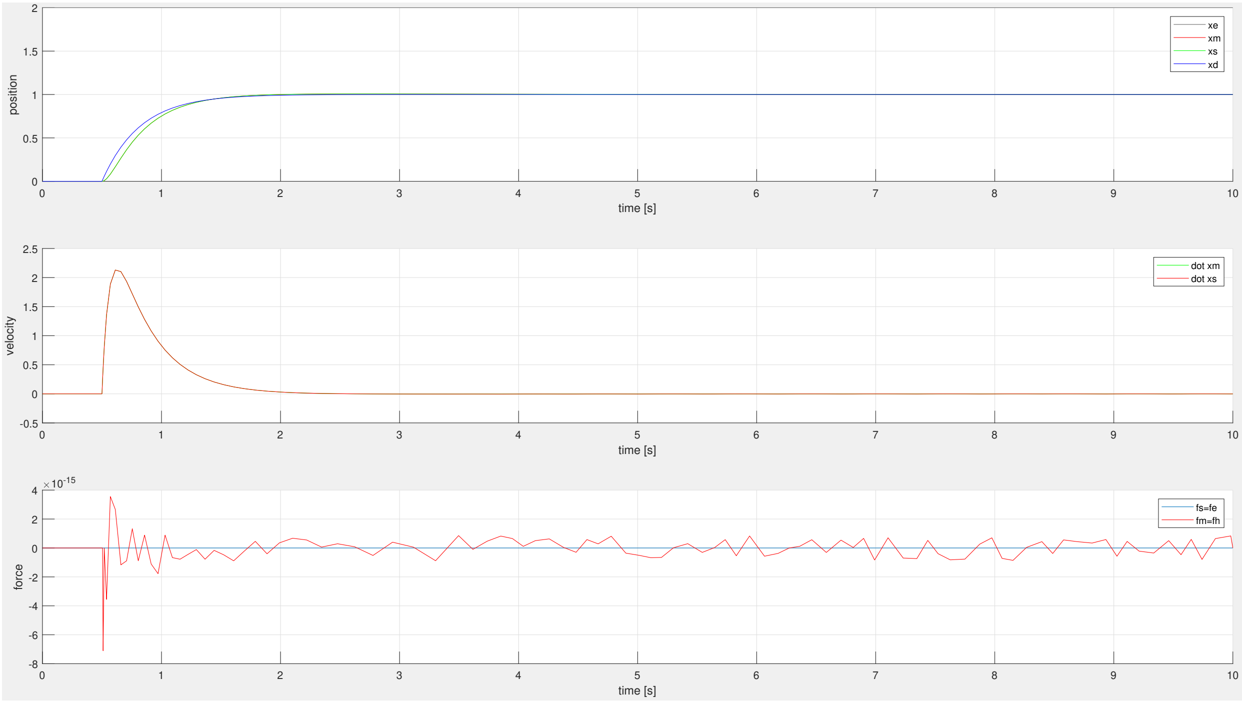

Figure 7: Three-Channel bilateral teleoperation architecture - free motion with step reference

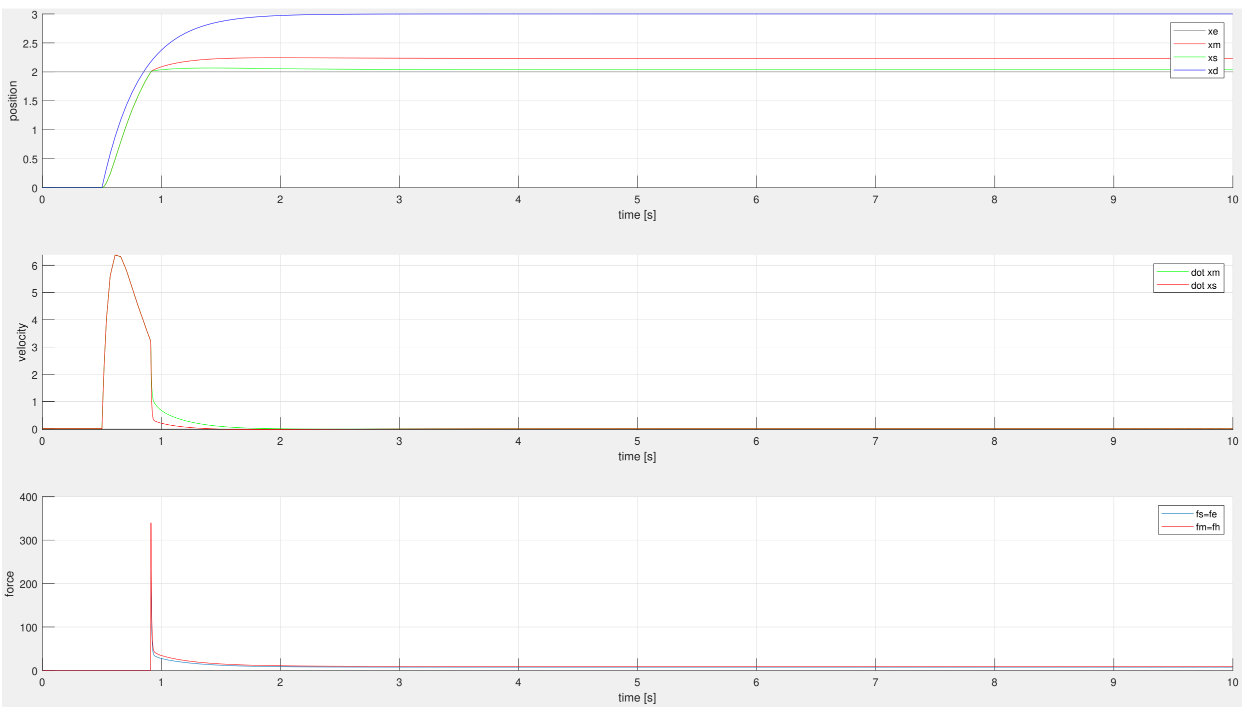

Figure 8: Three-Channel bilateral teleoperation architecture - contact with step reference

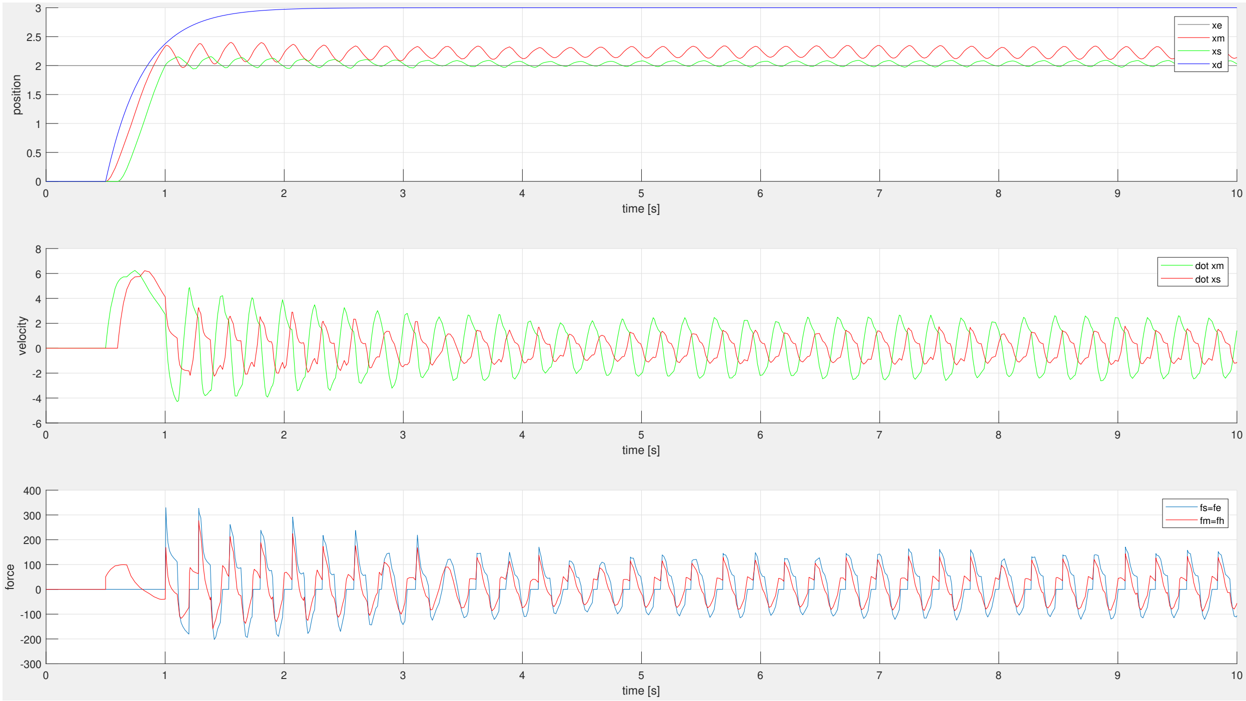

Figure 9: Three-Channel bilateral teleoperation architecture - contact with step reference and 0.1s delay

2.4 Two-Channel position-position bilateral teleoperation architecture

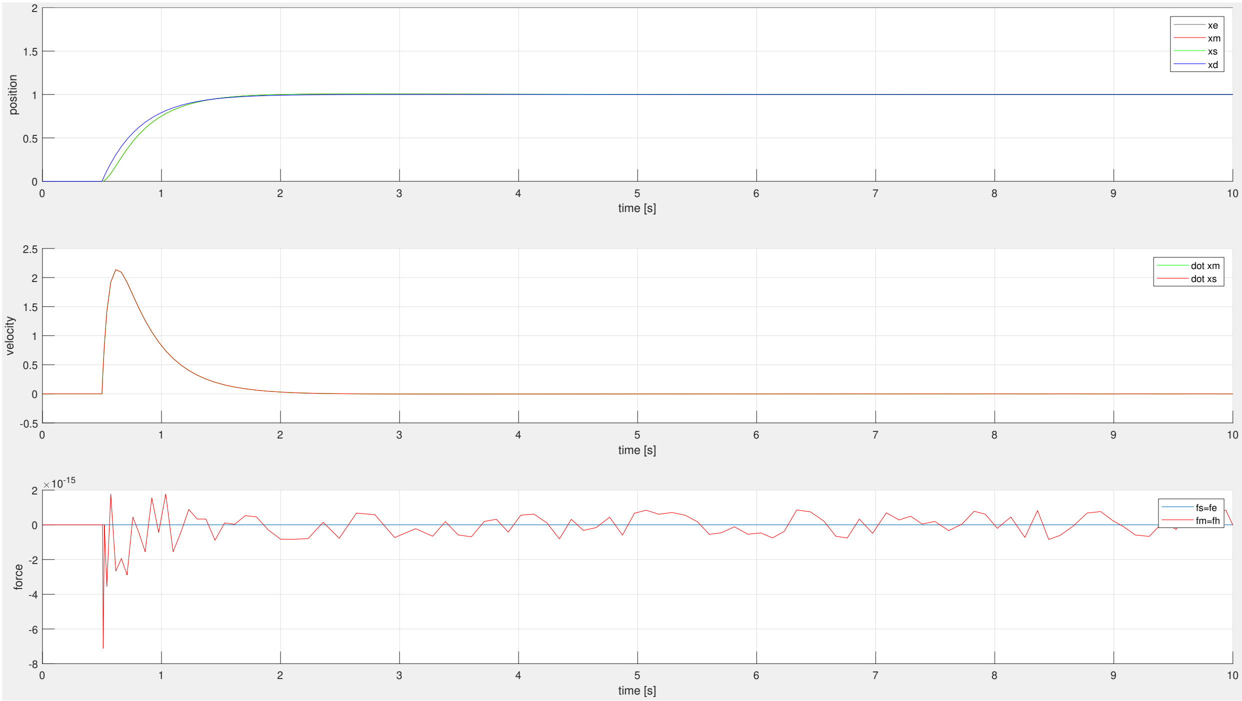

Figure 10: Two-Channel position-position bilateral teleoperation architecture - free motion with step reference

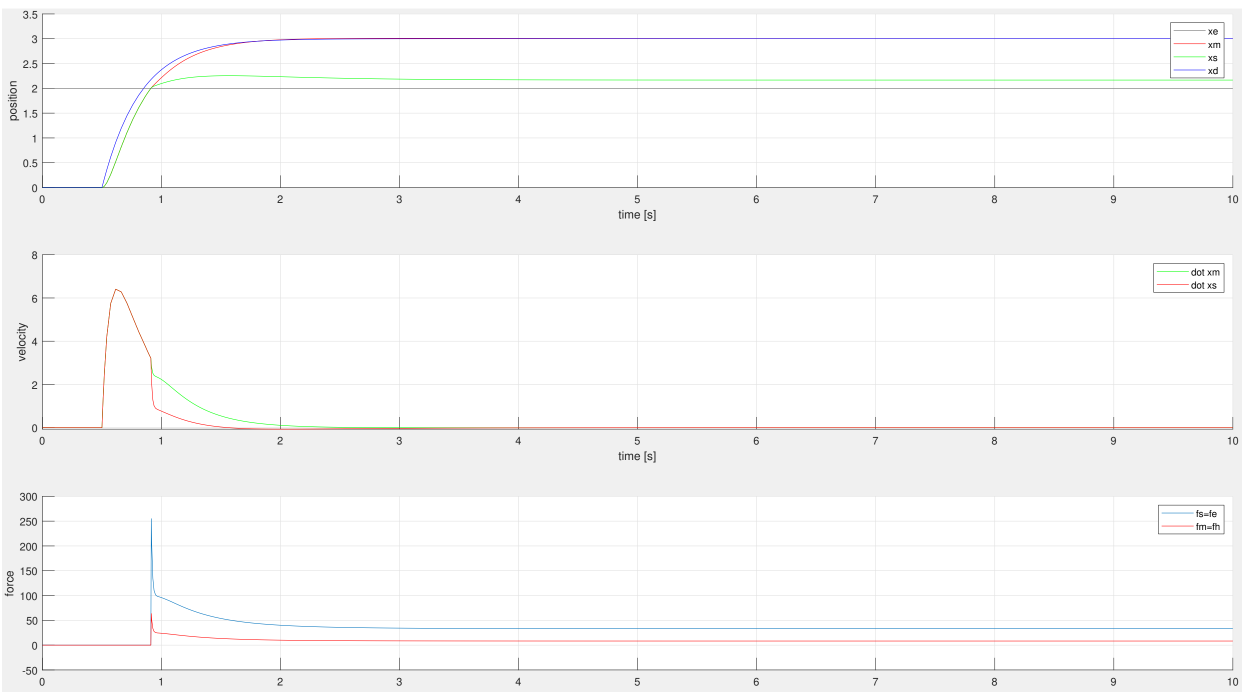

Figure 11: Two-Channel position-position bilateral teleoperation architecture - contact with step reference

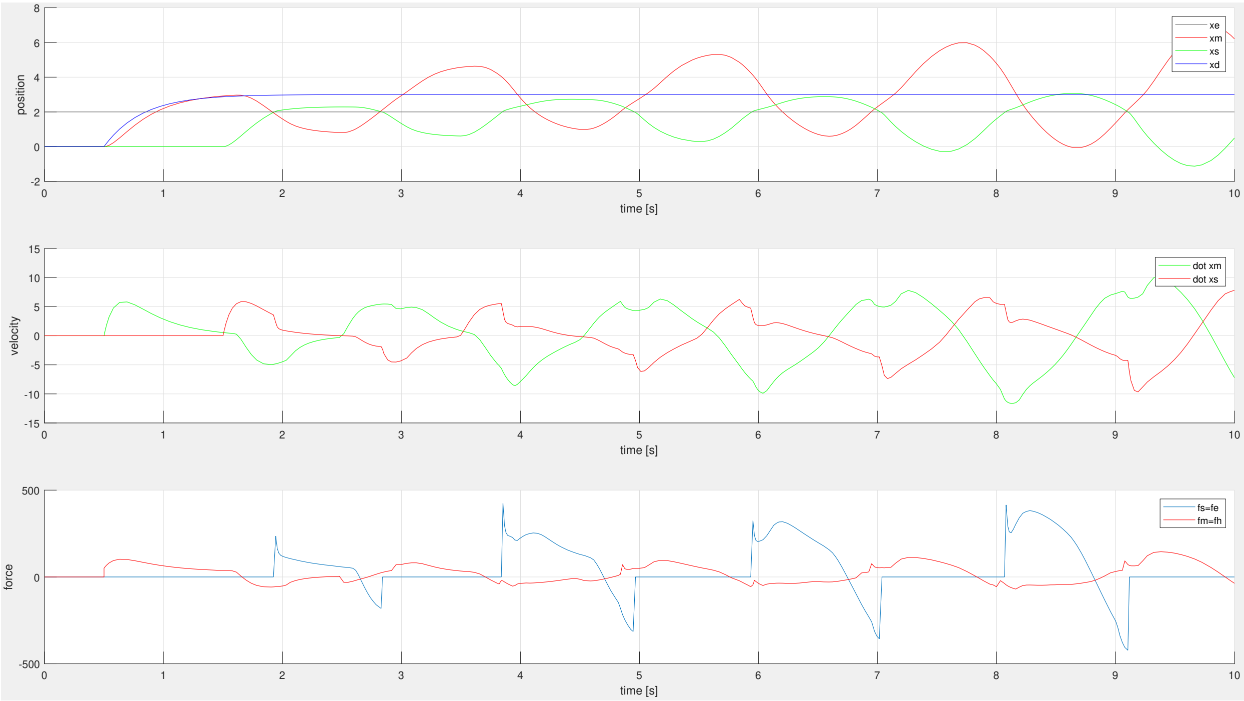

Figure 12: Two-Channel position-position bilateral teleoperation architecture - contact with step reference and $0.1s$ delay

2.5 Two-Channel force-position bilateral teleoperation architecture

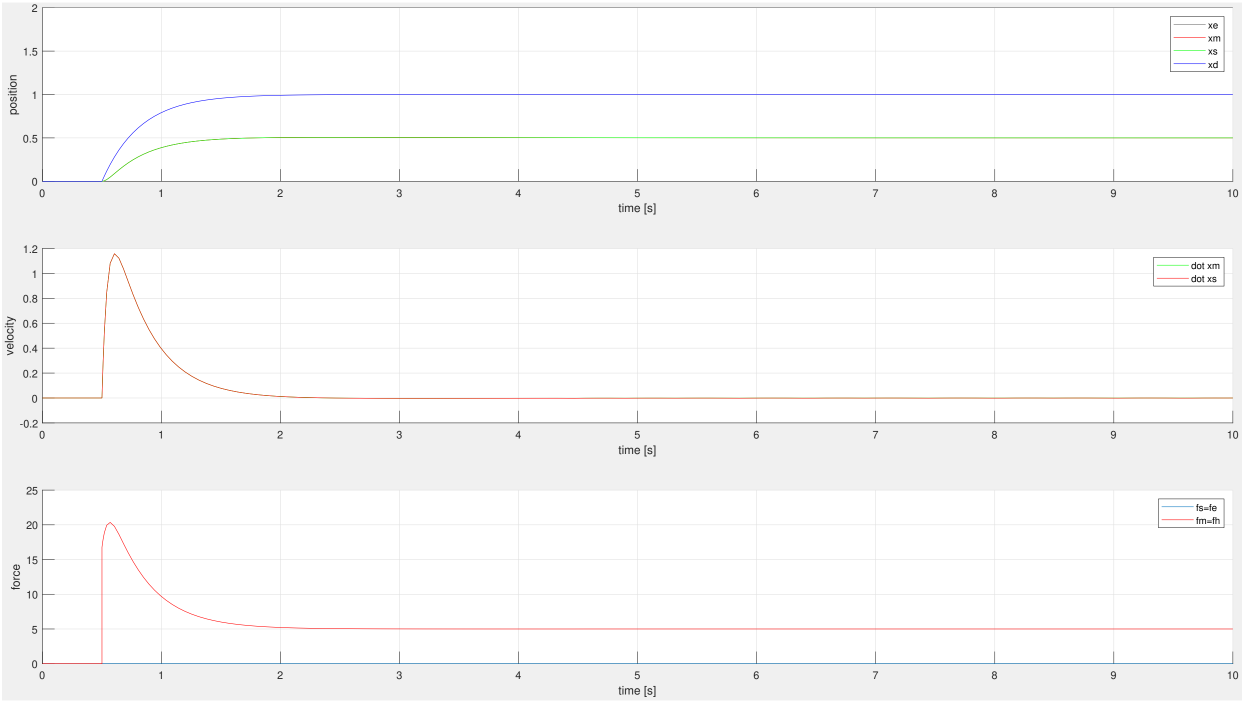

Figure 13: Two-Channel force-position bilateral teleoperation architecture - free motion with step reference

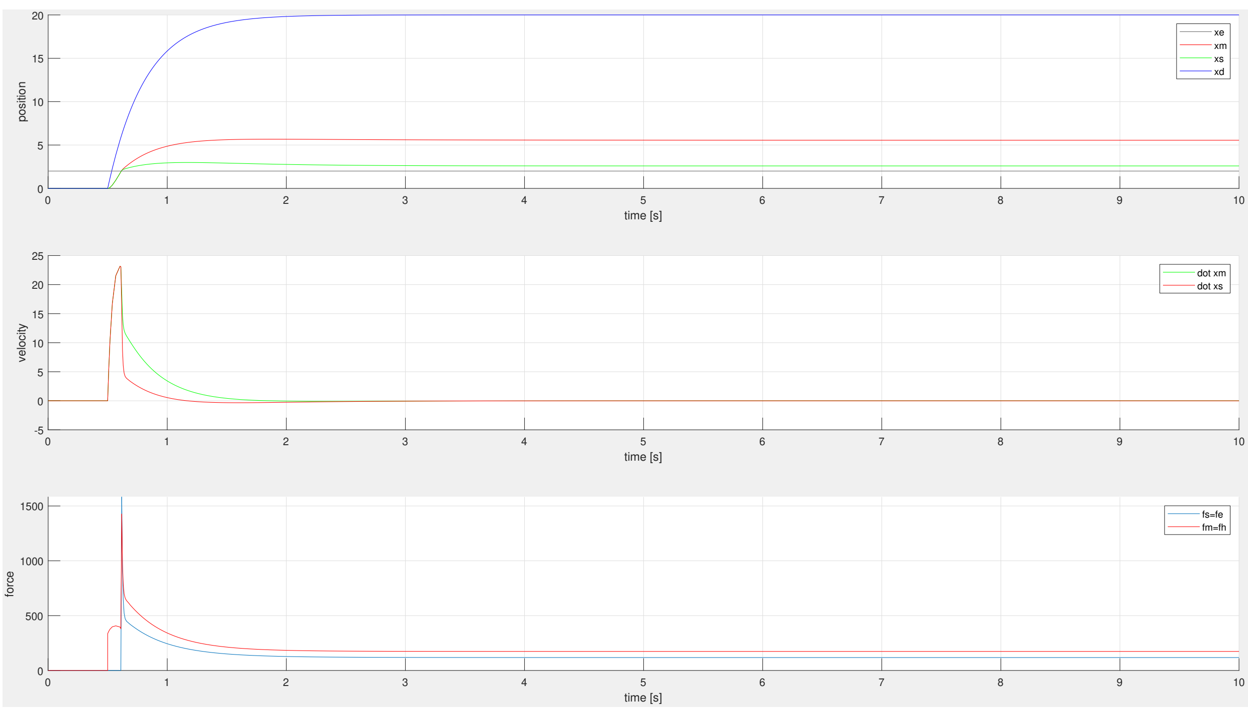

Figure 14: Two-Channel force-position bilateral teleoperation architecture - contact with step reference

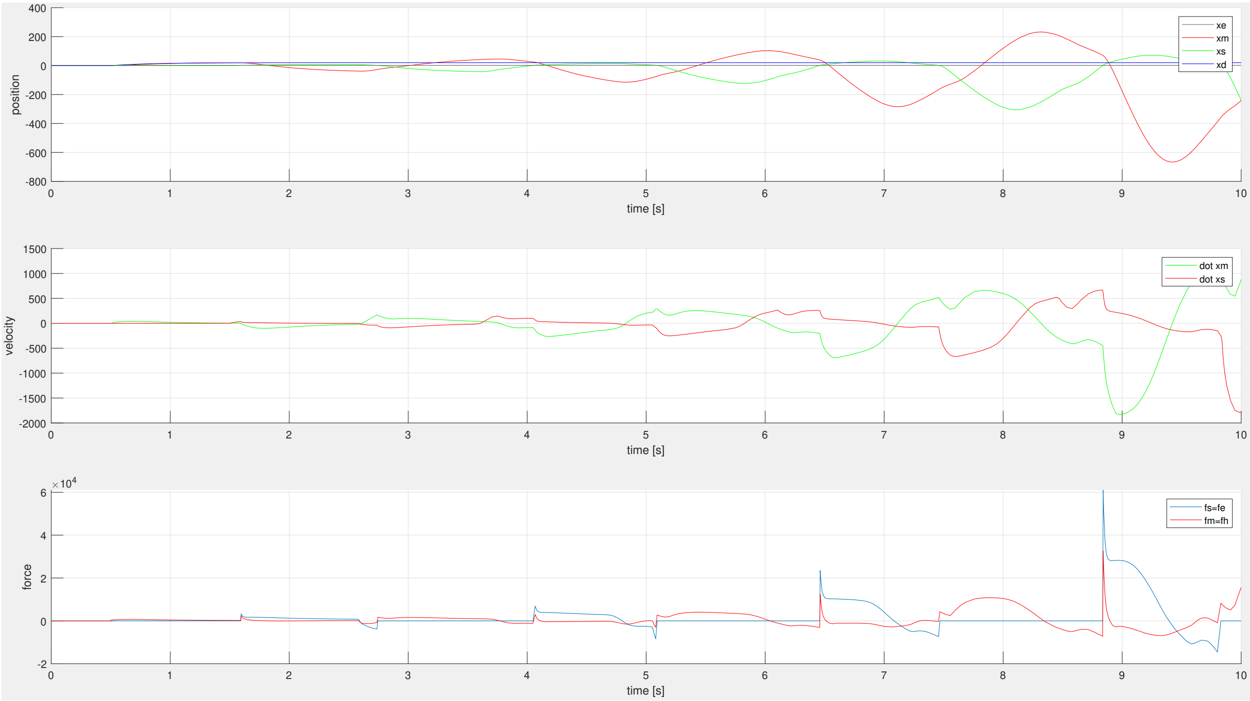

Figure 15: Two-Channel force-position bilateral teleoperation architecture - contact with step reference and $0.1s$ delay

2.6 Two-Channel force-force bilateral teleoperation architecture

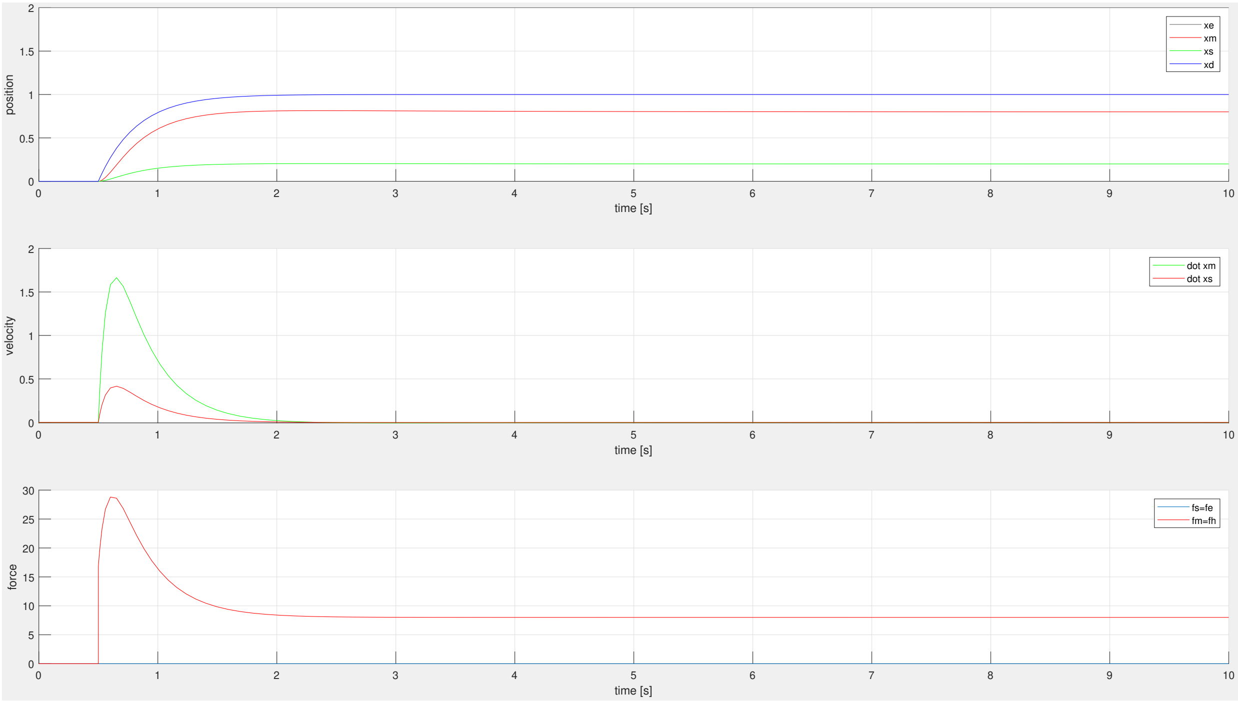

Figure 16: Two-Channel force-force bilateral teleoperation architecture - free motion with step reference

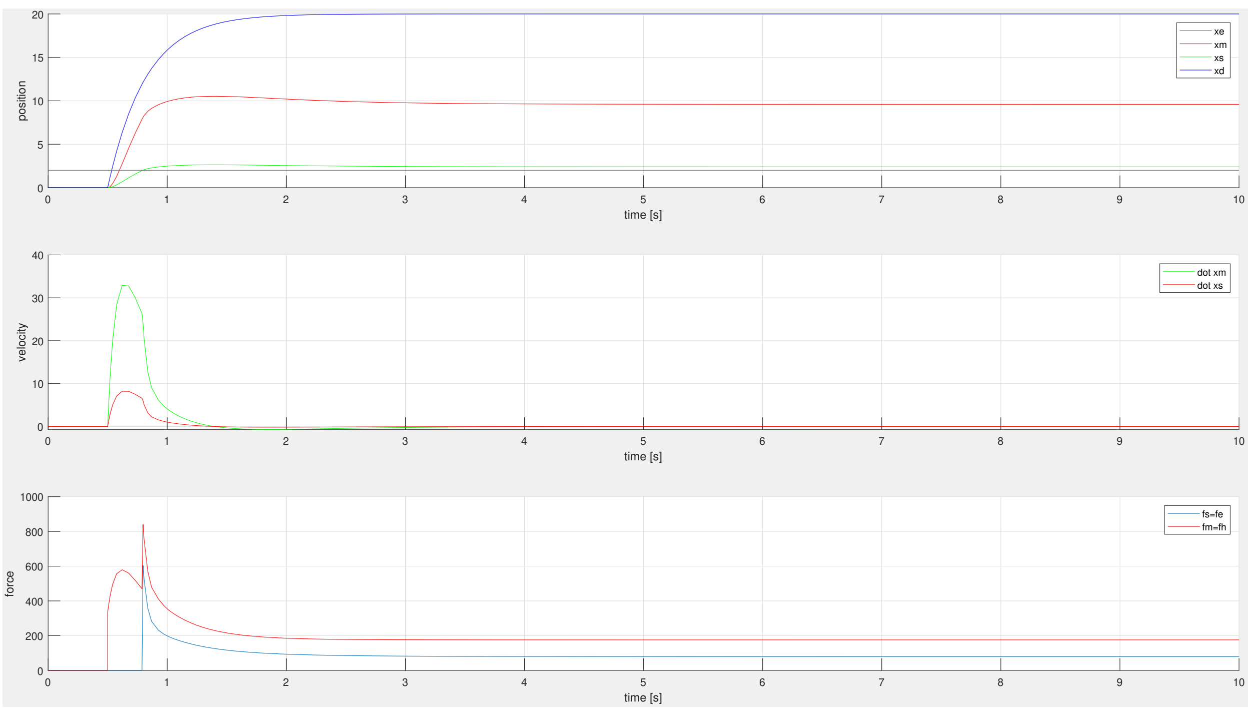

Figure 17: Two-Channel force-force bilateral teleoperation architecture - contact with step reference

3 Homework 3-4

3.1 Assignment description

- Implement the Kalman filter/predictor and estimate the velocity and acceleration from noisy position measurements (see .mat file)
- Implement the steady-state Kalman filter/predictor and estimate the velocity and acceleration from noisy position measurements (see .mat file)
- Implement the Kalman smoother and estimate the velocity and acceleration from noisy position measurements (see .mat file)

3.2 Considerations

3.2.1 Continuous time system

$$x = \begin{bmatrix} \theta \\ \omega \\ \dot{\omega} \end{bmatrix}$$

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

The derivative of the angular acceleration $\ddot{\omega}$, i.e. the third element of $\dot{x}(t)$, is described by white noise $w(t)$ defined as a gaussian random variable $\mathcal{N}(\mu = 0, \sigma^2 = Q)$.
The measurement equation $y(t)$ is the 'real' angular displacement θ + white noise $v(t)$, defined as $\mathcal{N}(\mu = 0, \sigma^2 = R)$

3.2.2 Discrete time system

The discrete version of the continuous time model must be computed, since we introduce $t = KT_s$

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

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

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

3.3 Kalman filter and predictor

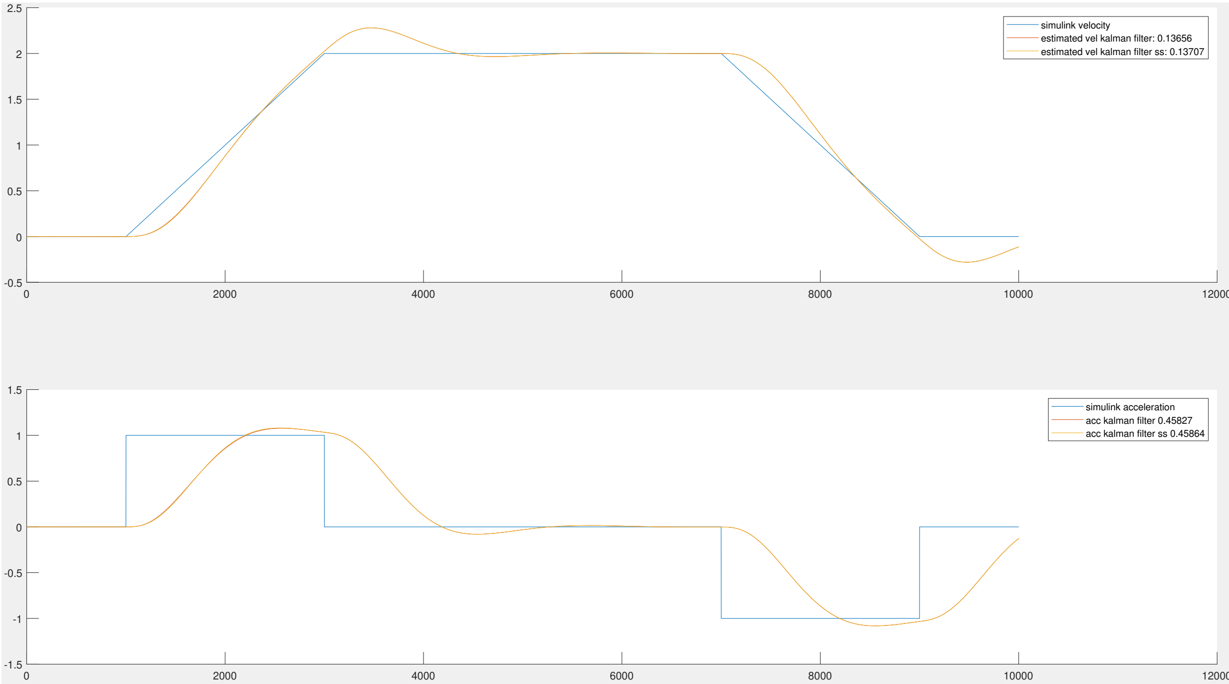

Figure 18: Velocity and acceleration estimation from noisy position data - filter, filter ss

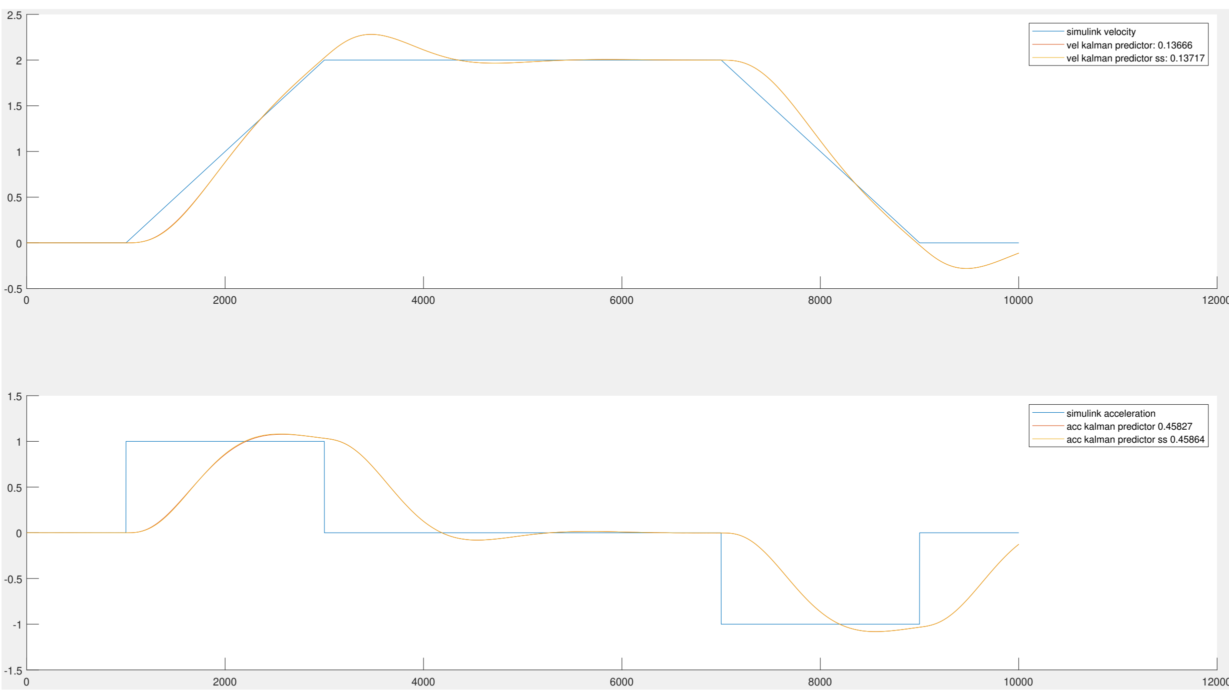

Figure 19: Velocity and acceleration estimation from noisy position data - predictor, predictor ss

3.4 Kalman filter and smoother

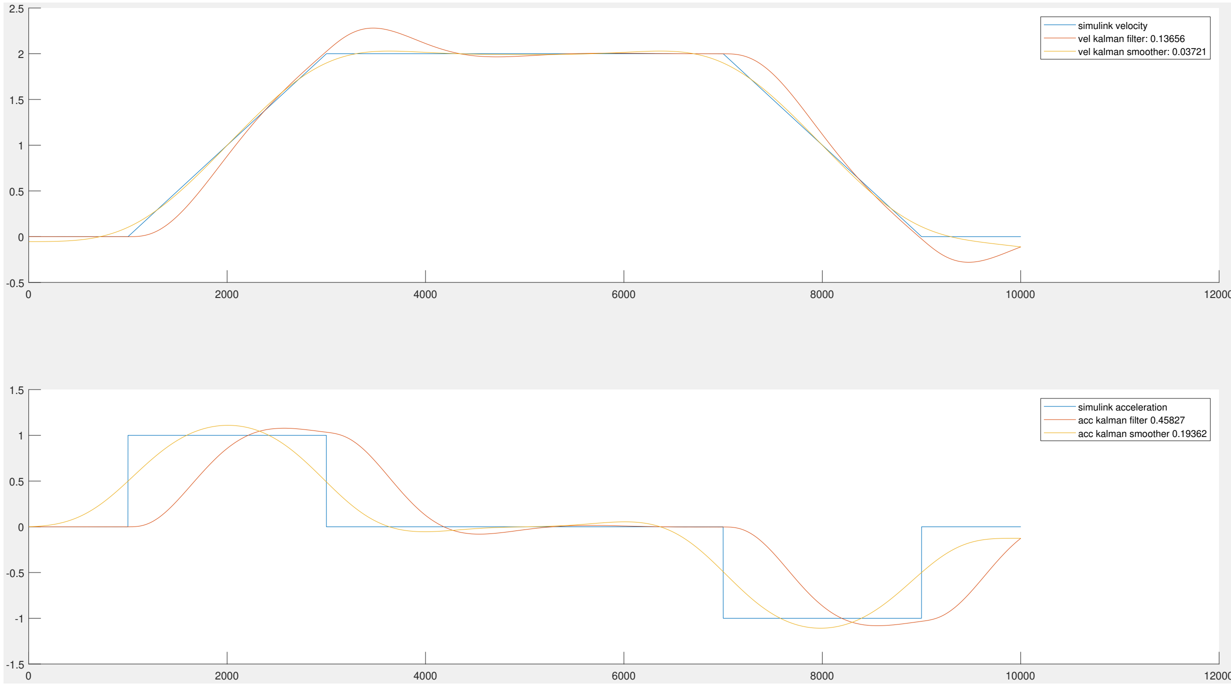

Figure 20: Velocity and acceleration estimation from noisy position data - filter, smoother

3.5 Euler derivatives, Kalman filter, predictor, smoother

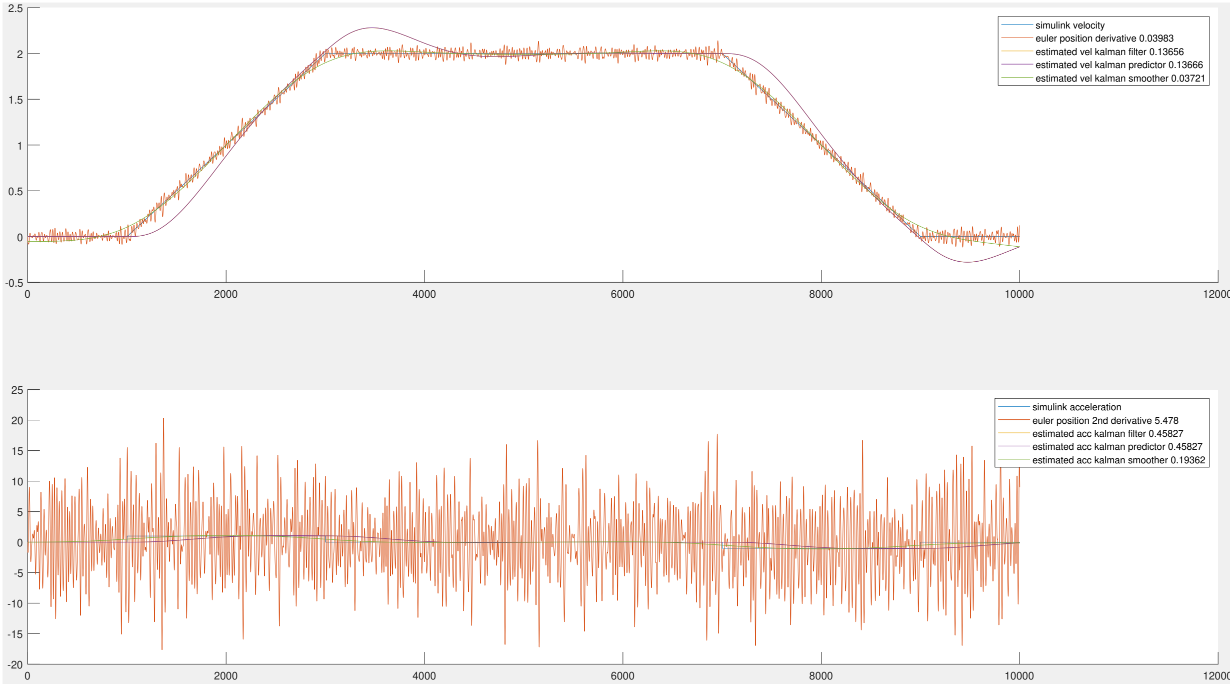

Figure 21: Velocity and acceleration estimation from noisy position data - euler derivative, filter, predictor, smoother

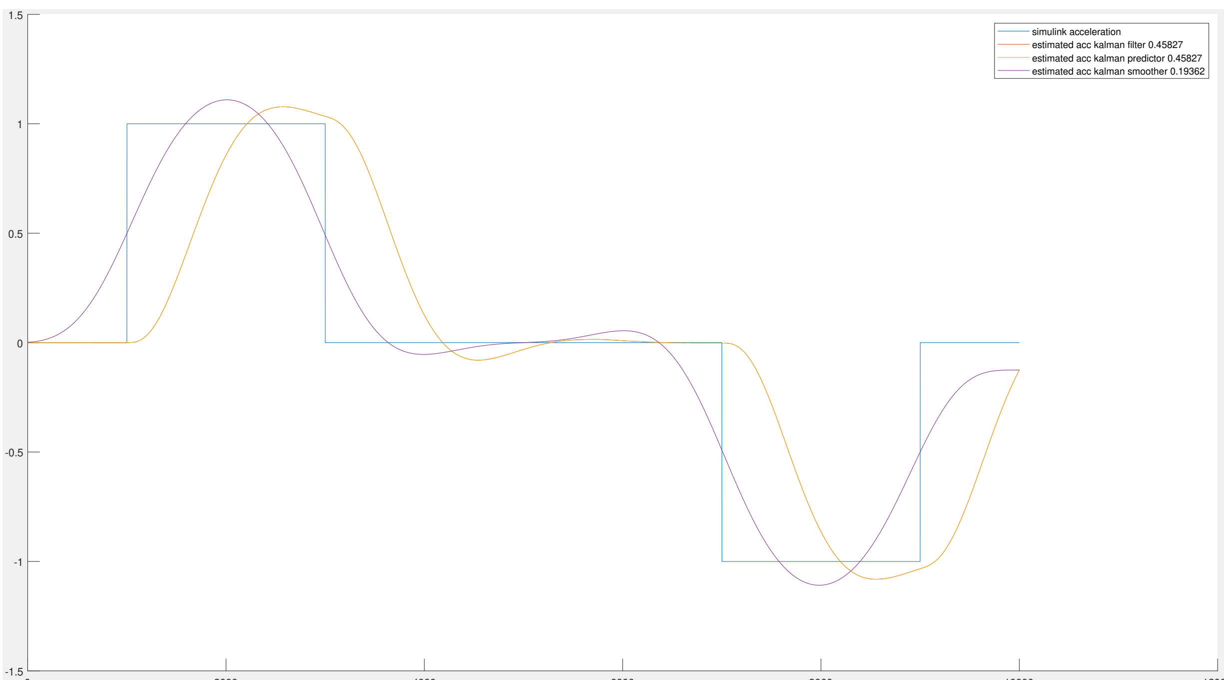

Figure 22: Acceleration estimation from noisy position data - filter, predictor, smoother

4 Homework 5

4.1 Assignment description

- Identify the parameters k and τ (i.e. J and D) using the LS and the RLS on the DC motors data.

4.2 Considerations

4.3 Model

Starting from the differential equation of a DC motor, we get the following relationship:

$$x = \begin{bmatrix} \dot{\omega} \\ \omega \end{bmatrix} \qquad y = V \qquad \theta = \begin{bmatrix} \tau/k \\ 1/k \end{bmatrix}$$

$$y = x\theta$$

and given N samples of x and y:

$$Y = X\theta$$

The prediction of y given a new value of x is:

$$\hat{y} = x\hat{\theta}$$

where $\hat{\theta} = (X^T X)^{-1} X^T Y$.
NB: θ is the vector of unknowns we want to estimate

4.4 Goal

Starting from noisy position measurements (exported from Simulink) velocity ω and acceleration $\dot{\omega}$ are computed using the Kalman Filter. Voltage measurements are also exported from Simulink. Both LS and RLS are used to compute τ and k.

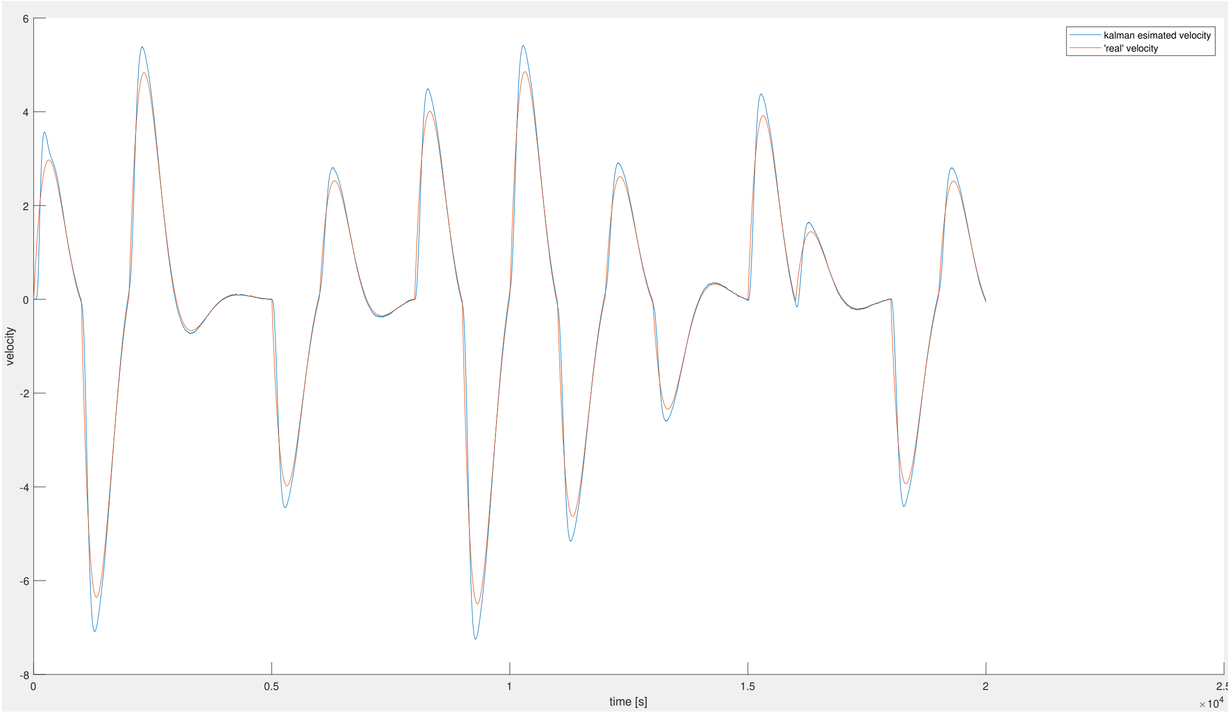

Figure 23: Velocity estimation from noisy position data - kalman filter vs real velocity

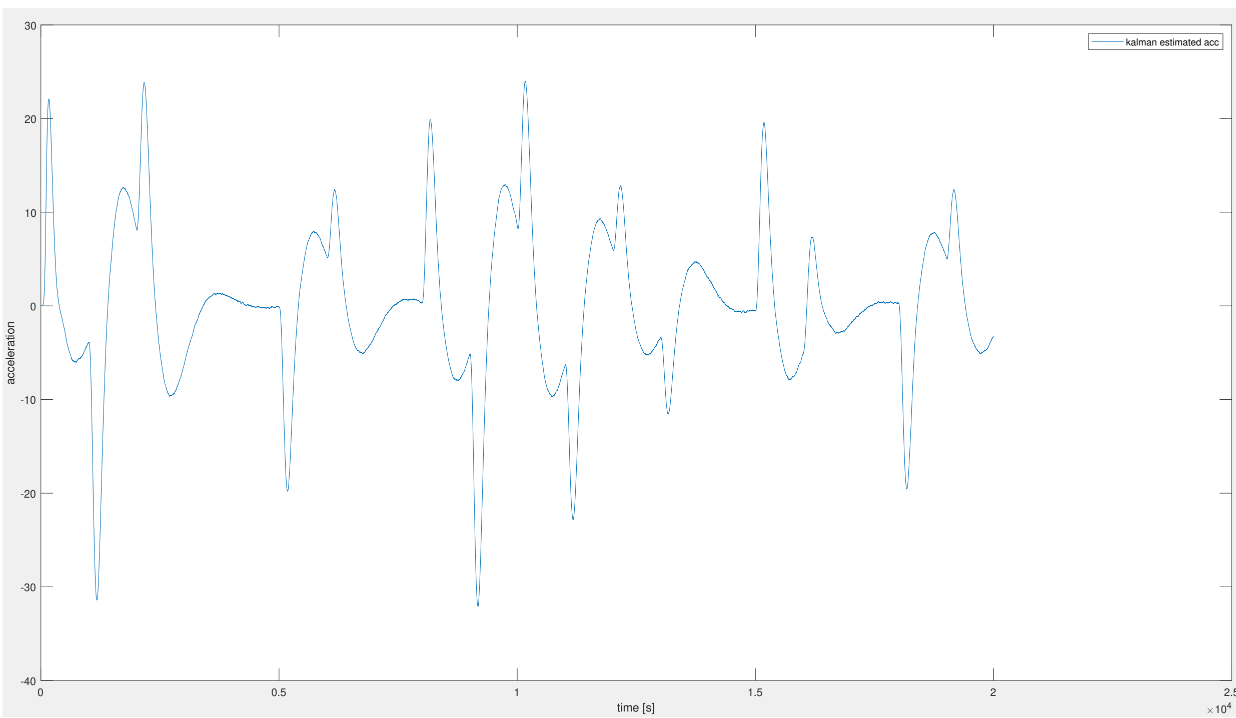

Figure 24: Acceleration estimation from noisy position data - kalman filter

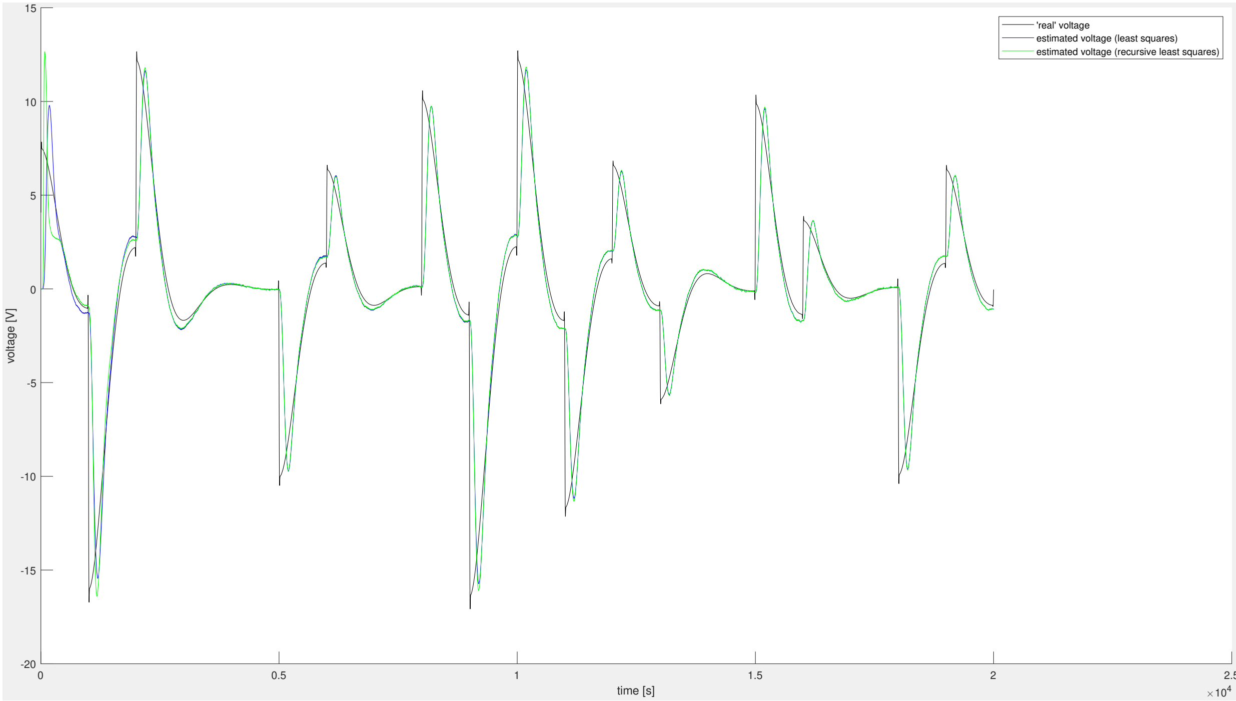

Figure 25: Voltage estimation using estimated parameters - Least Squares estimated voltage vs Recursive LS estimated voltage vs real voltage

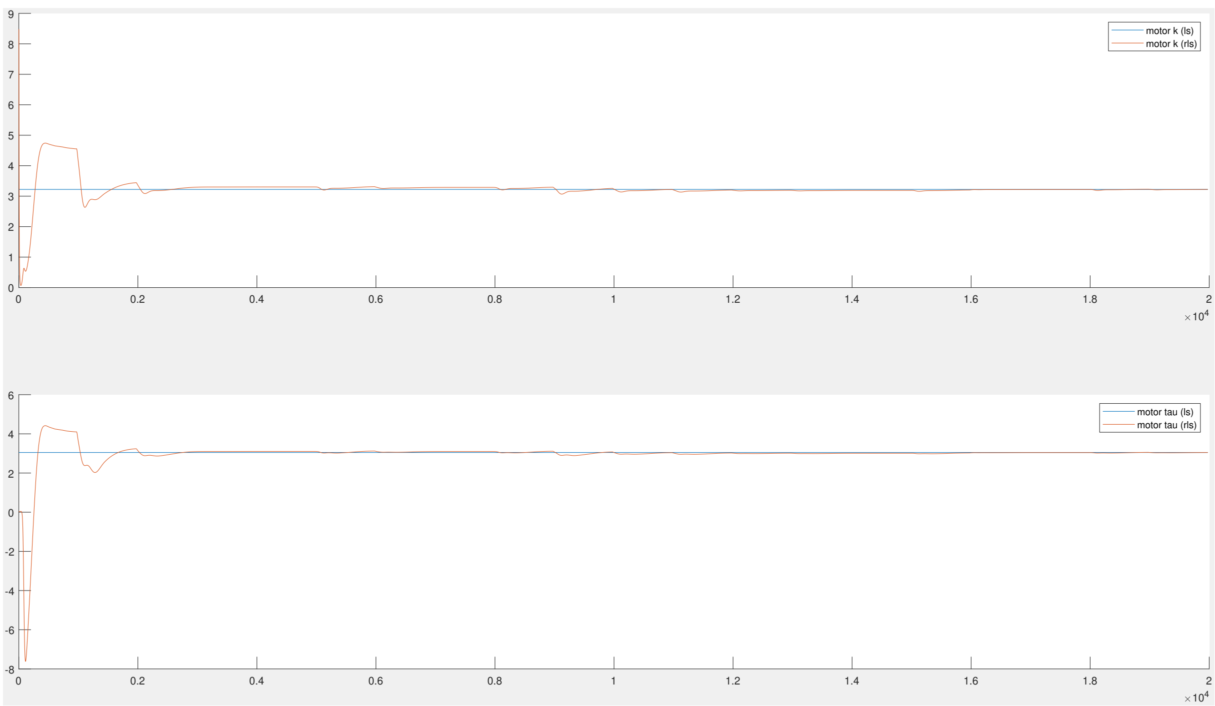

Figure 26: Motor parameter estimation - LS vs RLS

5 Homework 6

5.1 Assignment description

- Implement the Scattering-based bilateral teleoperation architecture for the
  - Force-Position case
  - Position-Position case
- Compare positions, velocities, forces, commands in free motion and in contact
- Create another simulink model and (a) add the measurement noise to the position/force signals, and (b) estimate velocities from positions

5.2 Considerations

- The scattering-based bilateral teleoperation architecture was evaluated with a delay of 100 time steps in the communication channel. Since the sampling time $T_s = 0.001s$, the total delay is about $0.1s$.
- A low pass filter with cutoff frequency of $100Hz$ was introduced in the communication channel to correct numerical errors.
- The environment is modeled as a spring-damper system with $K_e = 200$ and $Be = 100$.
- In the following images, a 'ripple' appears in the velocity plots. This is due to the delay, if the architecture is evaluated with a shorter delay, e.g. 10 time steps, then the ripple disappears.

5.3 Scattering-based Position-Position

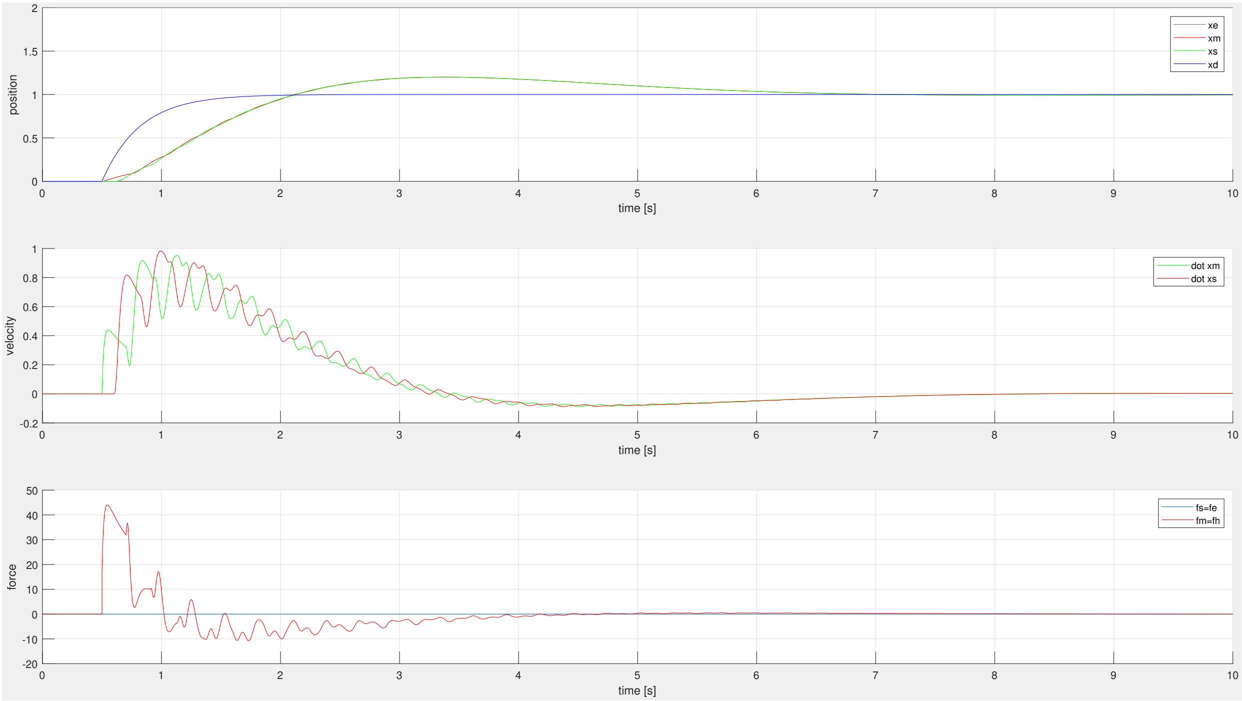

Figure 27: Scattering-based Position-Position: free motion

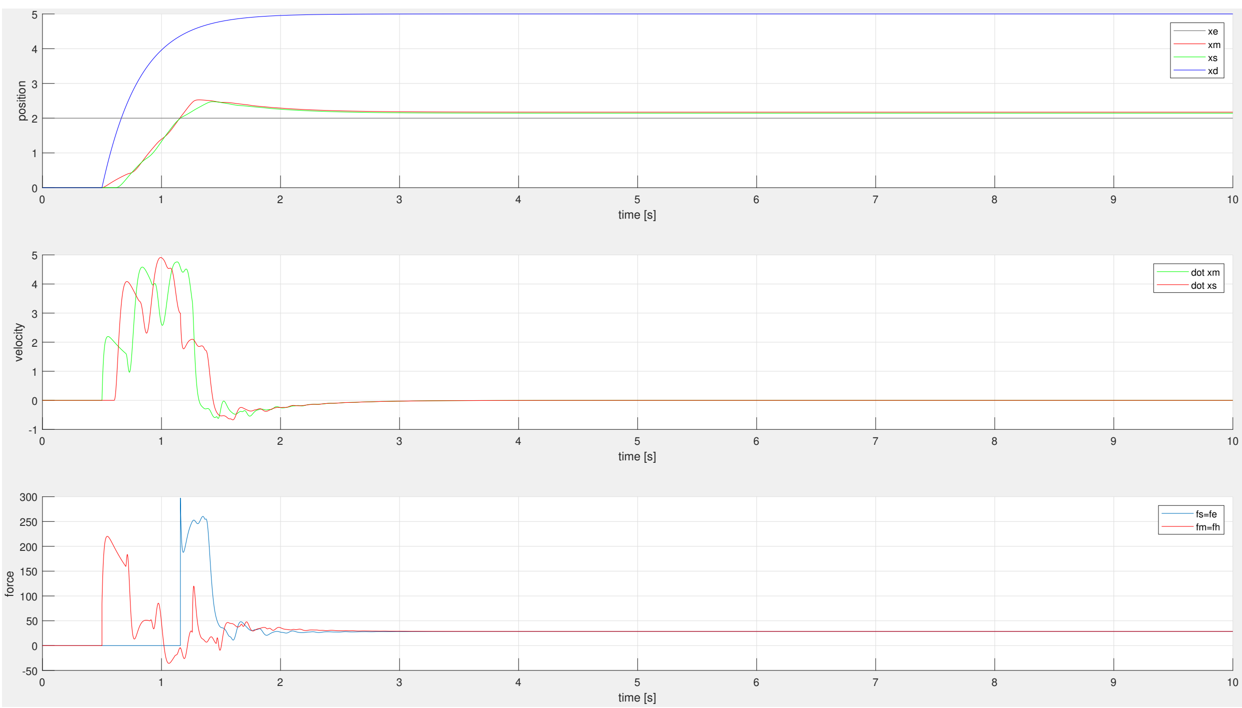

Figure 28: Scattering-based Position-Position: contact with environment

5.4 Scattering-based Force-Position

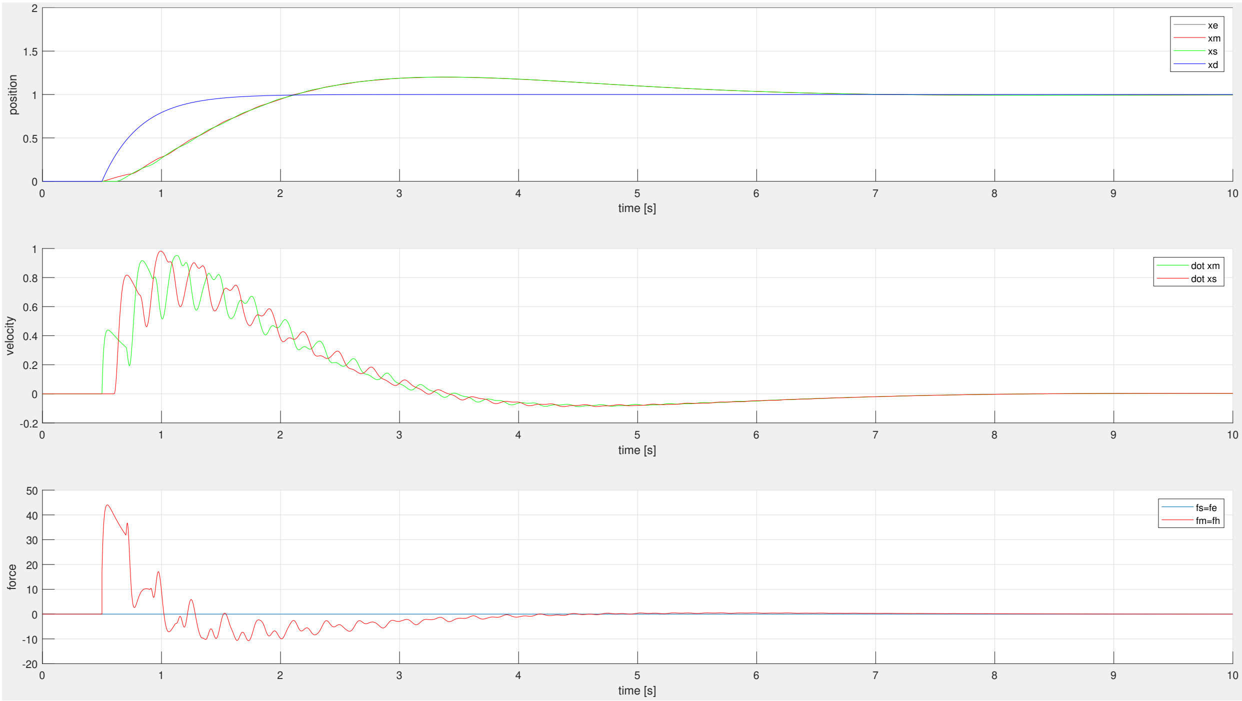

Figure 29: Scattering-based Force-Position: free motion

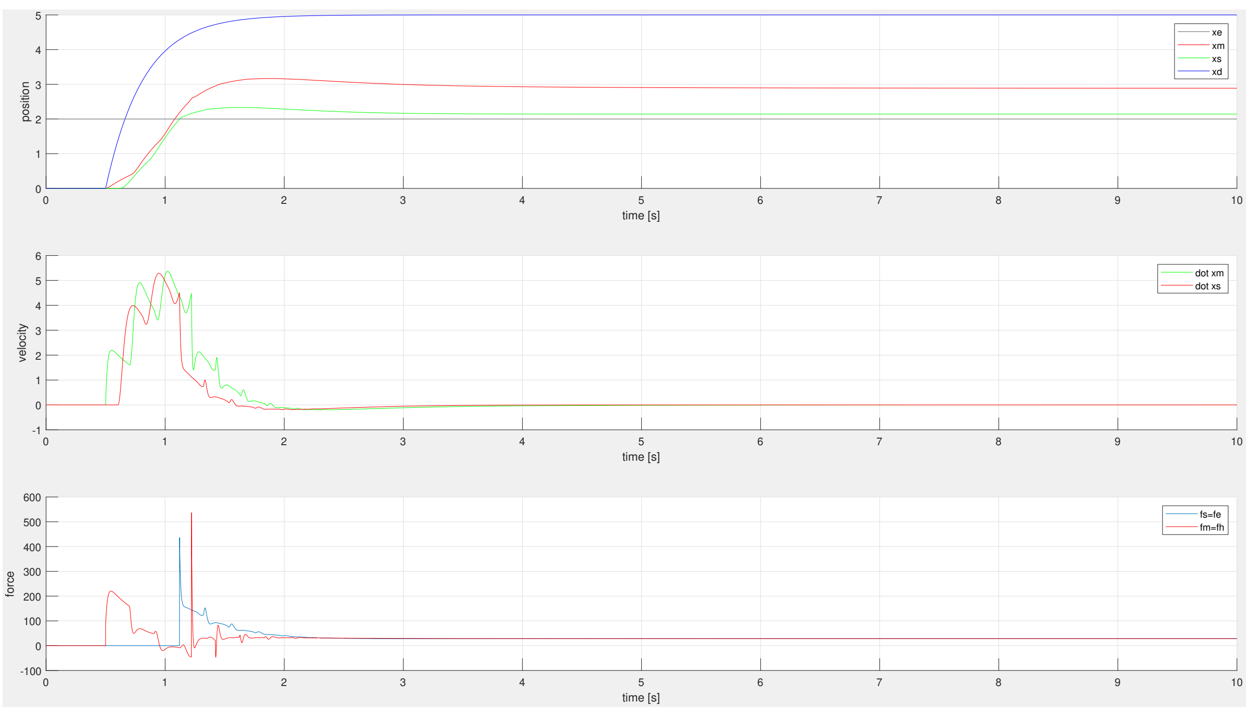

Figure 30: Scattering-based Force-Position: contact with environment

6 Homework 7

6.1 Assignment description

- Implement the Tank-based bilateral teleoperation architecture for the
  - Force-Position case
  - Position-Position case
- Compare positions, velocities, forces, commands in free motion and in contact
- Create another simulink model and (a) add the measurement noise to the position/force signals, and (b) estimate velocities from positions (if needed)

6.2 Considerations

- The tank-based bilateral teleoperation architecture was evaluated with a delay of 100 time steps in the communication channel. Since the sampling time $T_s = 0.001s$, the total delay is about $0.1s$.
- The environment is modeled as a spring-damper system with $K_e = 200$ and $Be = 100$.

6.3 Tank-based Position-Position

$\beta = 0.01$, $\alpha = 0.2$, the initial tank energy is enough to perform the desired trajectory.

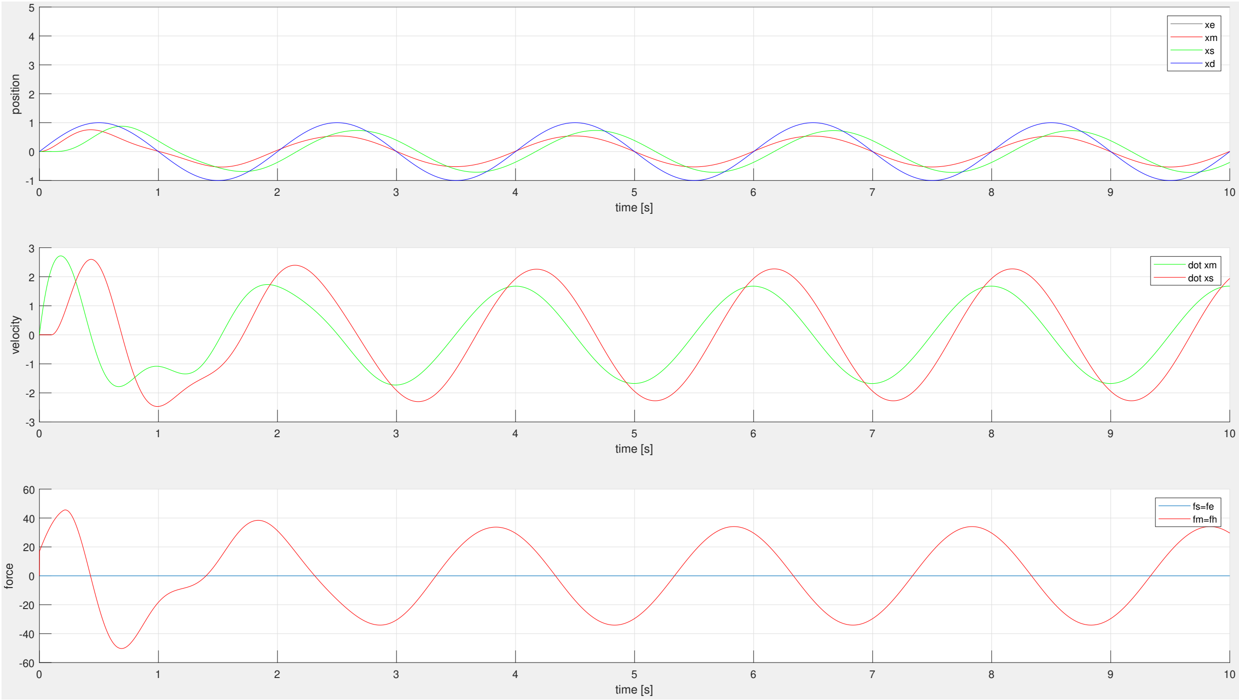

Figure 31: Tank-based Position-Position: free motion, $\beta = 0.01$, $\alpha = 0.2$

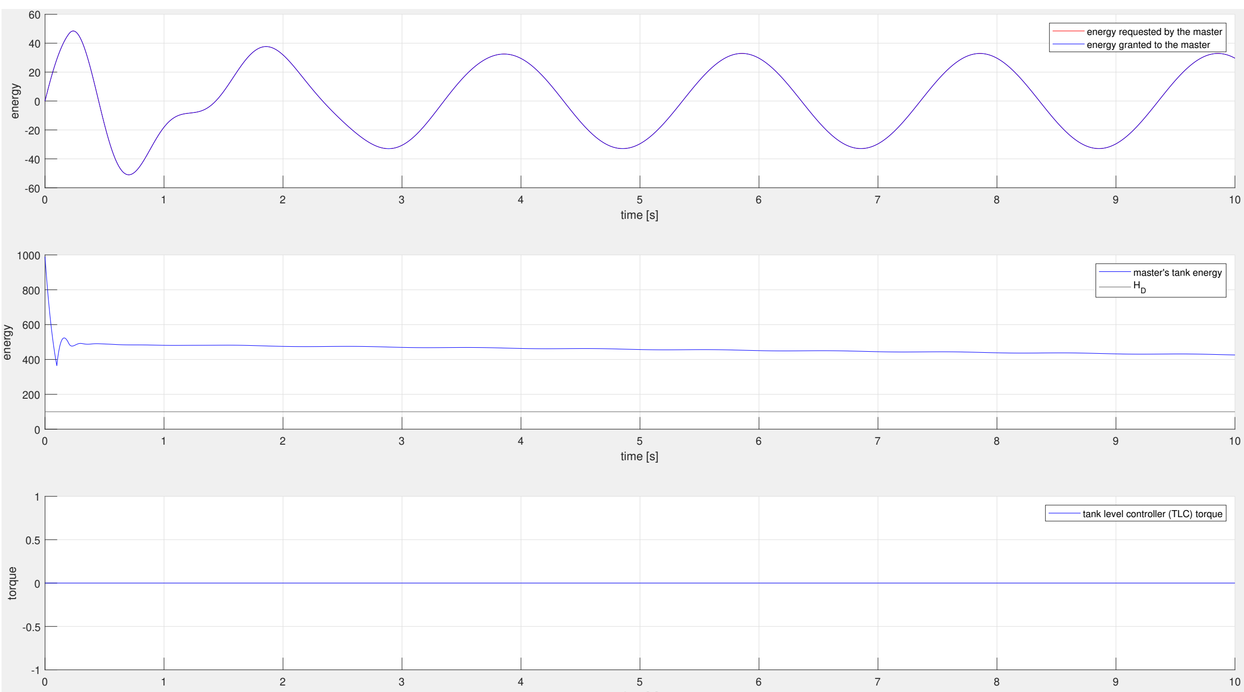

Figure 32: Tank-based Position-Position: free motion, $\beta = 0.01$, $\alpha = 0.2$

$\beta = 0.1$, $\alpha = 0.2$. Going from $\beta = 0.01$ to $\beta = 0.1$, the initial energy is not enough. By increasing β the trajectory is more energy-intensive.

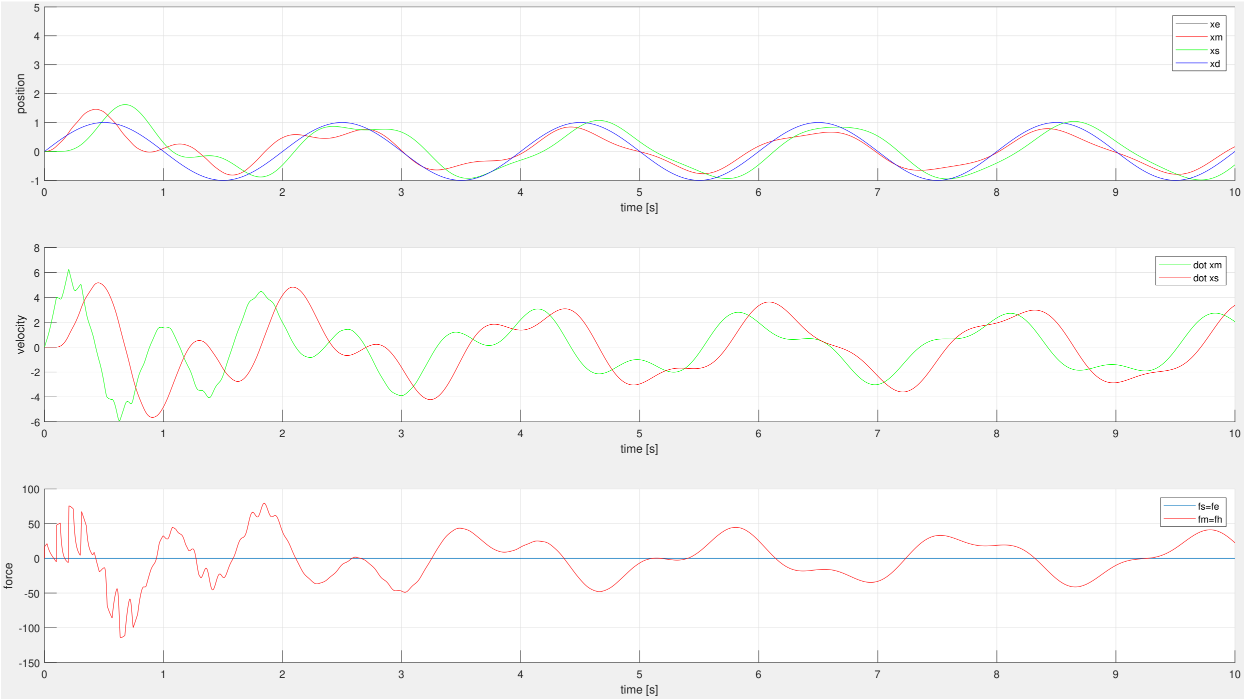

Figure 33: Tank-based Position-Position: free motion, $\beta = 0.1$, $\alpha = 0.2$

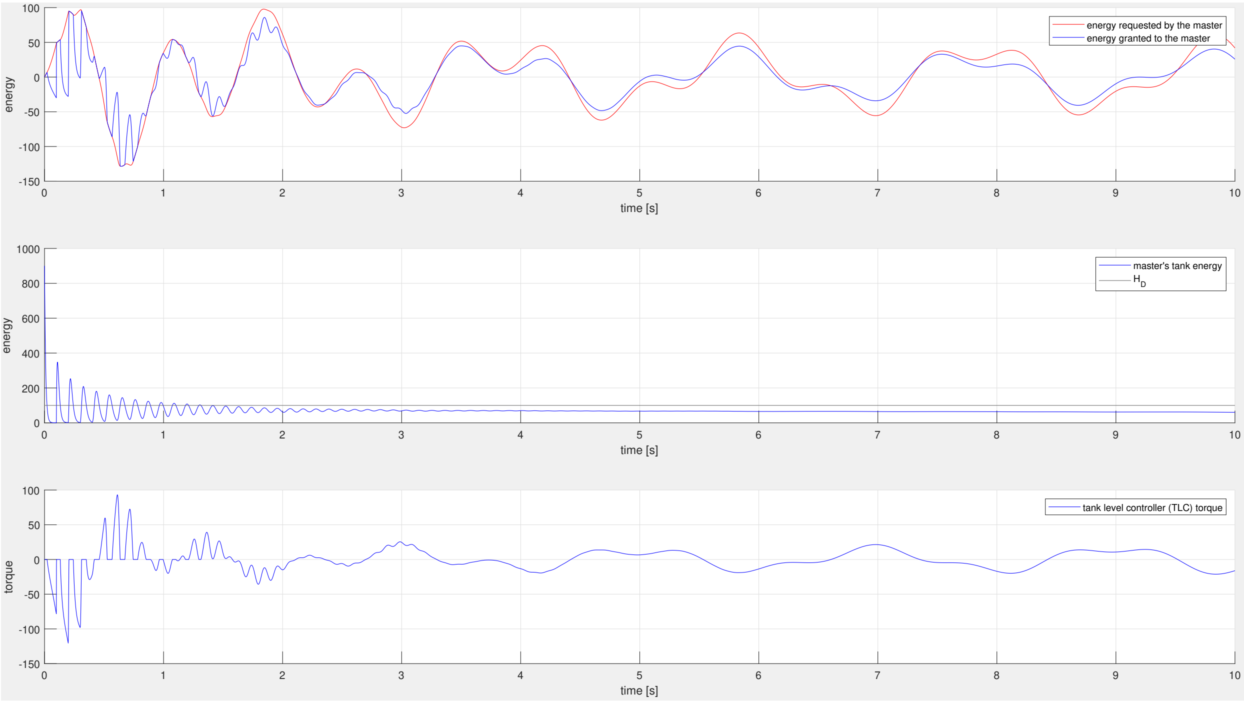

Figure 34: Tank-based Position-Position: free motion, $\beta = 0.1$, $\alpha = 0.2$

$\beta = 0.1$, $\alpha = 0.02$. Going from $\alpha = 0.2$ to $\alpha = 0.02$ we are harvesting less energy from the operator.

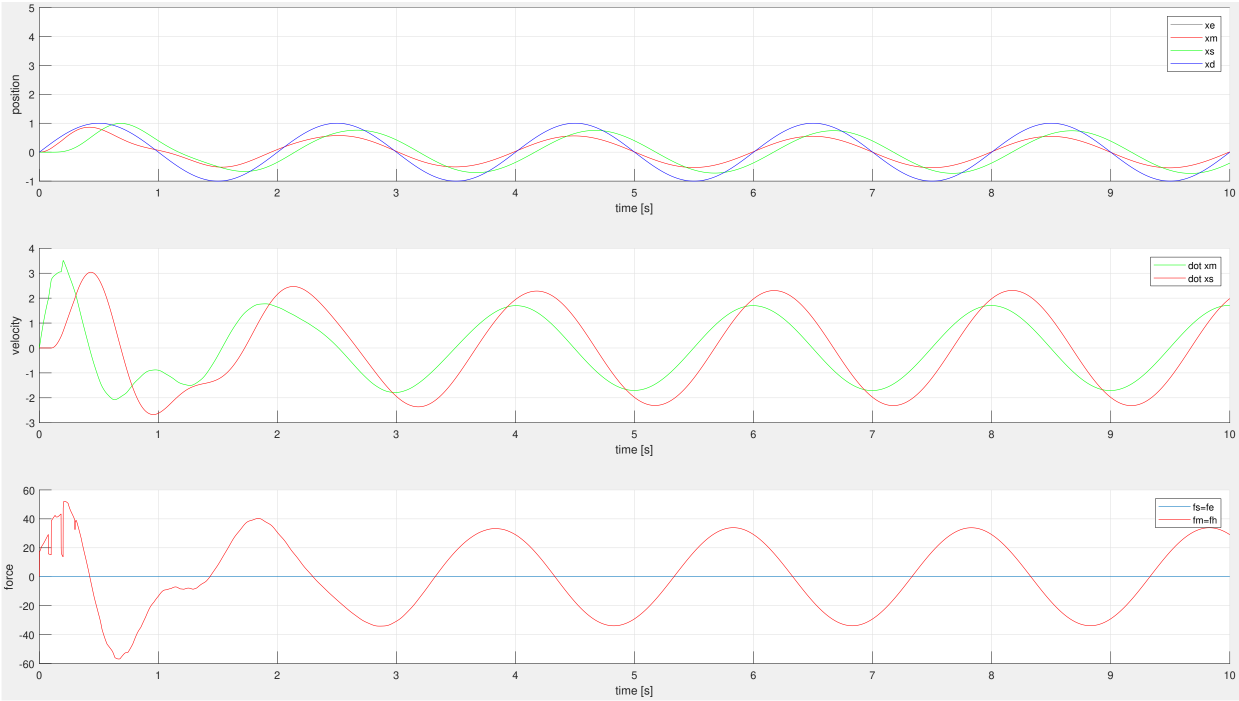

Figure 35: Tank-based Position-Position: free motion, $\beta = 0.1$, $\alpha = 0.02$

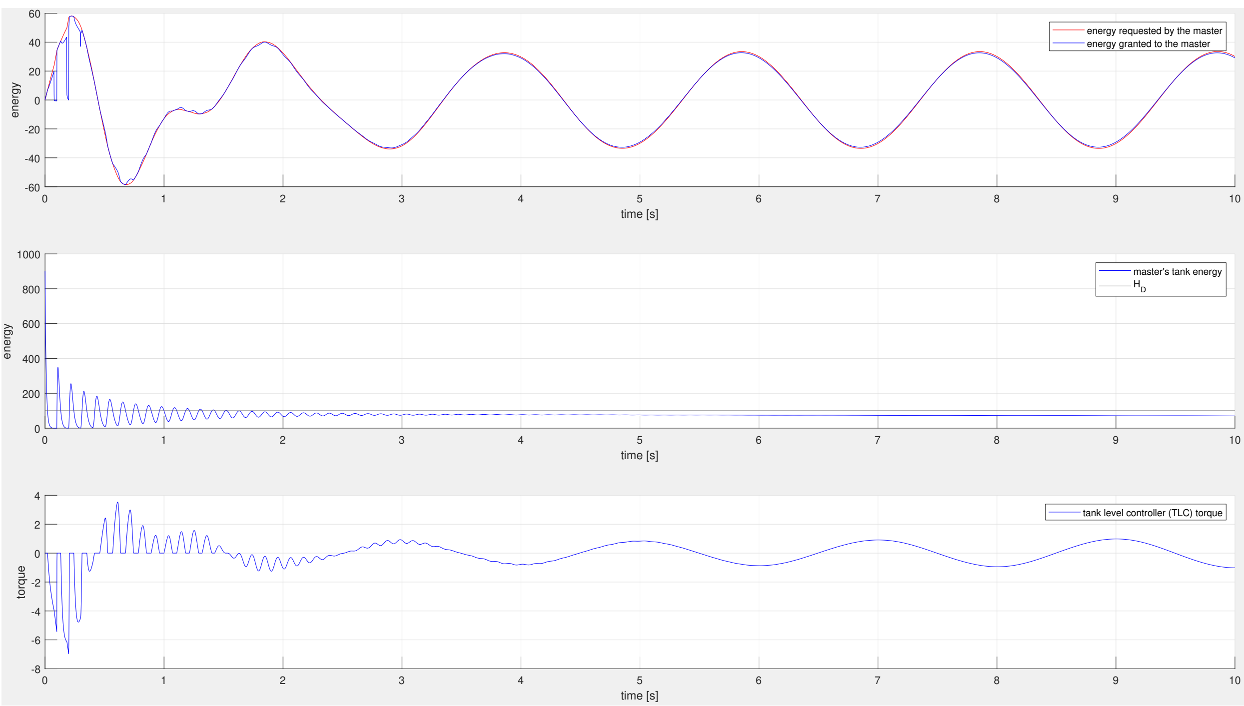

Figure 36: Tank-based Position-Position: free motion, $\beta = 0.1$, $\alpha = 0.02$

$\beta = 0.1$, $\alpha = 0.2$ and in contact with the environment.

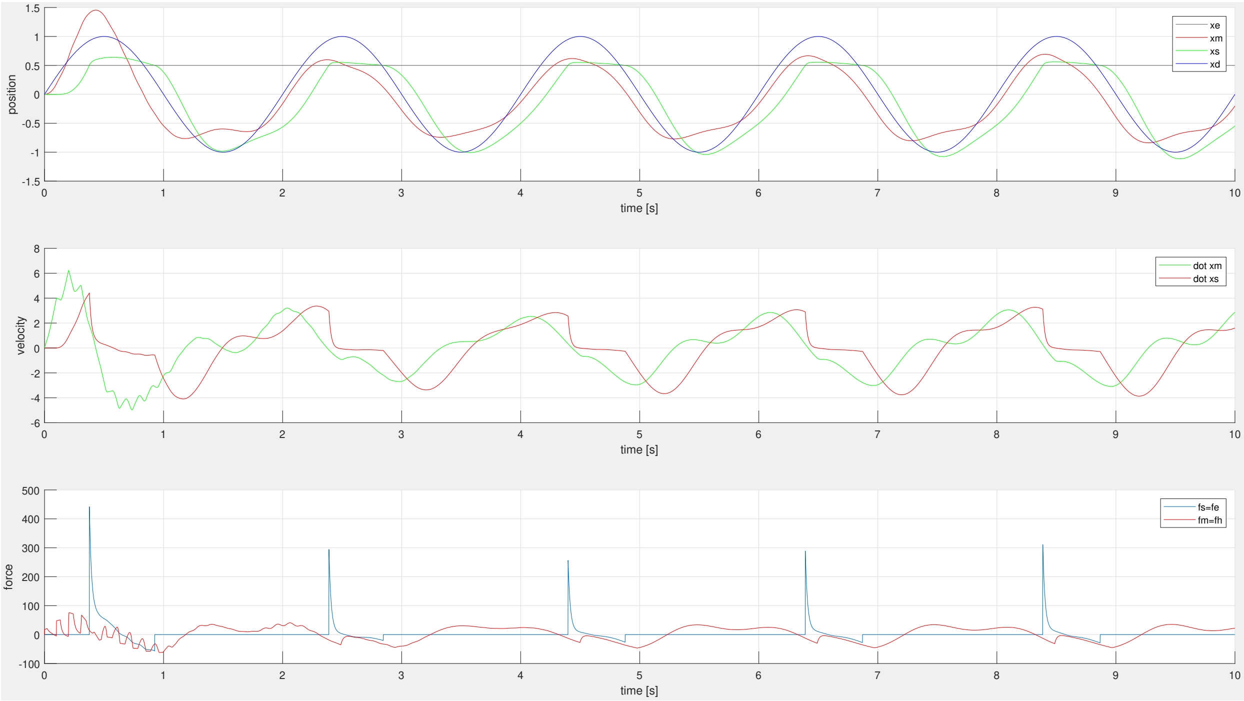

Figure 37: Tank-based Position-Position: contact, $\beta = 0.1$, $\alpha = 0.2$

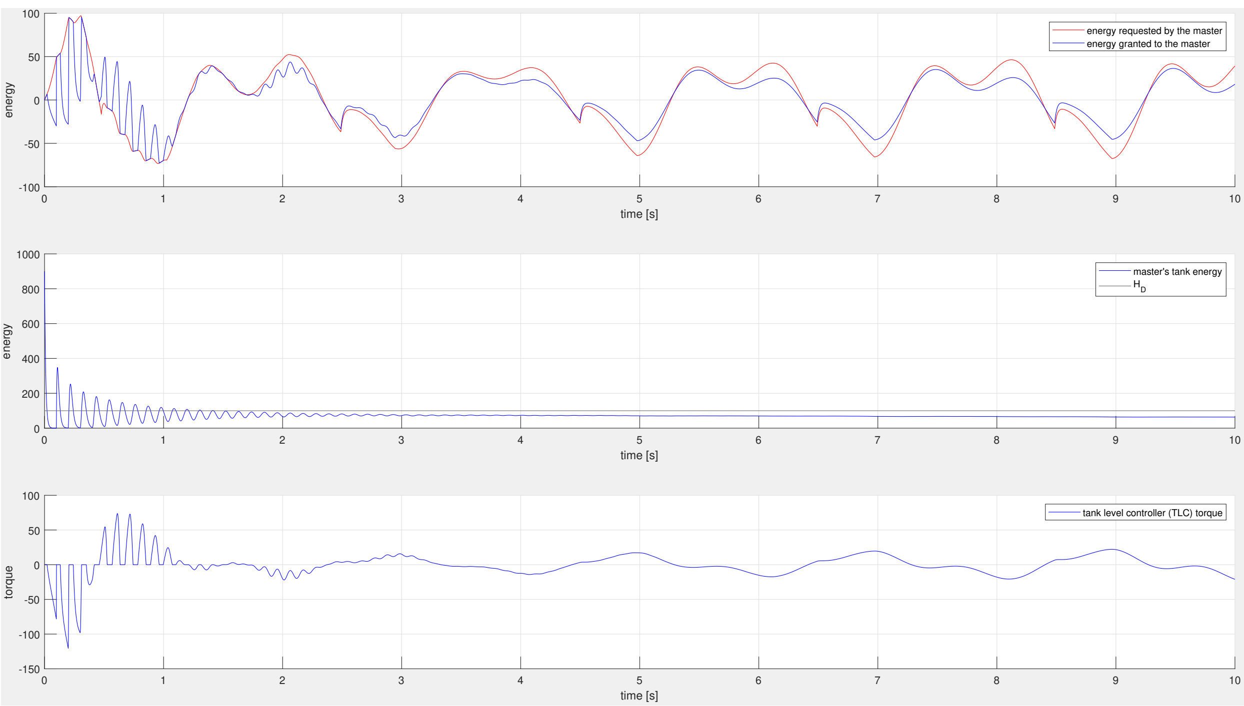

Figure 38: Tank-based Position-Position: contact, $\beta = 0.1$, $\alpha = 0.2$

6.4 Tank-based Force-Position

6.4.1 Sufficient initial energy

$\beta = 0.1$, $\alpha = 0.1$ and in contact with the environment. Initial energy is enough.

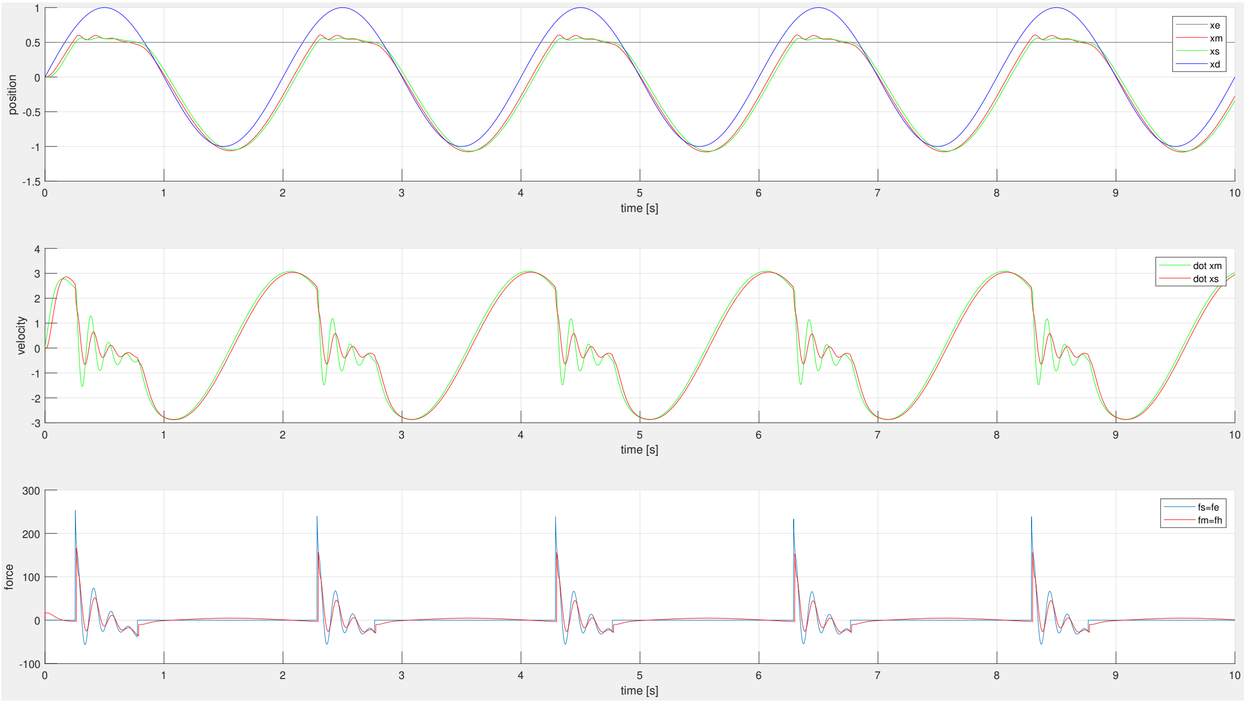

Figure 39: Tank-based Force-Position: contact, $\beta = 0.1$, $\alpha = 0.1$

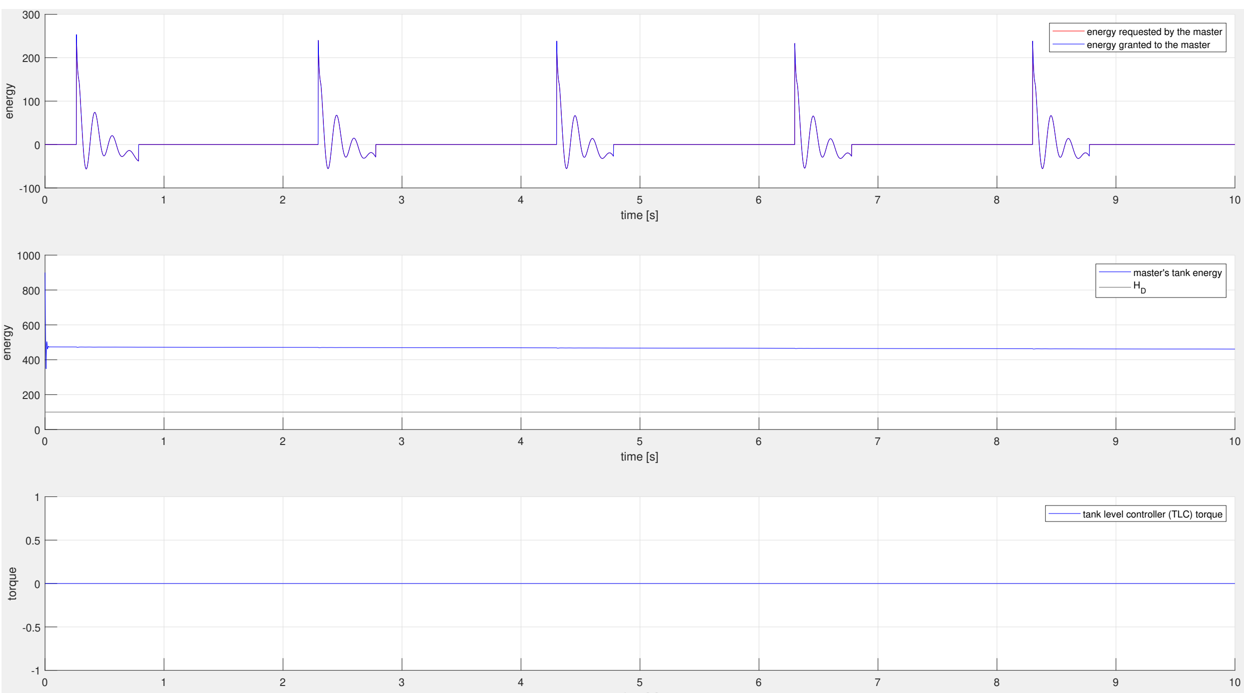

Figure 40: Tank-based Force-Position: contact, $\beta = 0.1$, $\alpha = 0.1$

6.4.2 Not sufficient initial energy

With $\beta = 0.1$, $\alpha = 0.1$ and in contact with the environment. Initial energy is not enough.

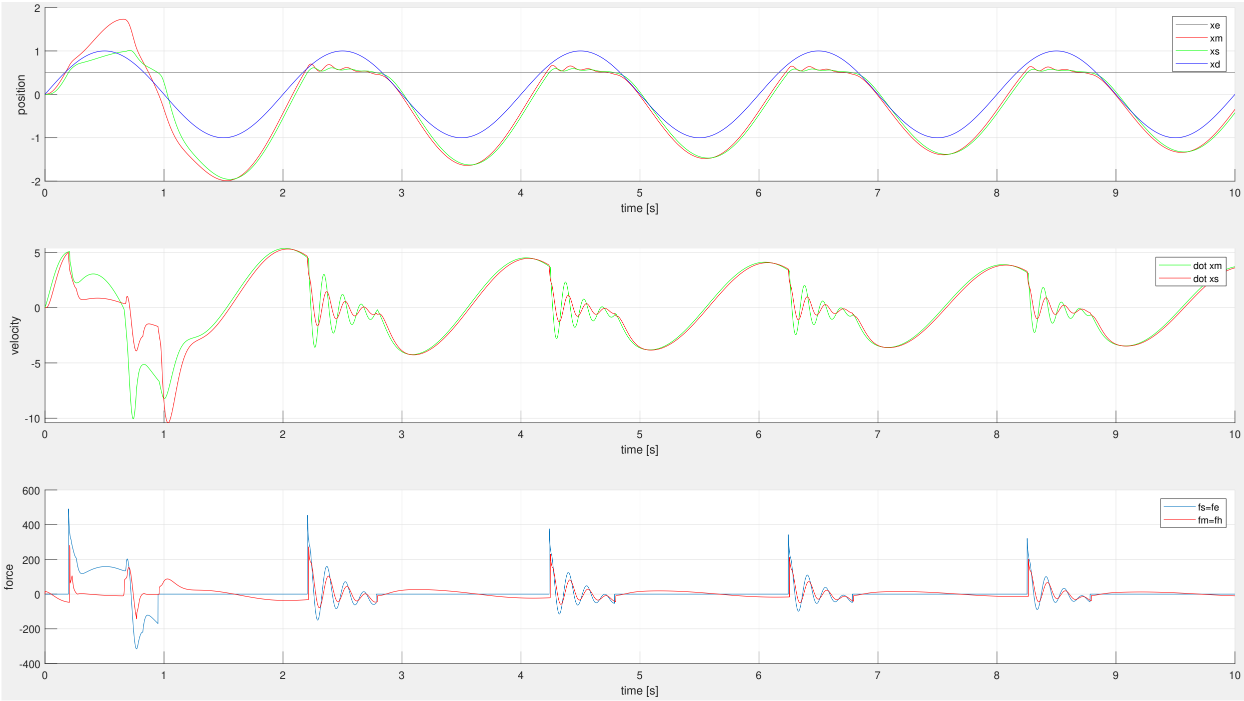

Figure 41: Tank-based Force-Position: contact, $\beta = 0.1$, $\alpha = 0.1$

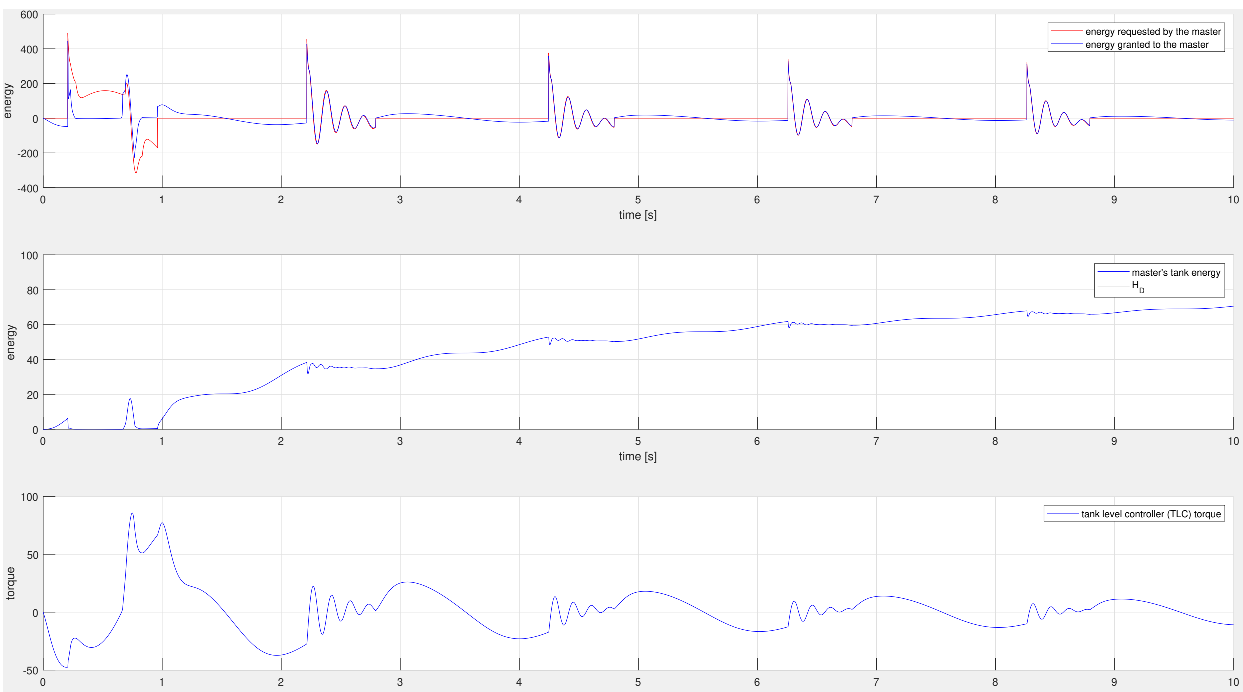

Figure 42: Tank-based Force-Position: contact, $\beta = 0.1$, $\alpha = 0.1$

6.4.3 Velocity estimation from noisy position measurement

Gaussian noise (mean = 0, variance = 0.00001) is added to the master position x_m. The master velocity is estimated from the noisy position signal using kalman (filter, predictor, smoother). The $RMSE$ is computed with respect to the reference velocity signal exported from simulink and it is reported in the legend.

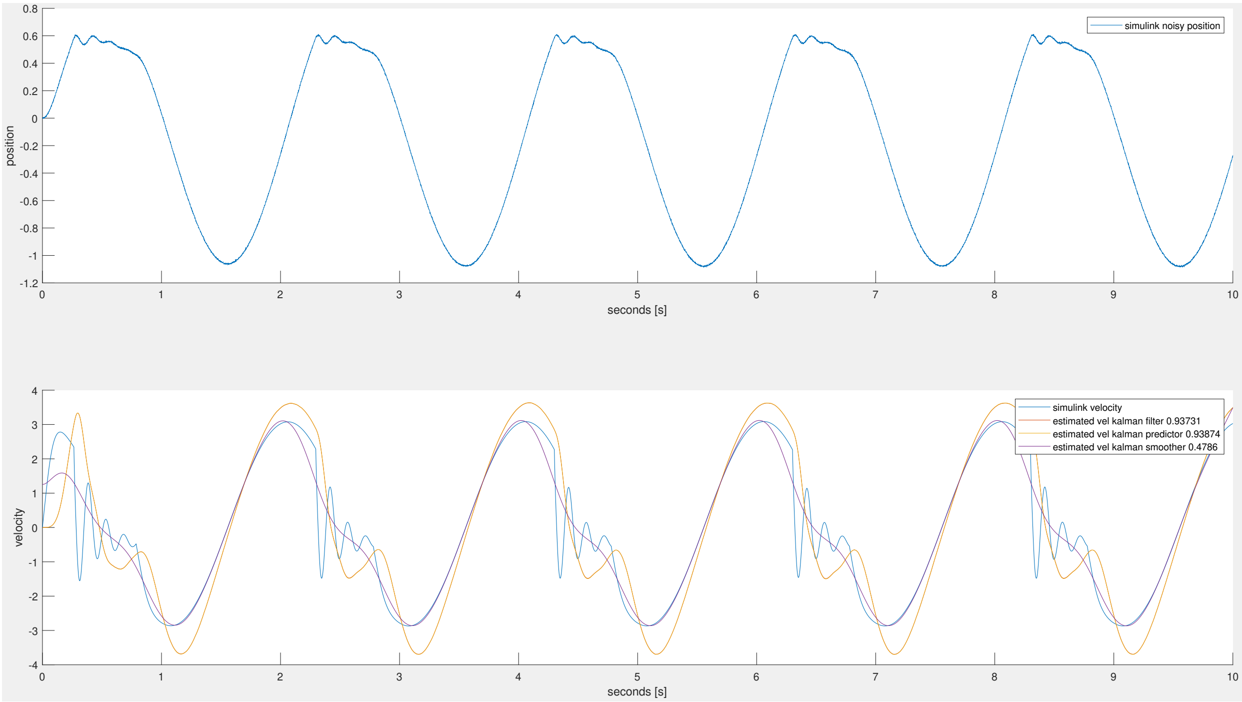

Figure 43: Tank-based Force-Position: master's velocity estimation from noisy position measurement of the master