



UNIVERSITÀ
di **VERONA**

Physical Human Robot Interaction

Physical Human Robot Interaction
(A.Y. 2022/2023)
Department of Computer Science

Table of Contents

Abstract.....	3
1. Introduction	3
2. Four-Channel Bilateral Teleoperation	4
3. Two Channel teleoperations	15
3.1. Two-Channel Position-Position (P-P) teleoperation	15
3.2. Two-Channel Force-Position (F-P) teleoperation.....	18
3.3. Two-Channel Force-Force (F-F) teleoperation.....	23
3.4. Three-Channel Position-Force and Position (PF-P) teleoperation.....	26
4. Kalman Filter/Predictor	29
4.1. Kalman Filter	31
4.2. Kalman Predictor.....	31
4.3. Kalman Steady-State	34
5. Kalman Smoother	35
6. Linear Methods for Regression	37
6.1. Least Square	38
6.2. Recursive Least Square.....	39
7. Scattering-based Bilateral Teleoperation Architecture.....	40
7.1. Force-Position (F-P)	42
7.2. Position-Position (P-P).....	46
8. Tank-based bilateral teleoperation architecture	53
8.1. Force-Position (F-P).....	56
8.2. Position-Position (P-P).....	59
9. Conclusion	61
References	62

Abstract

Teleoperation refers to the remote control of a robot by a human operator, allowing the operator to perform tasks in environments that may be dangerous or inaccessible to humans. The literature provides a detailed analysis of various teleoperation architectures, including Four-Channel Bilateral Teleoperation, Two-Channel Teleoperation, and Three-Channel Position-Force and Position (PF-P) teleoperation. These architectures are designed to enable effective communication between a human operator and a robot, allowing the operator to control the robot's movements and actions. The article also discusses the use of Kalman Filter/Predictor and Kalman Smoother in teleoperation. The torque of the motor is predicted using the linear regression. These techniques are used to improve the accuracy of position, velocity, and force estimation in teleoperation systems. The tank-based and scattering wave-based architecture is used to eliminate the significance delays in the communication channel. This literature provides valuable insights into the field of Physical Human Robot Interaction and its applications in teleoperation.

1. Introduction

Physical human-robot interaction (pHRI) is a subfield of robotics that studies how humans and robots can physically interact with each other in a safe and efficient manner[1]. Unlike traditional human-robot interaction (HRI), which focuses on the cognitive and social aspects of communication, pHRI deals with the physical contact and force exchange between humans and robots. pHRI is important for applications such as collaborative manufacturing, assistive robotics, rehabilitation robotics, surgical robotics, and social robotics. pHRI poses many challenges and opportunities for roboticists, such as designing compliant and adaptive control strategies, ensuring safety and comfort of human users, modeling human behavior and preferences, and developing intuitive and natural interfaces. pHRI also raises ethical and social issues, such as the impact of robots on human dignity, autonomy, and well-being.

Teleoperation is the process of controlling a robot or a machine remotely by a human operator. Teleoperation allows the human operator to perform tasks that are too dangerous, difficult, or distant for direct human intervention [2]. For example, teleoperation can be used to operate robots in hazardous environments such as nuclear plants, mines, or space; to perform delicate surgeries or manipulations that require high precision and dexterity; or to explore and interact with remote locations such as the ocean floor or other planets.

Teleoperation has several advantages and challenges in robotics. One advantage is that it can leverage human intelligence and skills to complement the robot's capabilities and overcome its limitations. Another advantage is that it can provide a sense of presence and immersion for the human operator, which can enhance the user experience and satisfaction[3]. Some challenges of teleoperation are: ensuring reliable and secure communication between the human and the robot; providing adequate feedback and information to the human operator; designing intuitive

and ergonomic interfaces and controllers; and dealing with latency, bandwidth, and noise issues that can affect the performance and safety of the teleoperation system.

This scientific article is divided into several subsections. The 2nd section introduces the concept of the four-channel bilateral teleoperation while the 3rd section will discuss its variation of two channel and three channels. The 4th section will discuss the significance of Kalman Filter and its practical uses. The 5th section will elaborate the working of the Kalman smoother and how it is different from standard filter. The 6th section will help to predict the velocity and acceleration with the help of linear regress using least square method. Section 7 and 8 will discuss the teleoperation architecture to eliminate the delay in the communication channel. Last but not least, the 9th section will conclude all the literature.

2. Four-Channel Bilateral Teleoperation

Four-Channel Bilateral Teleoperation is the architecture based on four signals; the two signals are from the master side and two signals are from the slave side [4]. These signals are of the form of force and velocities of slave and master robot. However, for the sake of this architecture it is assumed that there is no delay in the telecommunication channels, forces, and velocities for both slave and master robot can be achieved by the sensors, and perfect knowledge of dynamics are known for slave and master robot. The goal is to design the controller and teleoperation channel for the slave and master robot. The architecture is displayed in Figure 1.

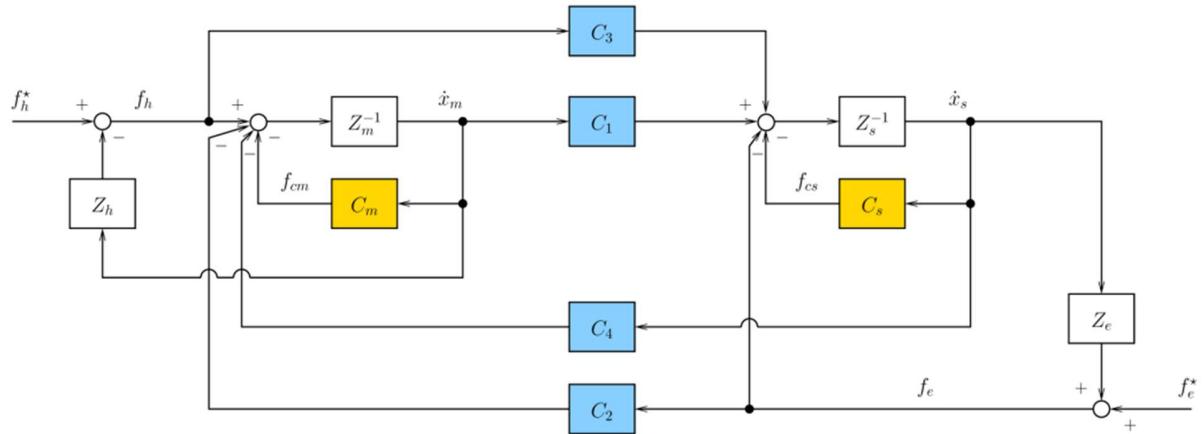


Figure 1 Four-Channel Bilateral Teleoperation Architecture

To achieve transparency in a master-slave robotic system, it is essential to cancel out the dynamics of the robot itself. This is typically done using inverse dynamics, which involves calculating the torques or forces required to achieve a desired motion of the robot. In addition, it is crucial to ensure that the forces fed forward by the master robot match the net forces exerted by the operator and the environment. This means that the slave robot should not apply any

additional forces or torques beyond what is necessary to move in response to the operator's commands [5]. However, following condition should be fulfilled by the controller.

$$C_3 C_2 = I \quad (1)$$

$$C_4 = -(Z_m + C_m) \quad (2)$$

$$C_1 = Z_s + C_s \quad (3)$$

$$C_2 = I \quad (4)$$

The Equation 1 and 4, implies that

$$C_3 = I \quad (5)$$

By using Equations 1-5, Figure 2 shows the perfect transparent four-channel bilateral teleoperation architecture.

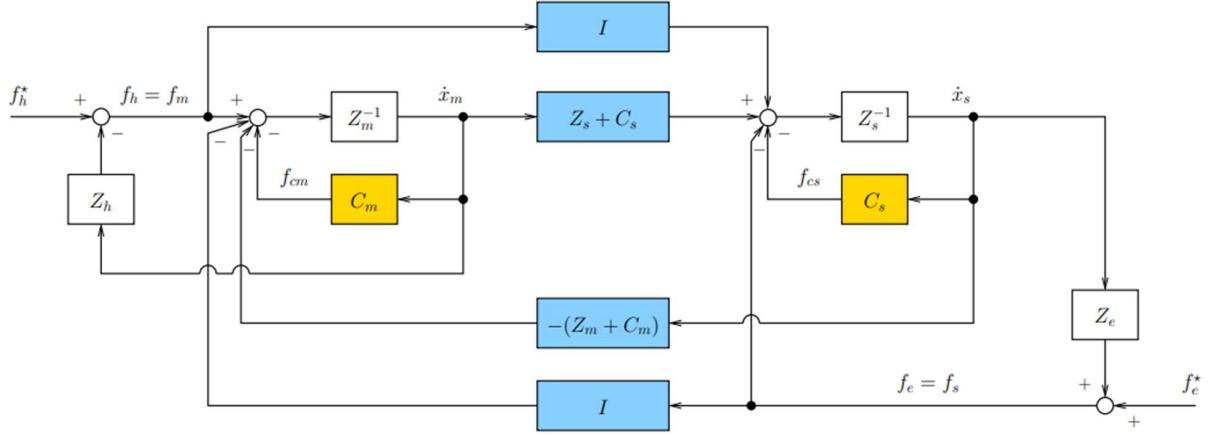


Figure 2 Transparent four-channel bilateral teleoperation architecture

In the Figure 2, Z_m^{-1} and Z_s^{-1} are the robot dynamics while Z_m and Z_s are robot inverse dynamics for the master and slave robot respectively.

When the robot dynamics are not known then it is possible to skip Z_m and Z_s for master and slave robot. The controller's channel will form the shape of

$$C_3 = I \quad (6)$$

$$C_4 = C_m \quad (7)$$

$$C_1 = C_s \quad (8)$$

$$C_2 = I \quad (9)$$

To model the visual feedback, the C_h controller is used as shown in the Figure 3. To estimate the intention force of f_h^* , the desired position and velocity of slave robot x_s^d, \dot{x}_s^d is compared with

the actual slave position and velocity x_s, \dot{x}_s . The C_h controller is modeled as PD controller describe below

$$f_h^* = P_h(x_s^d - x_s) + D_h(\dot{x}_s^d - \dot{x}_s) \quad (10)$$

Furthermore, in real life the impedance of human and environment is not known. But for the sake of this experiment, the impedance of human hand can be modeled as

$$Z_h = J_h \ddot{x}_m + B_h \dot{x}_m \quad (11)$$

and

$$f_h = f_h^* - Z_h \quad (12)$$

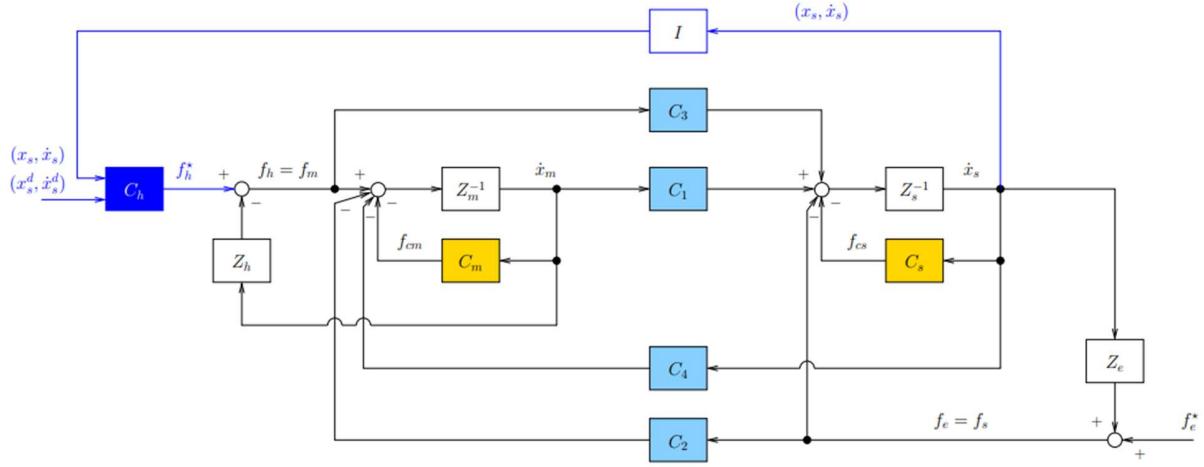


Figure 3 Four-channel bilateral teleoperation architecture with human intention.

The environment is modeled as spring K_e and damping B_e , the environment impedance directly map velocity into force which is model defined as

$$f_e = \begin{cases} K_e(x_s - x_e) + B_e \dot{x}_s & \text{if } x_s > x_e \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

However, in this experiment, the noise is ignored as well as other uncertainties are not taken into account as well. The slave and master robot model Z_m^{-1} and Z_s^{-1} is defined by the simple inertia dynamics as well as with the addition of the damping.

$$Z_m^{-1} = \frac{1}{M_m s} \text{ or } Z_m^{-1} = \frac{1}{M_m s + D_m} \quad (14)$$

$$Z_s^{-1} = \frac{1}{M_s s} \text{ or } Z_s^{-1} = \frac{1}{M_m s + D_m} \quad (15)$$

where M is the inertia of the robot and D is the damping of the robot. The master and slave controller C_m, C_s is modeled by the law of PD controller on the position which has a form of

$$C_m = B_m \dot{x}_m + K_m x_m \quad (16)$$

$$C_s = B_s \dot{x}_s + K_s x_s \quad (17)$$

Figure 4 shows the Four-channel bilateral teleoperation architecture with human intention in Simulink simulation. The two inputs i.e. sine wave and step function with low pass filter is used to produce the desired position x_s^d . The light blue blocks are the communication channel where the outer channels are force, and the inner blocks are the velocity channels. The yellow blocks are the controllers defined in Equation 16 and 17 for the master and slave robot. Furthermore, the dark blue color controller is the human intention controller which is defined in Equation 10.

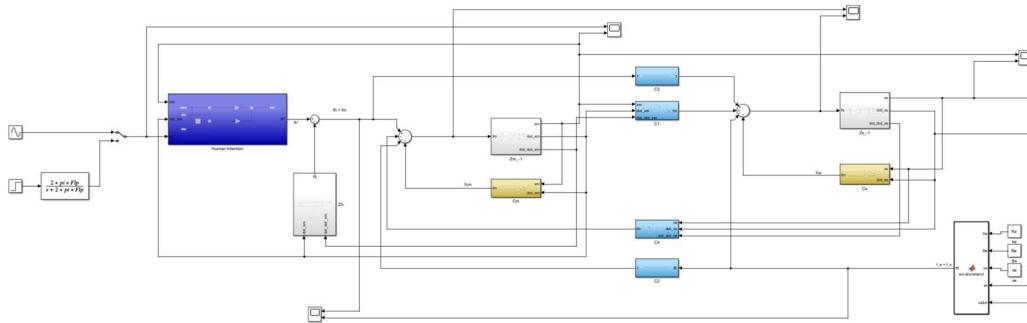


Figure 4 Four-channel bilateral teleoperation architecture with human intention in Simulink simulation.

The robot dynamics is important in the controller, the robot dynamics responsible for modeling the robot behavior in the virtual domain. In the Figure 5, it can be seen that the robot dynamics is defined as simple inertia as well as simple inertia and damping. These two scenarios can be selected separately by using manual switch. However, this controller converts the given force to its respective position, velocity and acceleration. It is important to note that the controller displayed in the image is for the master robot controller, though similar controller can be designed for the slave robot with its own dynamics parameters.

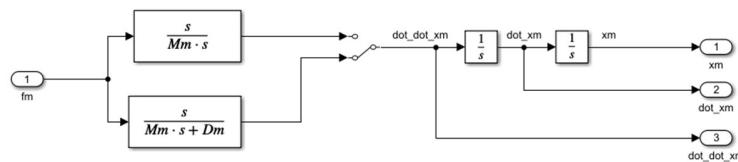


Figure 5 Robot dynamics with switch for damping selection.

The Figure 6 shows the communication channel more specifically C_1 . This channel has compacity to switch between perfect transparency as describe in Equations 1-4 and between the simple

communication channel describe in Equation 6-9 when the perfect dynamic model is not available. Similar channel can be defined for the C_4 channel with the negative sign at the end of the output with its own parameters.

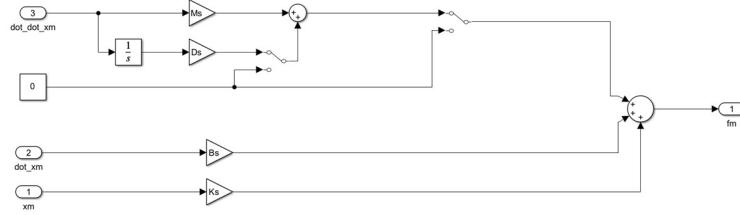


Figure 6 Communication channel with selection of inverse dynamics via switch.

To evaluate the performance of above controller, different graphs along with different scenarios are generated. The Figure 7, 8 and 9 shows the graph for the controller with perfect transparency and free motion which means that the inverse dynamics i.e. Z_m and Z_s is used in the channels. As it can be seen that the Figure 7 shows the graph between the x_m , x_s and x_d . The position of master and slave robot overlap accurately while the desired position do not overlap with each other, the possible reason about this behavior is the tuning of the PD parameters of the C_h controller as in this report the default parameters are used. Furthermore, according to Figure 8, the velocities of both slave and master robots are matched perfectly in the free motion. Moreover, the forces between the slave and master robot matches perfectly as well which is shown in the Figure 9. It is important to note that, the force of human f_h or f_m is considered zero as it is very small value.

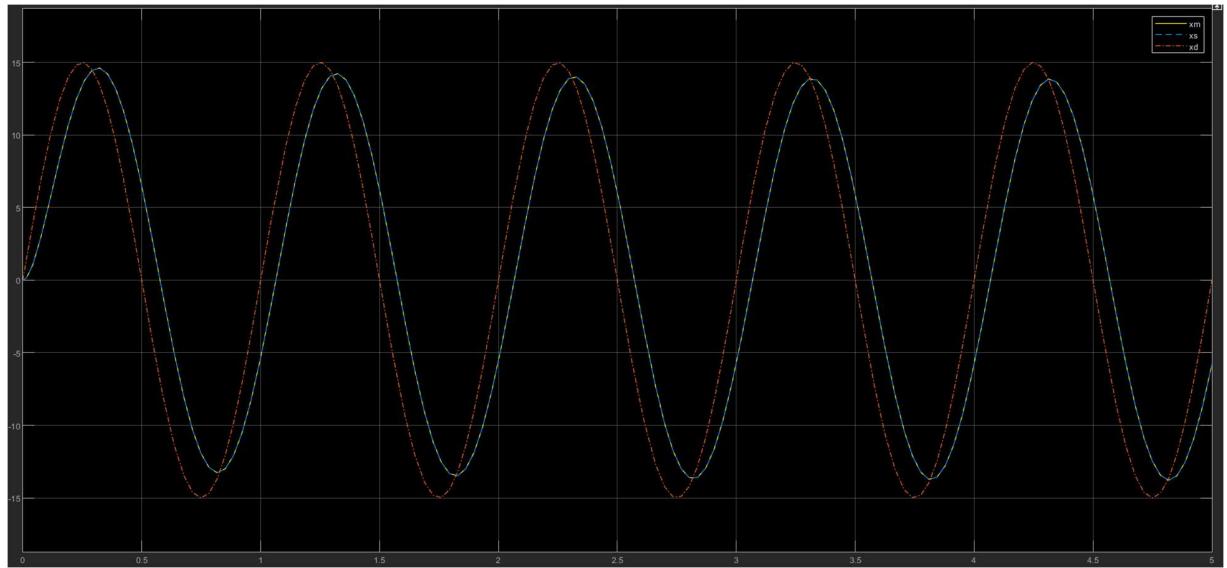


Figure 7 Position graph for the teleoperation with perfect transparency in free motion.

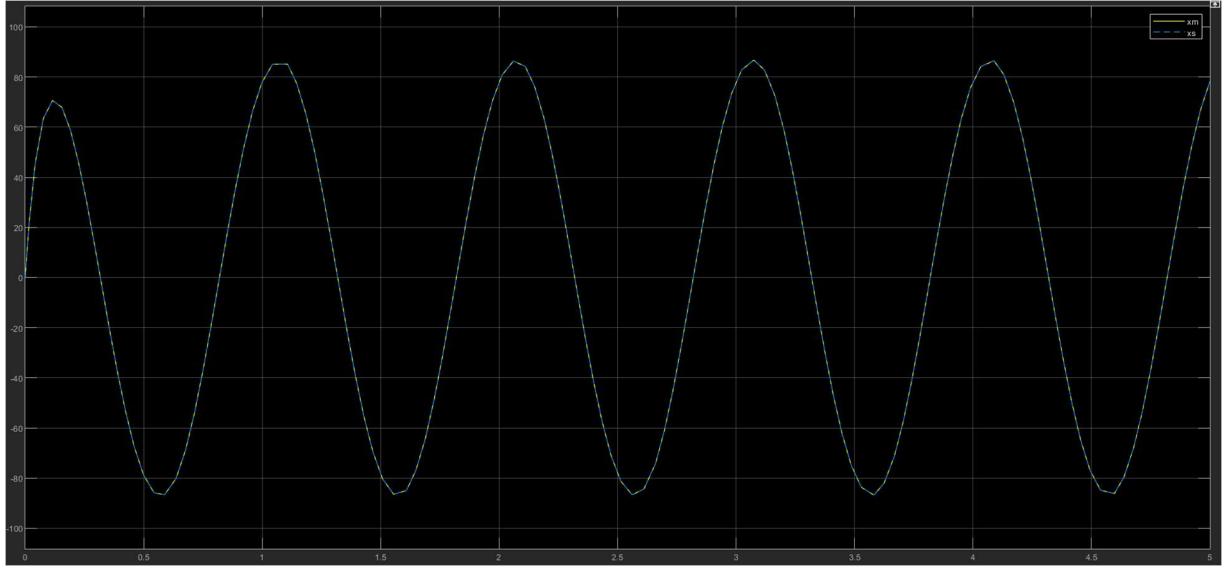


Figure 8 Velocity graph for the teleoperation with perfect transparency in free motion.

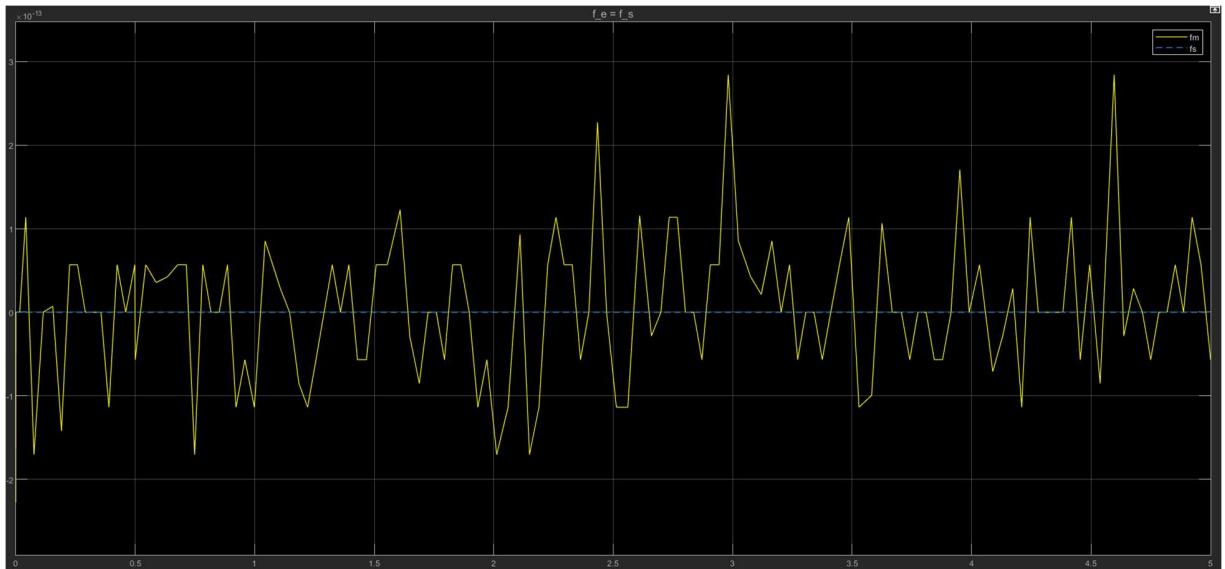


Figure 9 Slave and master robot's force graph for the teleoperation with perfect transparency in free motion.

Figure 10,11 and 12 shows the output in regards of position, velocity and force in free motion. The input to the controller is the step function which then pass through the low pass filter. The low pass filter cut off the higher frequencies and make the output much smoother. According to Figure 10, the master robot position accurately overlapped with the slave robot position, however the desired position graph did not overlap accurately as described above, the default parameters are used without tuning. The Figure 11 shows the graph for the velocity where as same as position, the master and slave graph overlap completely because of the inverse dynamics are used in the channels for perfect transparency. Moreover, the Figure 11 shows the graph for the force of master and slave, it can be seen that the forces of both robots match accurately.

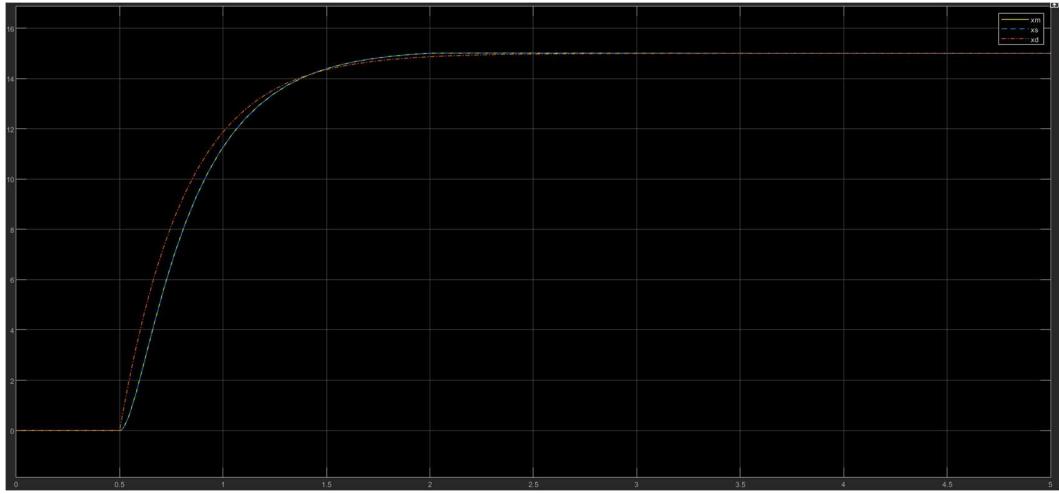


Figure 10 Position graph for the teleoperation with perfect transparency in free motion (step function).

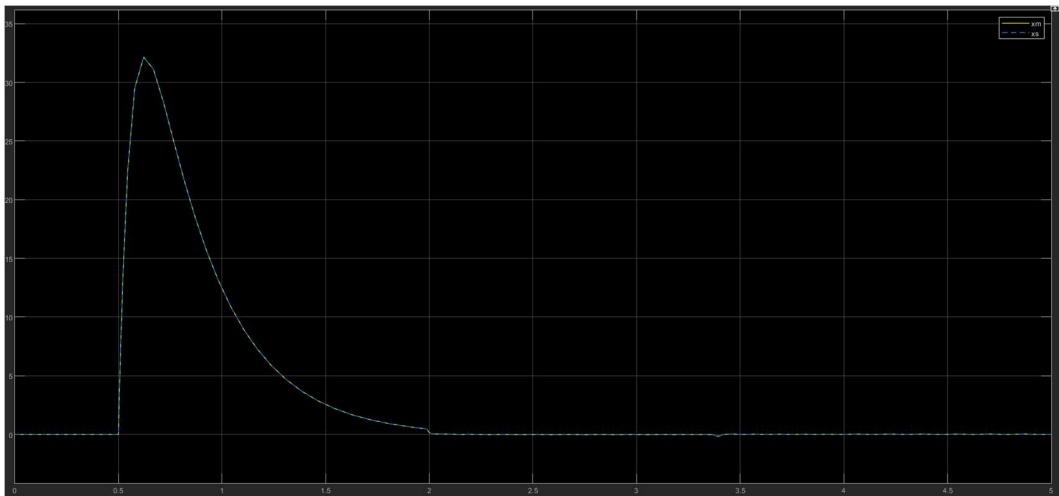


Figure 11 Velocity graph for the teleoperation with perfect transparency in free motion (step function).

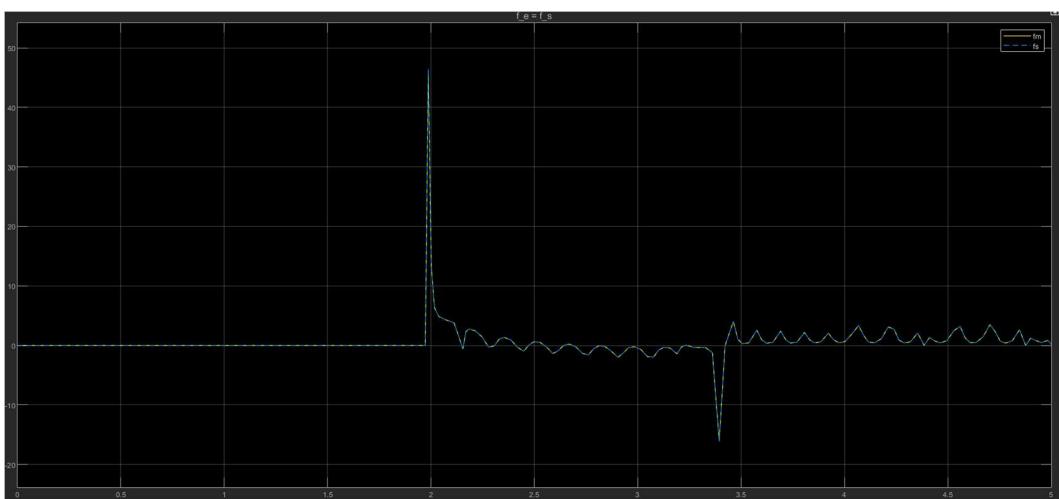


Figure 12 Slave and master robot's force graph for the teleoperation with perfect transparency in free motion (step function).

When there is no perfect transparency which means that there is a drift between the master and slave robot is due to not available of accurate dynamics of the robot. The Figure 13, 14 and 15 shows the graph of the position, velocity and force respectively without perfect transparency in the free motion. It can be seen that there is a small drift between the slave and master robot is due to the absence of the inverse dynamics in the communication channel. Similarly, the velocities also show some drift between the slave and master robots according to Figure 14. Furthermore, the slave and master force did not match as well because as stated earlier, the net forces exerted by the master should be equal to the net force exerted by the slave robot. However, it cab be seen in Figure 15 the forces of master and slave are not equal.

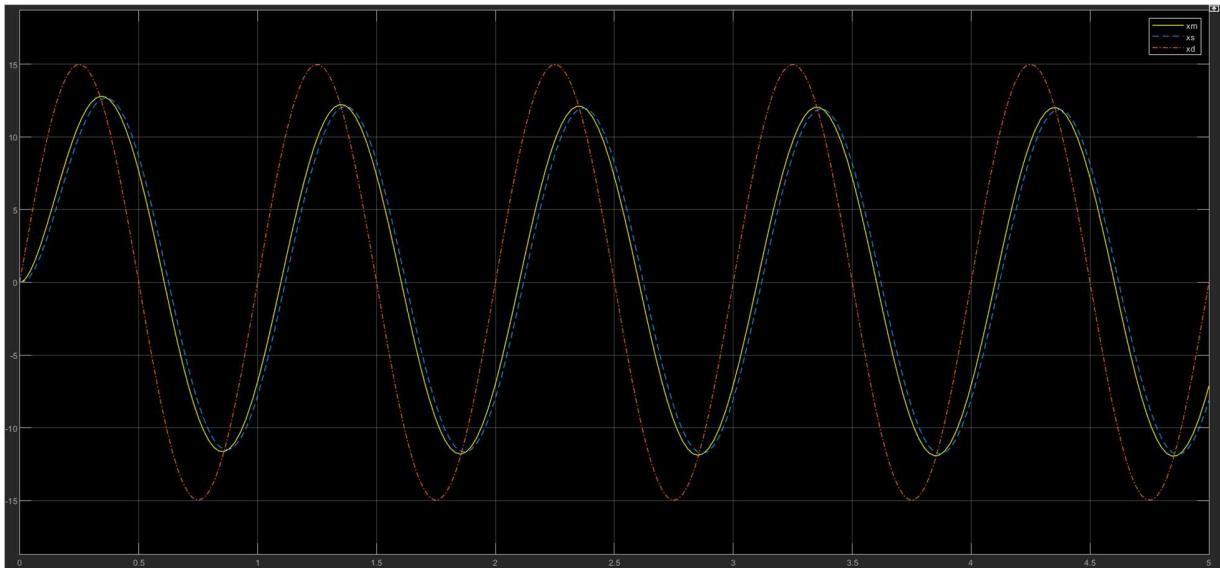


Figure 13 Position graph for the teleoperation without perfect transparency in free motion.

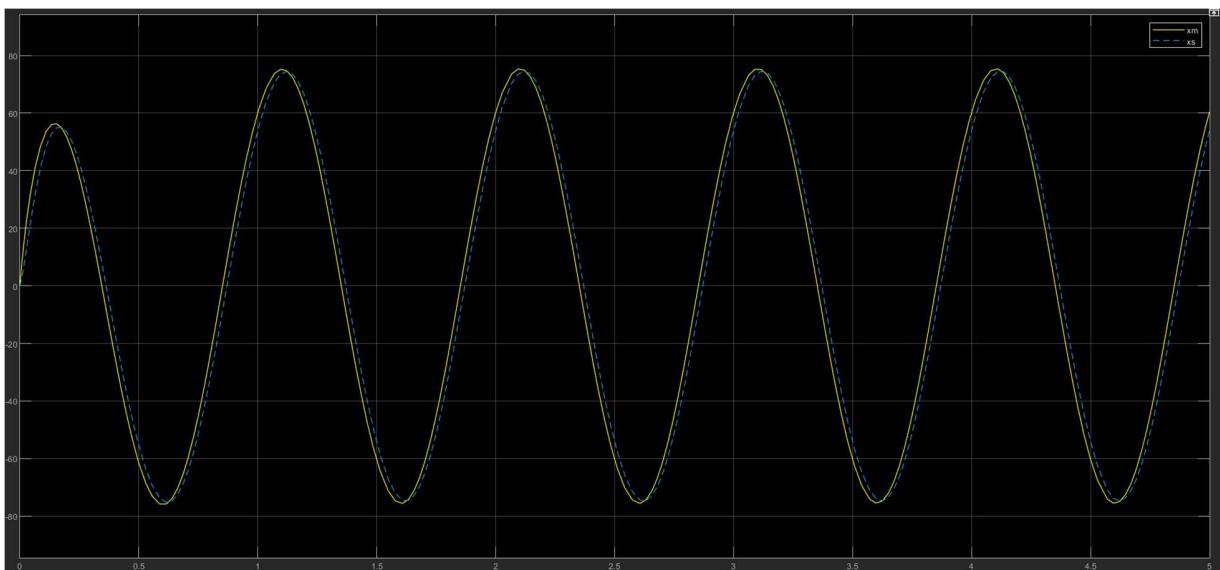


Figure 14 Velocity graph for the teleoperation without perfect transparency in free motion.

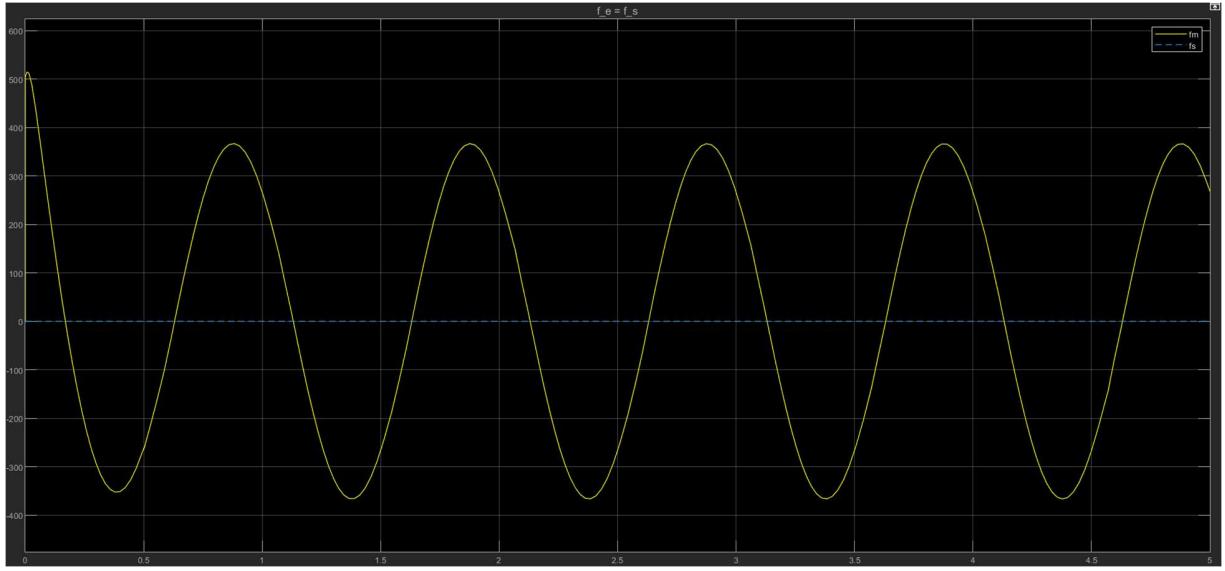


Figure 15 Slave and master robot's force graph for the teleoperation without perfect transparency in free motion.

Similarly, without perfect transparency, the graphs that is generated using the step function response shows the drift as well. According to the Figures 16, 17 and 18 the graphs between the slave and master robots has some drift due to the no perfect dynamics of the robot is available. The figure 16 show the position graph for the step response, as it uses default parameters the desired position did not overlap with the master and slave robot. Similarly, the graph between the slave and master for the velocities has some small drift, while for the forces, the master graph command the slave to apply some forces to the environment but the slave robot did not interact to the environment due to the position is not achieved by the slave to interact with the environment as shown in Figure 18.

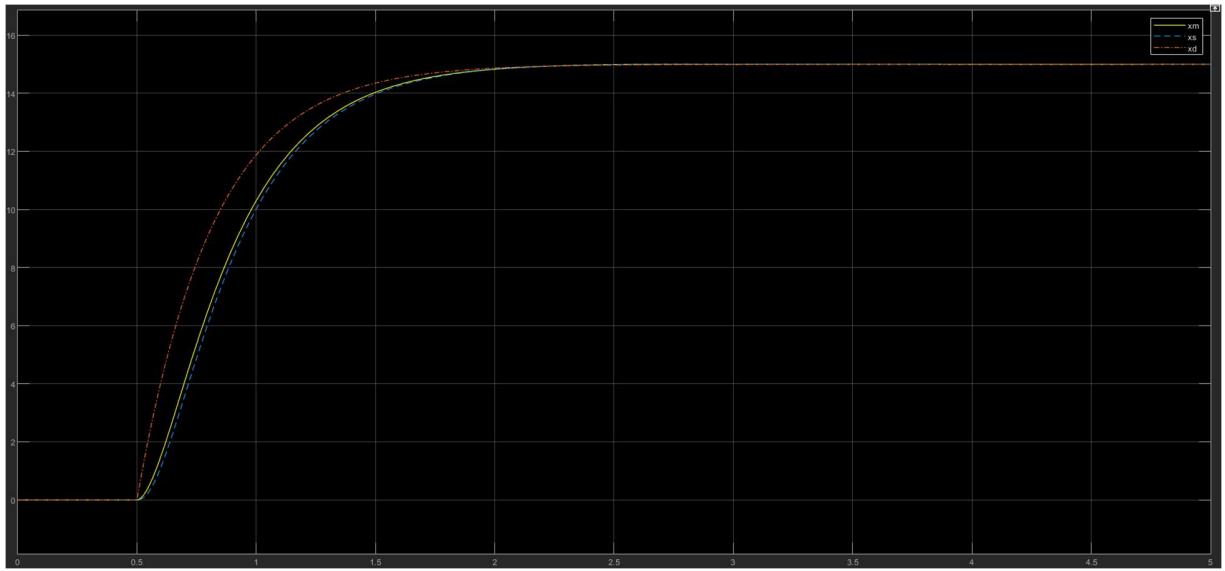


Figure 16 Position graph for the teleoperation without perfect transparency in free motion (step function).

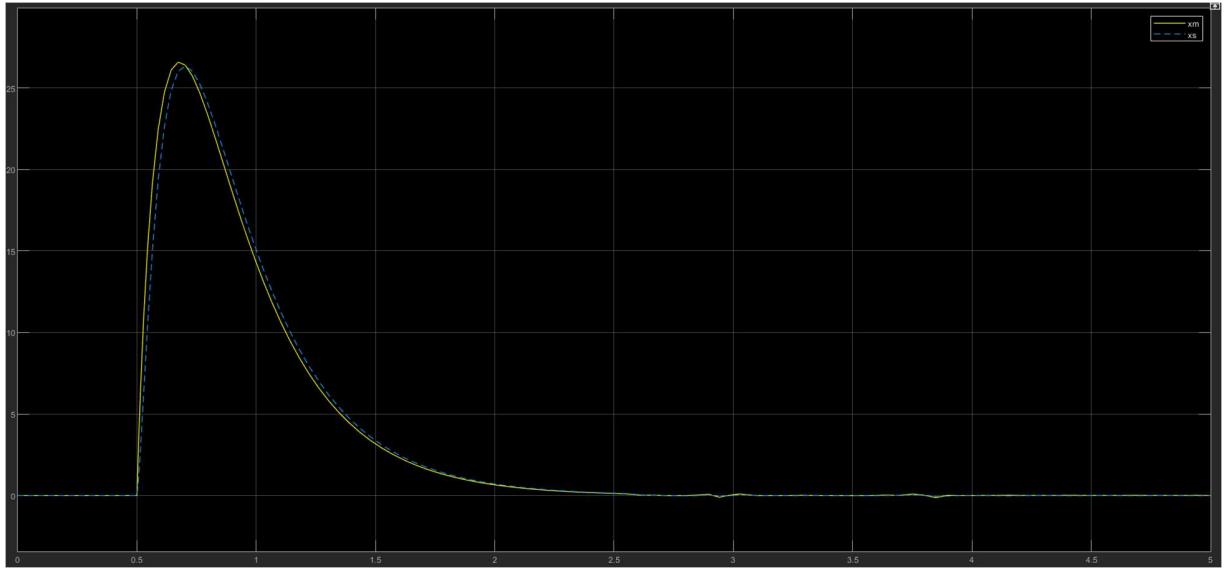


Figure 17 Velocity graph for the teleoperation without perfect transparency in free motion (step function).

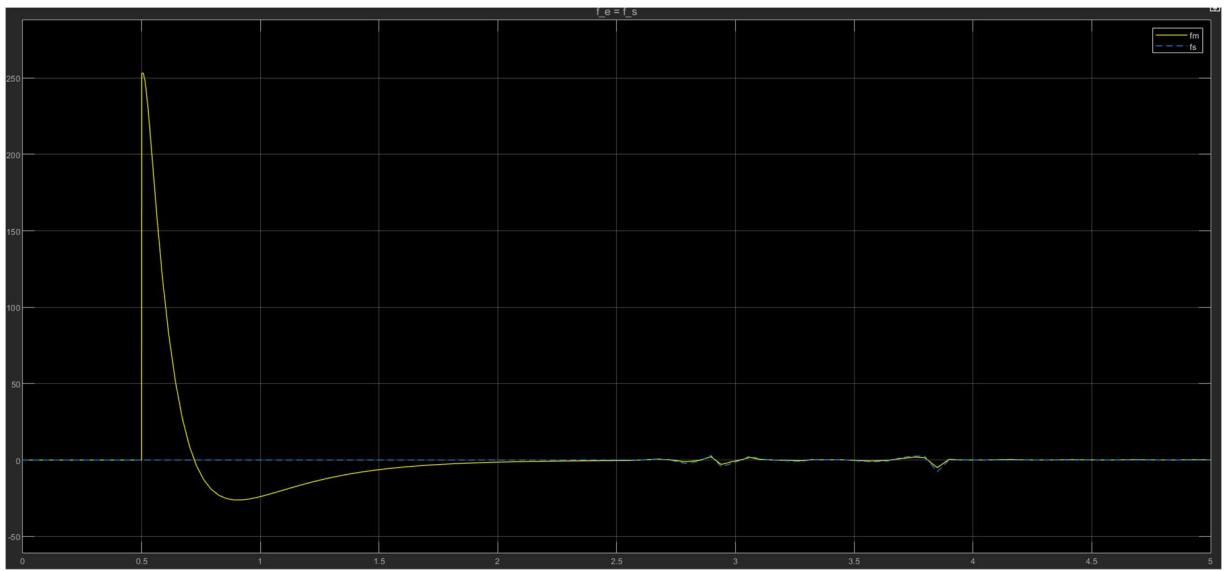


Figure 18 Slave and master robot's force graph for the teleoperation without perfect transparency in free motion (step function).

The environment is designed as the spring and damping model. When the slave robot interact with the environment the slave robot deform the environment by moving to desired position however the rate of deformation is dependent on the stiffness of the environment. However when the environment stops deforming, the slave robot is restricted to reach its desired position. This phenomena can be seen in the Figure 19, where the slave and master robot can not achieve the desired position and the peaks are cut off due to the environment restriction. However the same effect can be seen in velocity graph as well which is shown in the Figure 20. The force of slave robot should be non-zero because of the interaction of slave robot with the environment as shown in the Figure 21.

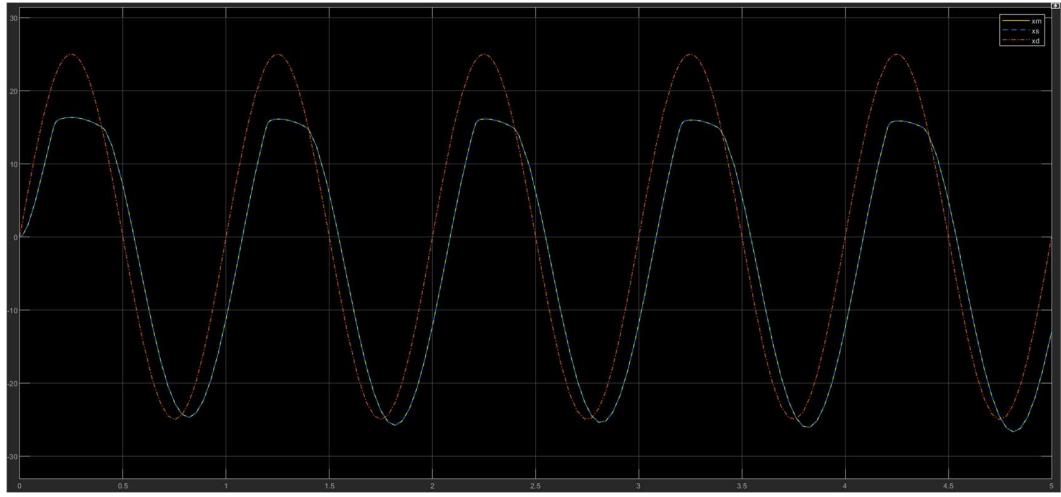


Figure 19 Position graph for the teleoperation with perfect transparency with environment force.

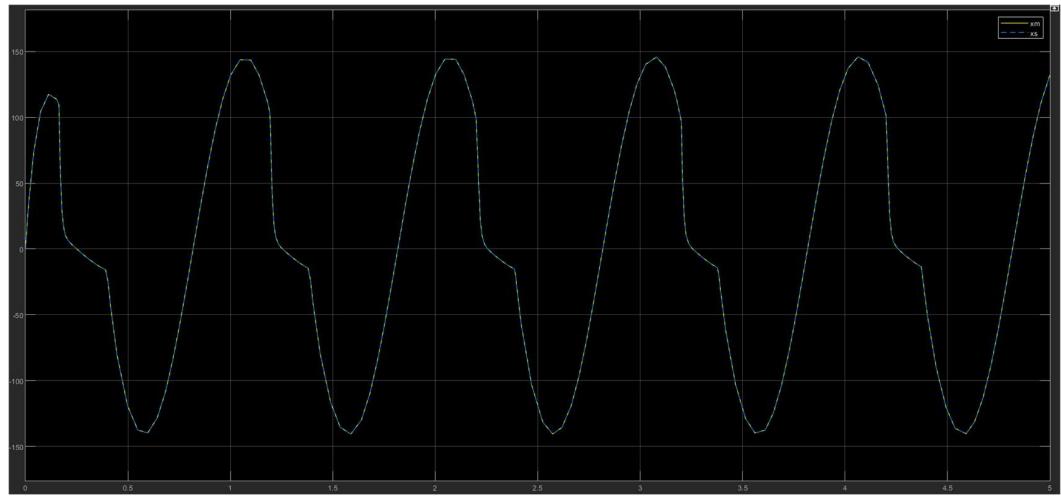


Figure 20 Velocity graph for the teleoperation with perfect transparency with environment force.

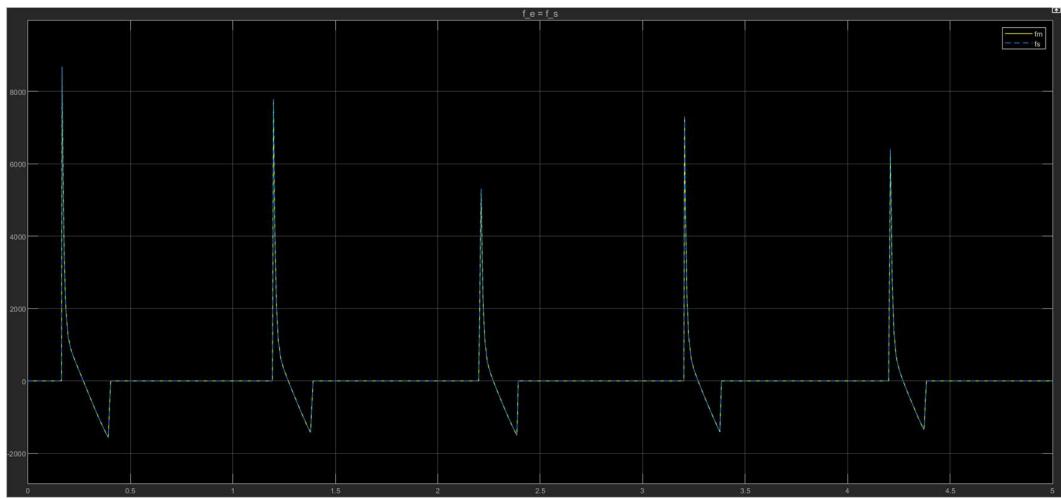


Figure 21 Slave and master robot's force graph for the teleoperation with perfect transparency with environment force.

3. Two Channel teleoperations

As similar to the four-channel teleoperation architecture, the two channel teleoperations uses two communication channels for communication. The one channel is used to transport the signal from the master robot to the slave robot and the other channel is used to transport the signal back from slave to the master robot. The type of signal can be force or position for each channel and depends on the architecture of the teleoperations. It is important to note that the two channel teleoperations do not perform as good as four channel teleoperations due to the limitation in the transportation channels. However, the two channel teleoperation is important because there are some signals that can not be measured accurately due to the limitation of the sensors.

3.1.Two-Channel Position-Position (P-P) teleoperation

The Position-Position two channel teleoperation uses the position only signals for teleoperation between the slave and master robot. As shown in Figure 22, the C_1 and C_4 channels are used to transport the velocity/position signals to and from the master and slave robot respectively.

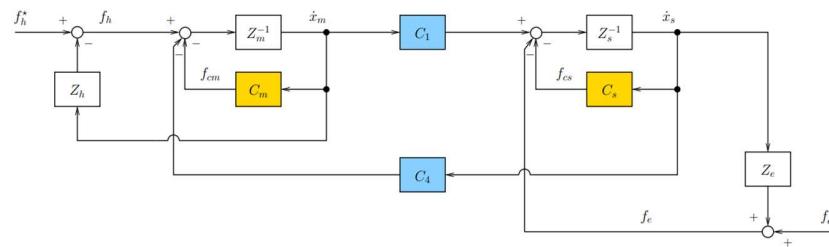


Figure 22 Two-Channel Position-Position (P-P) teleoperation schema

Moreover, Figure 23 shows the implementation of the teleoperation in Simulink. It can be seen that the human intention block is also used to mimic the human force on the master side. Moreover the delay in the transportation channels i.e. C_1 and C_4 can be added by the mean of the manual switches.

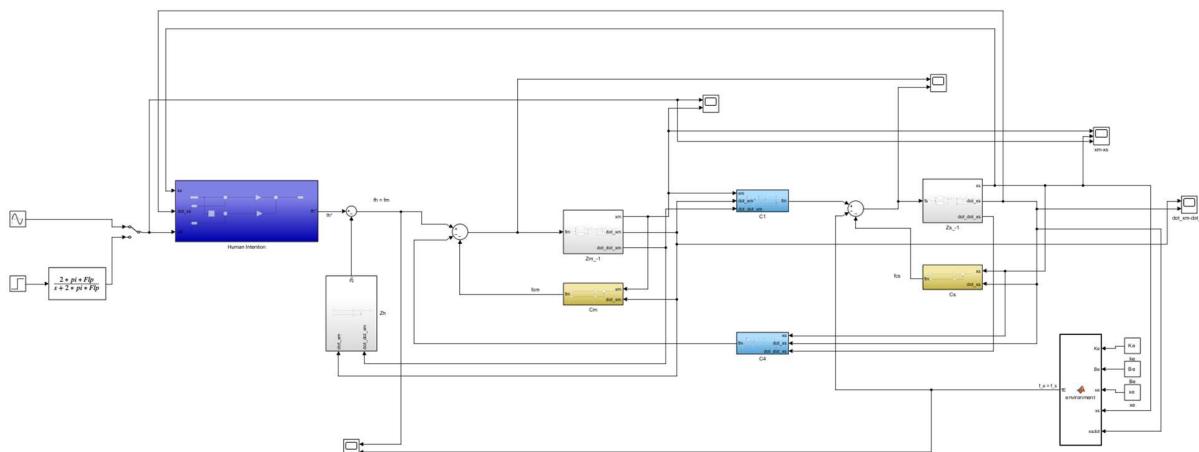


Figure 23 Two-Channel Position-Position (P-P) teleoperation schema in Simulink with human intention.

As the position is transported from the master to slave and slave to master both robots are converged to the positions as shown in Figure 24. Furthermore, the velocities of the slave robot also converge to the master robot as expected. The velocities graph is shown in Figure 25. However the forces of both robots i.e. master, and slave are zero in the free motion as it can be seen in Figure 26.

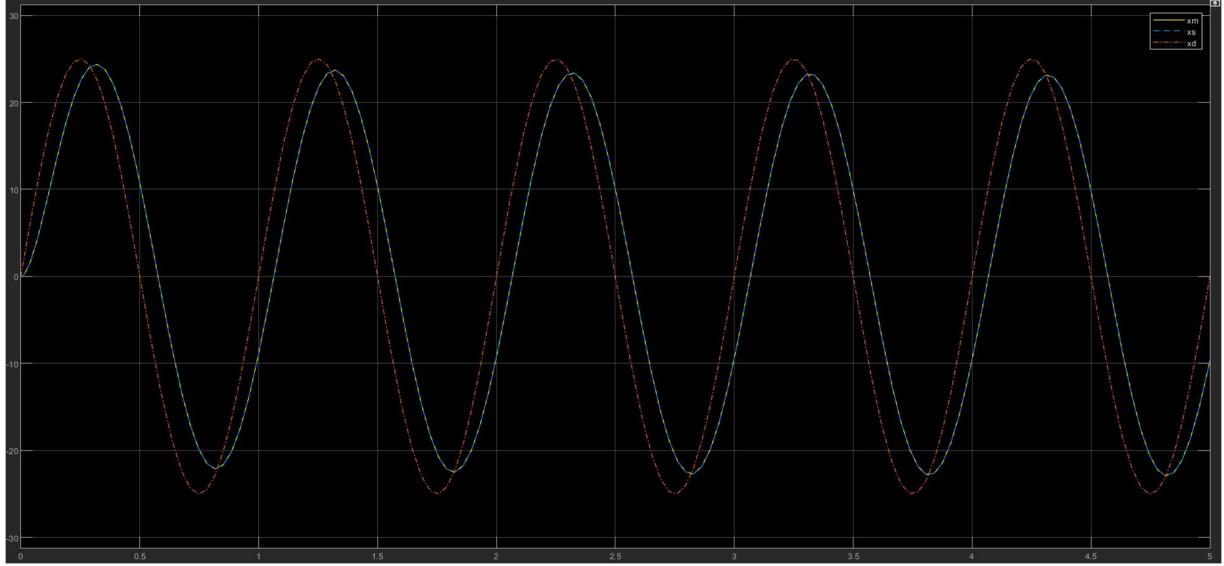


Figure 24 Position graph for slave and master in P-P teleoperation.

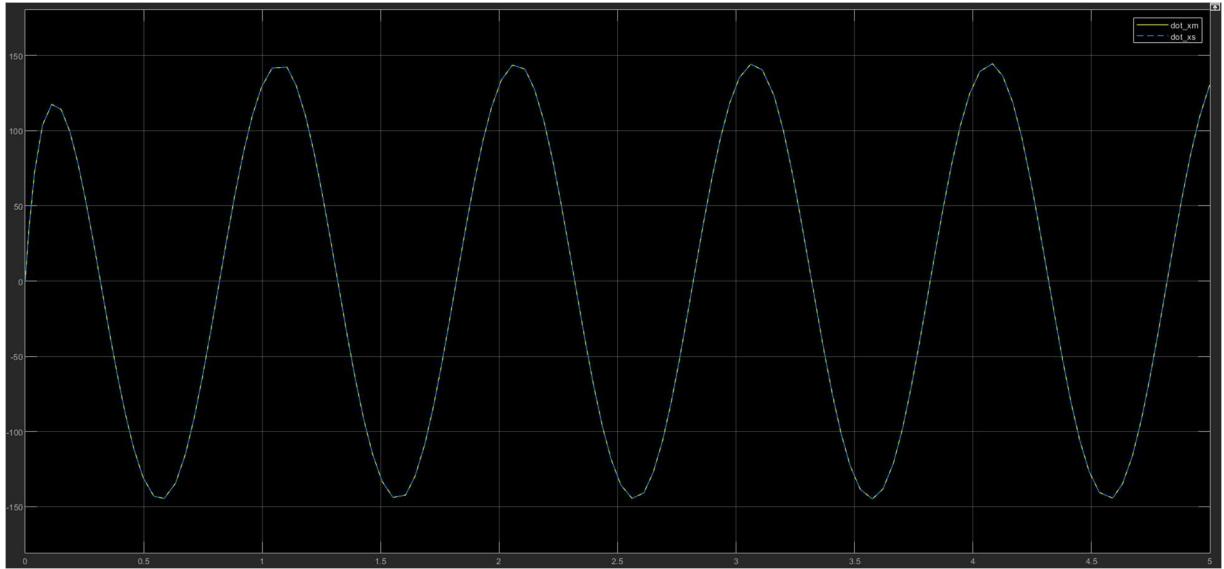


Figure 25 Velocity graph for slave and master in P-P teleoperation.

When the slave robot became in contact with the environment, then the scenario is different than the above explained. The environment kept the slave robot converging to the master robot while the master robot does not have the proper force feedback to restrict its motion as shown in Figure 26. Similar to position, the velocity is not fully converged to the master because of the

same reason explained above at the time when slave interact with the environment. However the Force at the slave side is ridiculously high because of the force loop directed toward the slave controller, the controller try to control the position with help of the feedback force loop and C_s

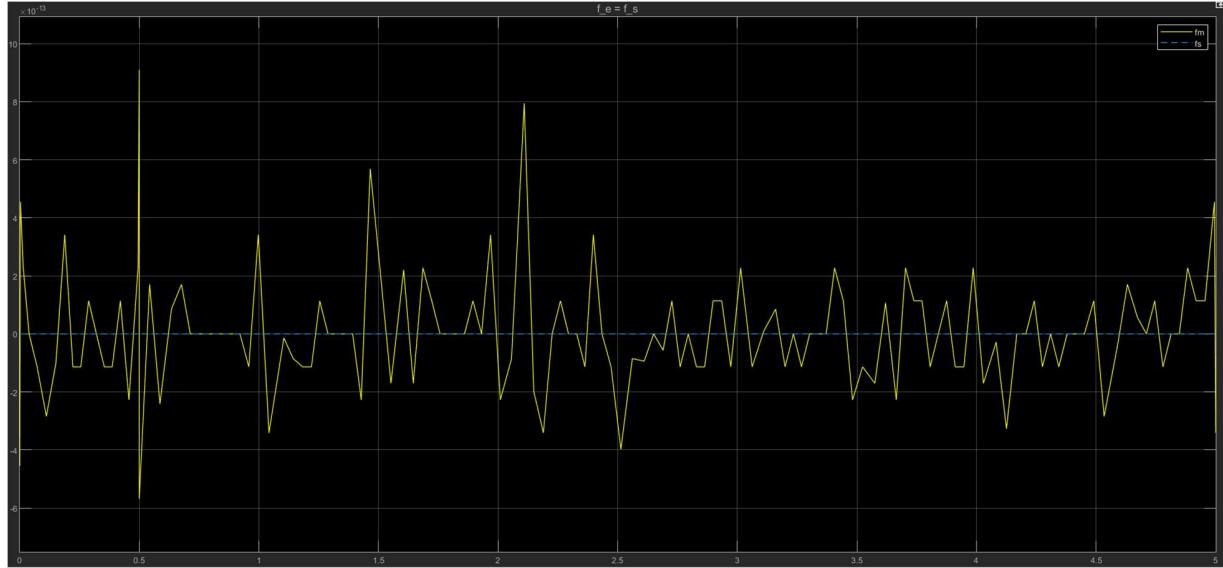


Figure 26 Force graph for slave and master in P-P teleoperation.

controller using the position of the slave. Moreover, the master controller also uses the intention force f_h to control the position of the master robot.

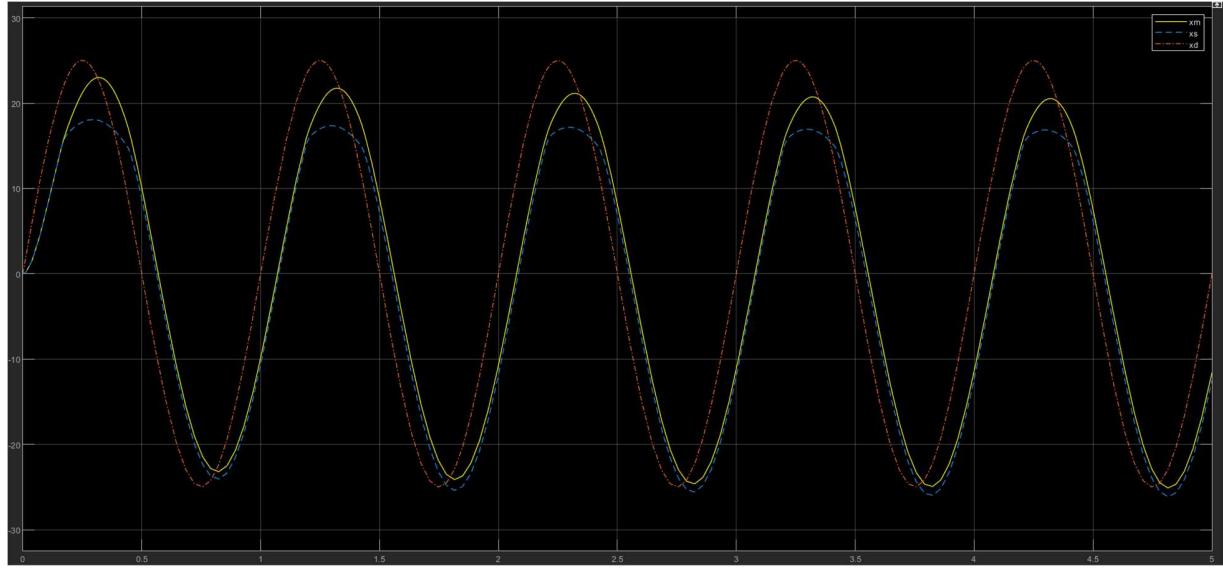


Figure 27 Position graph for slave and master in P-P teleoperation in contact with environment.

In teleoperation, it is very common to have a teleportation delay between the channels of master and slave robot. In Figures 30-32, the delay scenarios are mimicked to address the problems caused by the delays. As it can be seen in the figures, the position, velocities, and forces are not

converged as well as the behavior is random and cannot be described. However, the best guess will be, the slave robot response to the position of the master which is older than the 2 steps of the simulation and the master robot responds to the position of the slave robot which is again steps older. In this case the master and robot could not communicate, and results are not accurate.

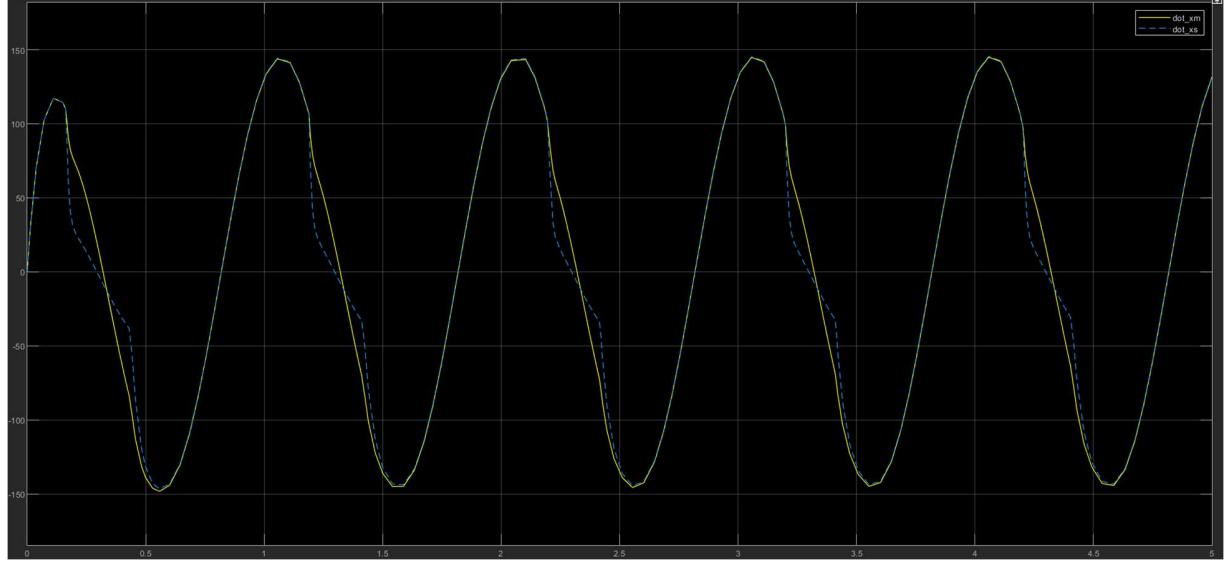


Figure 28 Velocity graph for slave and master in P-P teleoperation in contact with environment.

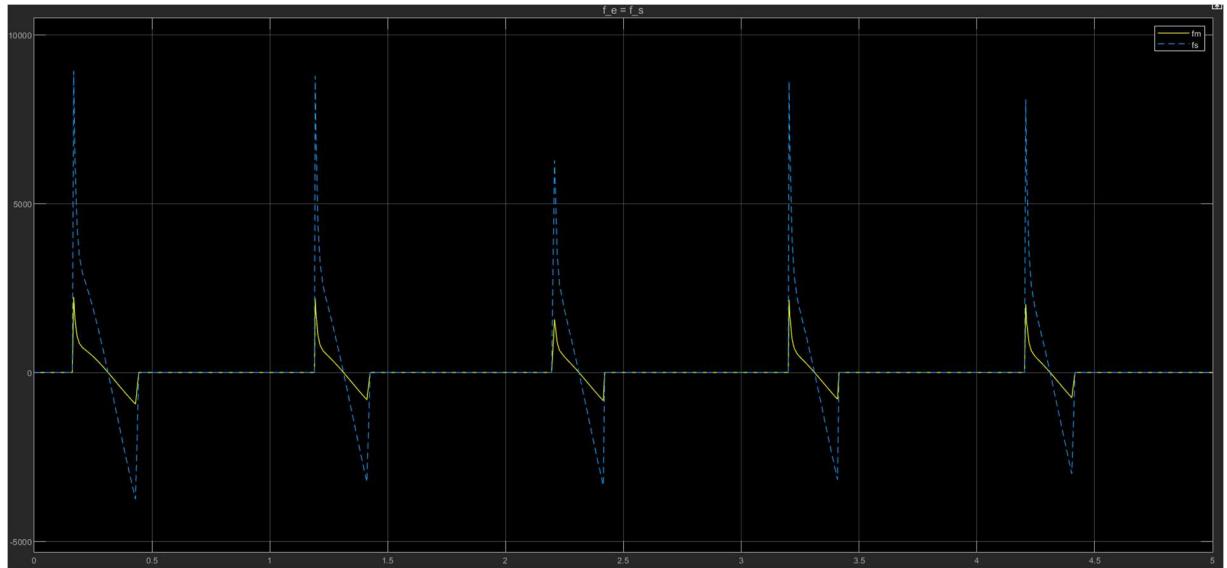


Figure 29 Force graph for slave and master in P-P teleoperation in contact with environment.

3.2.Two-Channel Force-Position (F-P) teleoperation

The Force and Position teleoperation architecture uses the mixture of position and force signal. It transfers the position signal from the master to slave robot and the force signal from the slave to robot signal. The basic schema is shown in Figure 33. According to figure, the channels i.e. C_4

and C_3 is set to zero. The implementation of the schema in Simulink with human intention block is shown in Figure 34.

As expected from the schema, the slave controller should try to converge to the master position due to the position feed-forward while the master position gets the feedback from the slave robot if the slave robot interacts with the environment. This phenomenon is shown in Figures 35-37.

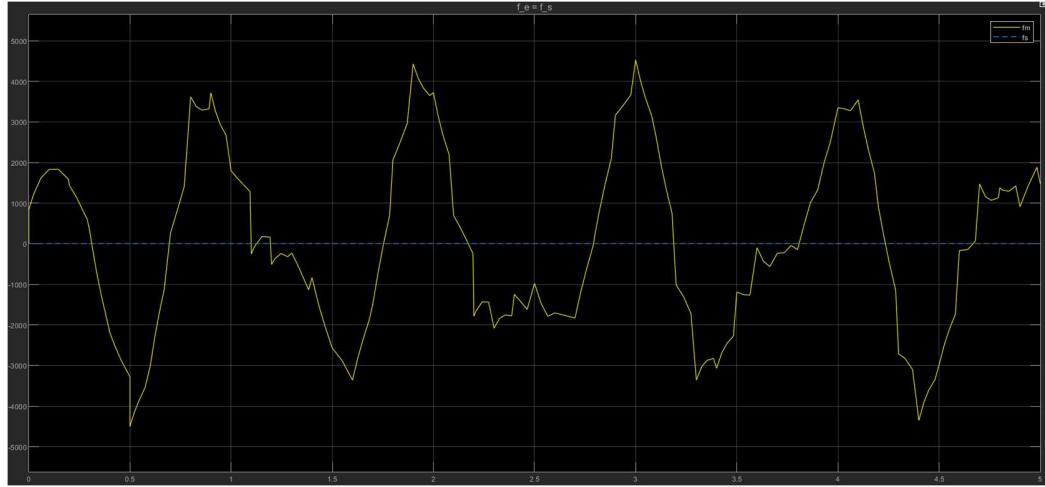


Figure 30 Force graph for slave and master in P-P teleoperation under delay.

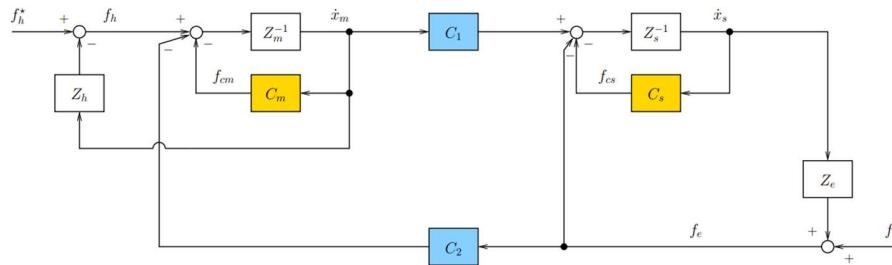


Figure 31 Two Channel Force Position teleoperation schema.

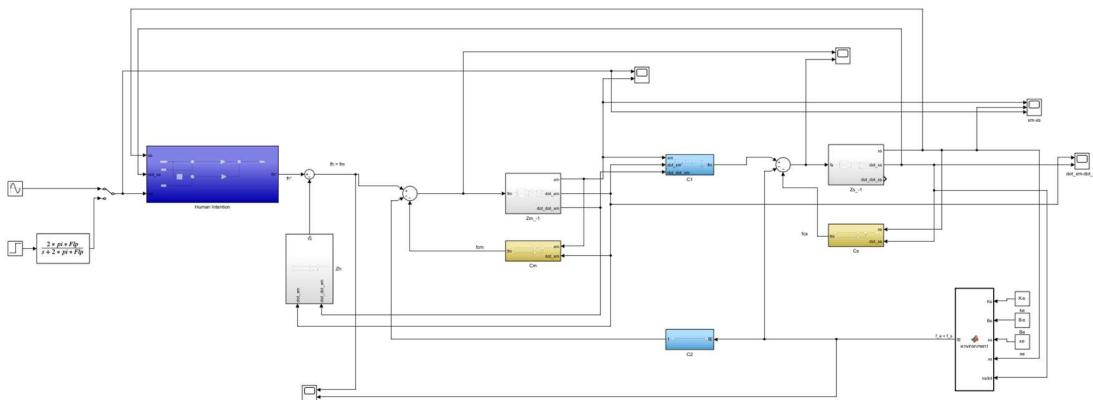


Figure 32 Two Channel Force Position teleoperation schema in Simulink.

According to Figure 35, the position of the master and slave robot graph is produced using the Force-Position controller. As it can be seen that the position of the slave robot converges to the position of master robot under free motion i.e. not interacting with environment. The velocity graph of slave robot also converges to the master velocity as expected shown in Figure 37. Moreover, the slave force is zero as it is not interacting with the environment and the human force is used to maintain the position of the master robot as shown in Figure 38.

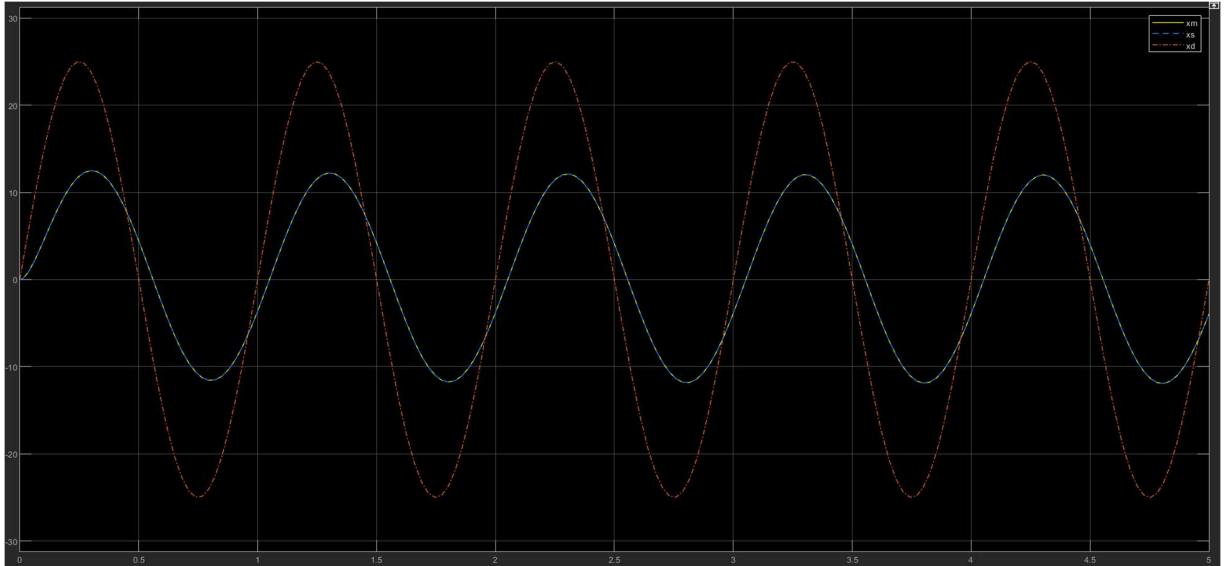


Figure 33 Position graph of master and slave for F-P controller.

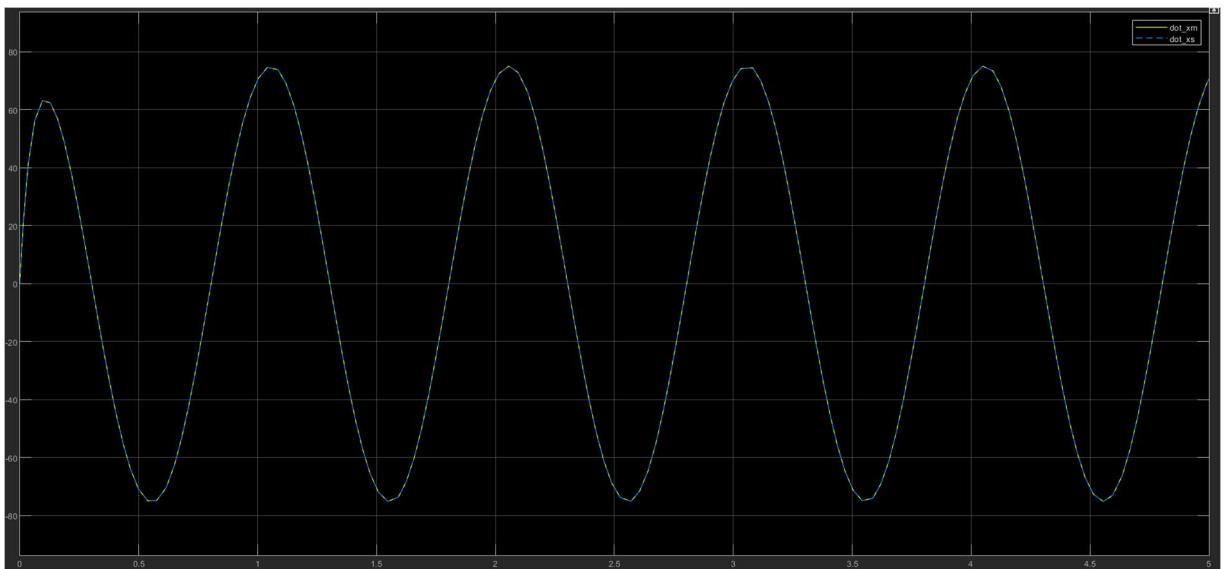


Figure 34 Velocity graph of master and slave for F-P controller.

When the slave robot interacts with the environment, the environment prevents the slave robot from achieving the desired master robot position as can be seen in Figure 36. The environment is modeled as a spring mechanism. Moreover, the velocity also does not converge to master

robot as expected at the time when the robot interacts with the environment which can be seen in Figure 37. When the slave robot interacts with the environment, the feedback force is fed back to the master controller, hence the master robot force act to accumulate the changes in the forces according to Figure 38.

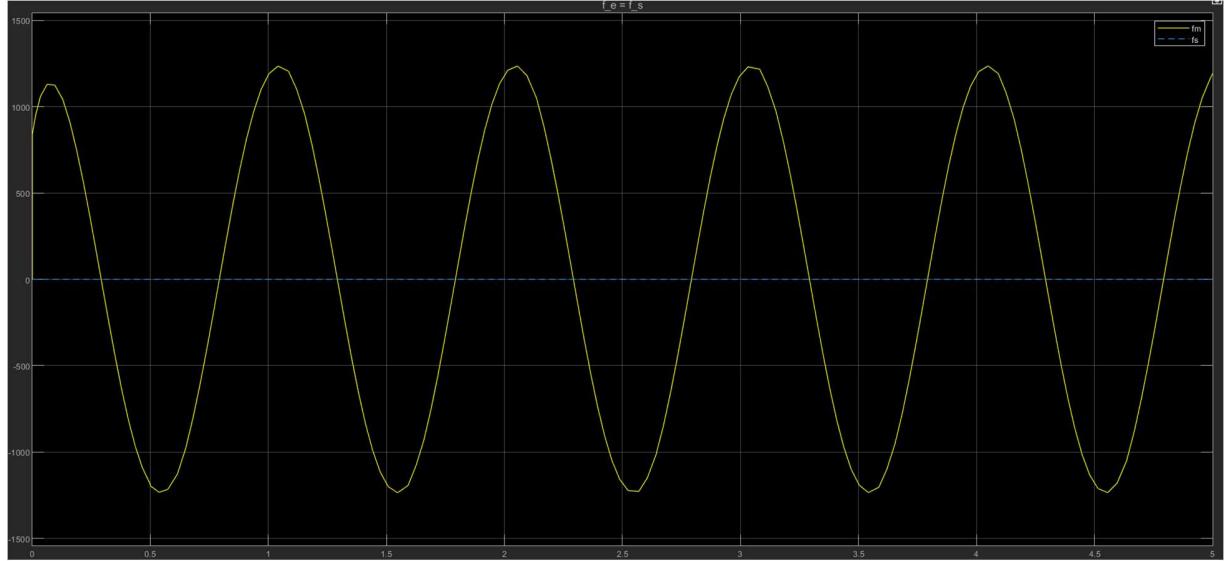


Figure 35 Force graph of master and slave for F-P controller.

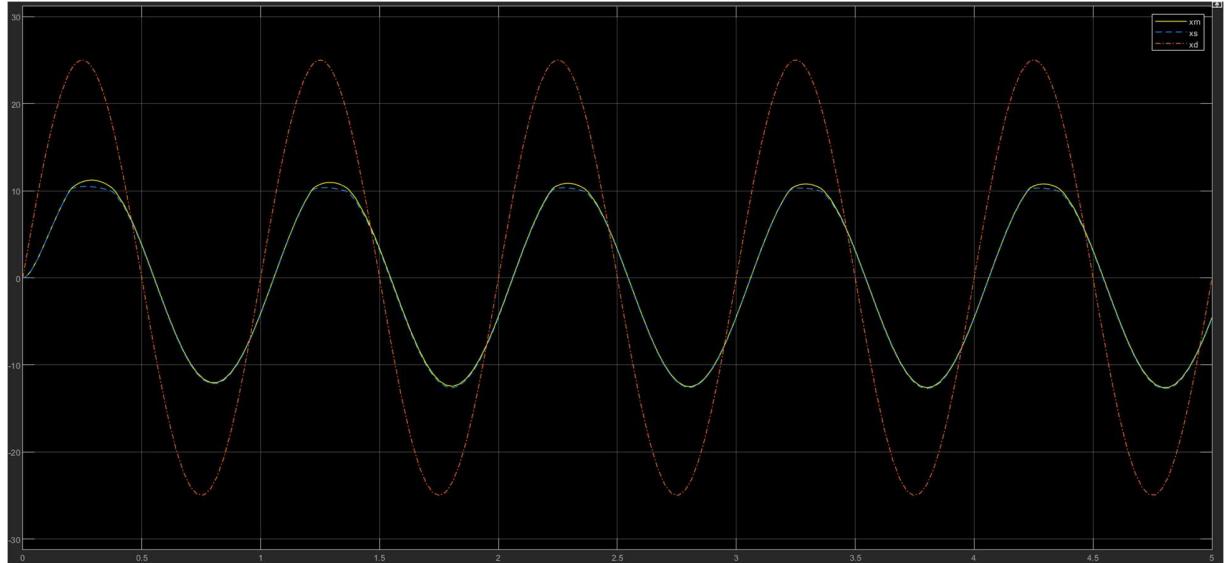


Figure 36 Position graph of master and slave for F-P controller in contact with environment.

As stated earlier, teleoperation can have a delay during the transfer of the force and position to and from master and slave robot. The same scenario is mimicked in Figures 39-40 to shed light on the impact of delay on the position, velocity and forces of the master and slave robot. As from the figures, it can be seen that the behavior of the master and slave robot is totally random because the delay is added of 2 steps between the channels. The slave robot gets the signal after

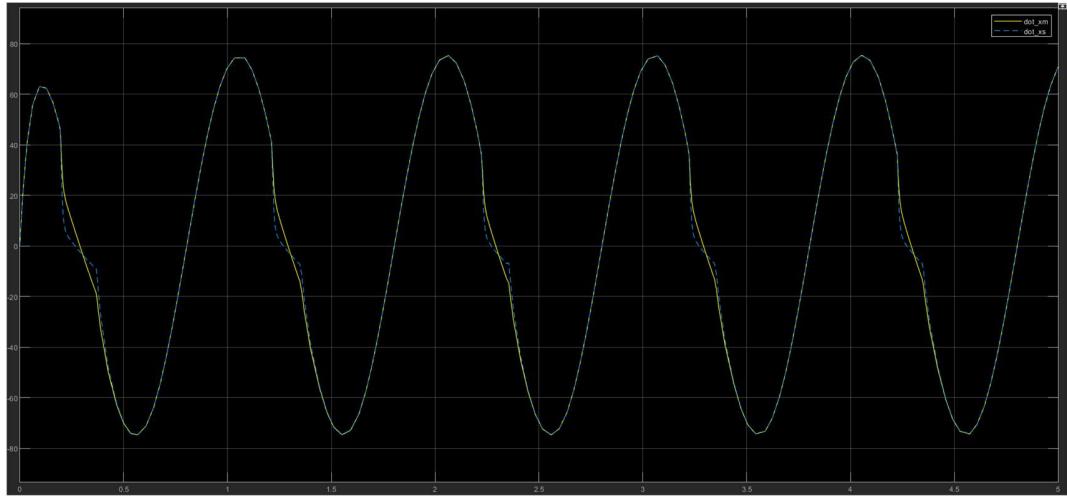


Figure 37 Velocity graph of master and slave for F-P controller in contact with environment.

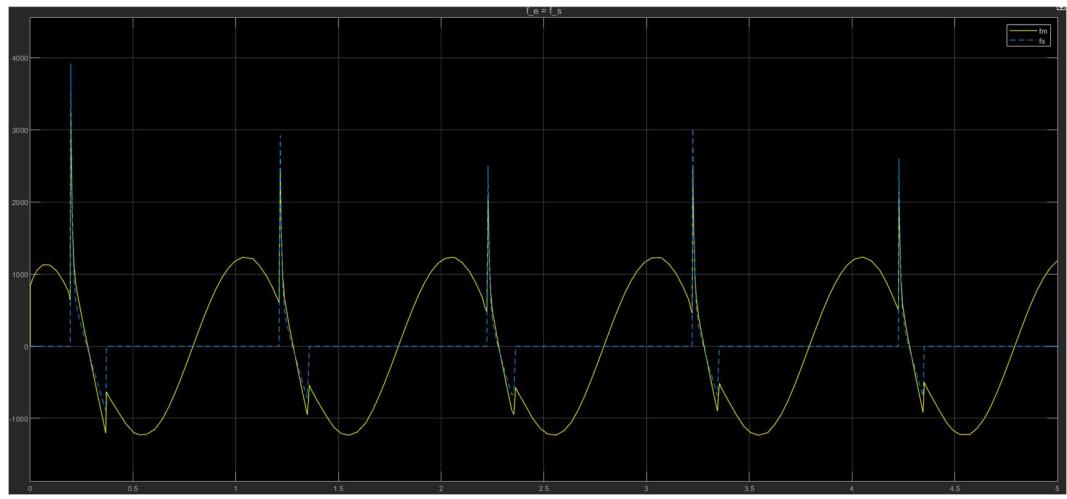


Figure 38 Force graph of master and slave for F-P controller in contact with environment.

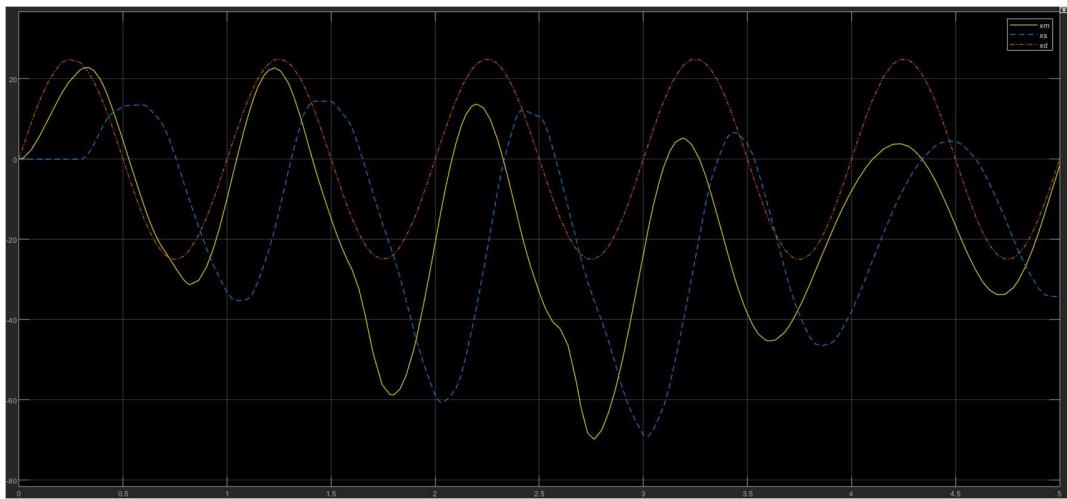


Figure 39 Position graph of master and slave for F-P controller under delay.

2 steps and tries to act upon it while the current signal of the master is something else. Similarly, the force feedback reaches to master slave after 2 steps that is also different from the current force of the slave robot.

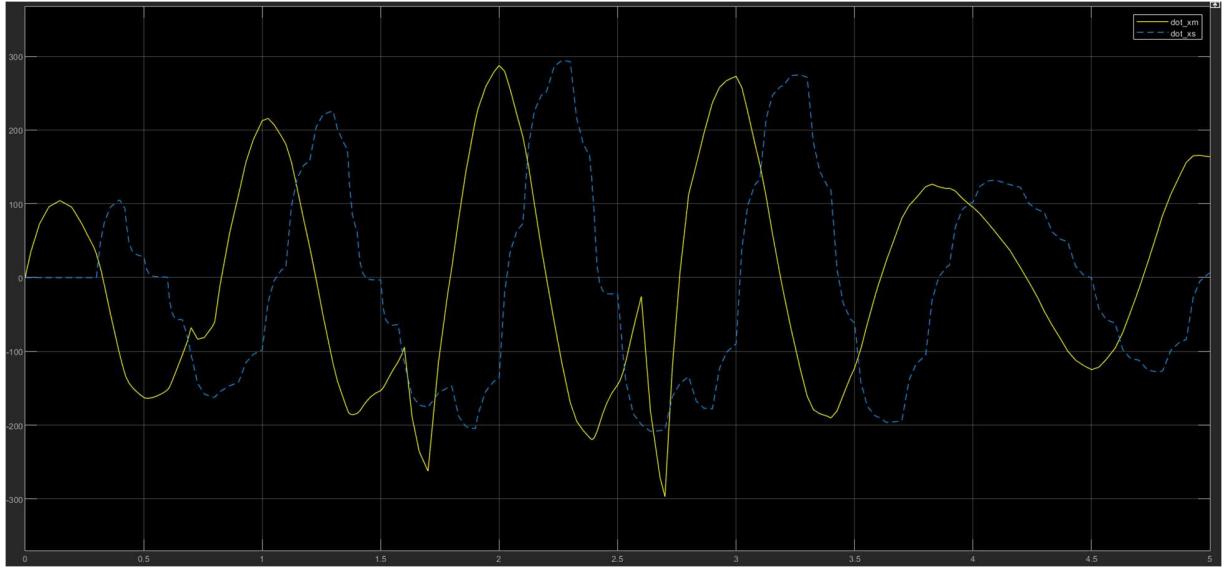


Figure 40 Velocity graph of master and slave for F-P controller under delay.

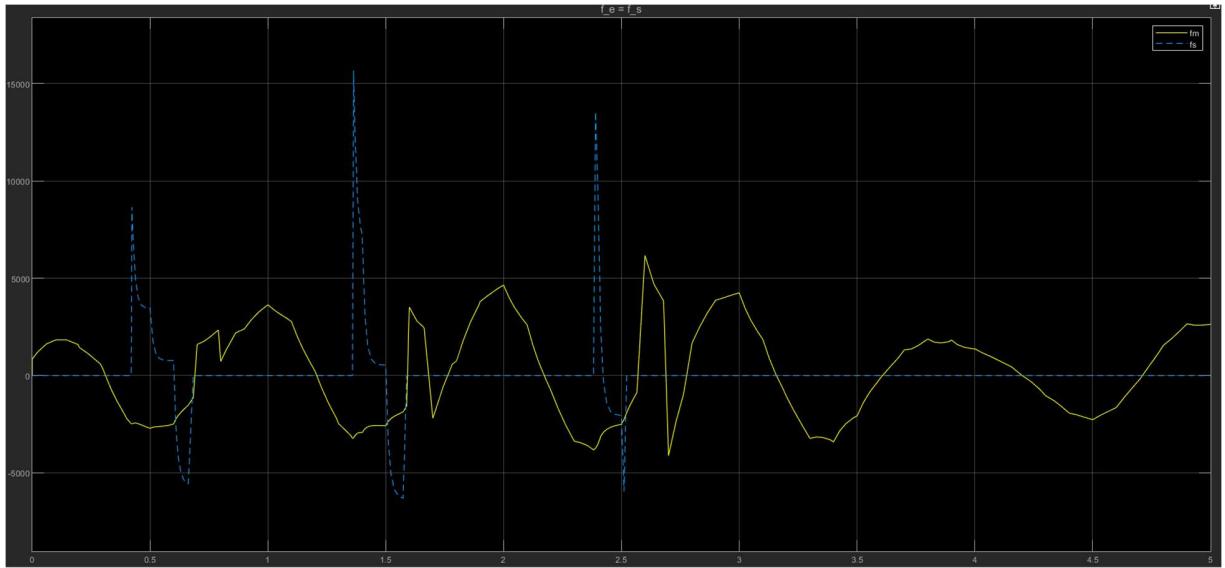


Figure 41 Force graph of master and slave for F-P controller under delay.

3.3.Two-Channel Force-Force (F-F) teleoperation

As name suggests, the Two-Channel Force-Force teleoperation uses the forces to as a transport signal from master to slave and vice versa. The basic schema of teleoperation is shown in Figure 42 and its implementation in Simulink is shown in Figure 43. In this implementation, the C_1 and C_4 block is eliminated and only force channels are used. It is important to note, due to the force

only signal to slave robot, the slave robot tries to converge to the force of the master instead of the position, hence the position of the master robot is not achieved by the slave robot.

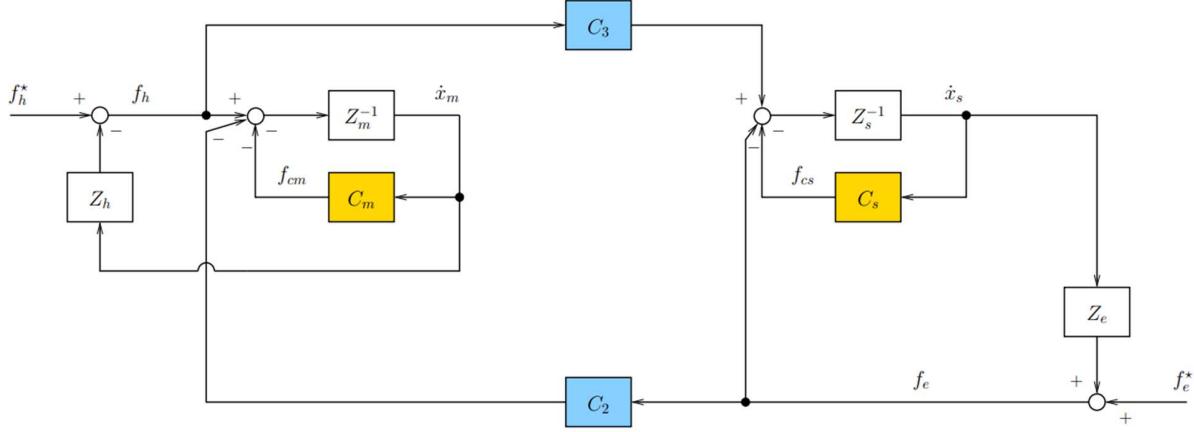


Figure 42 Two-Channel Force-Force teleoperation schema.

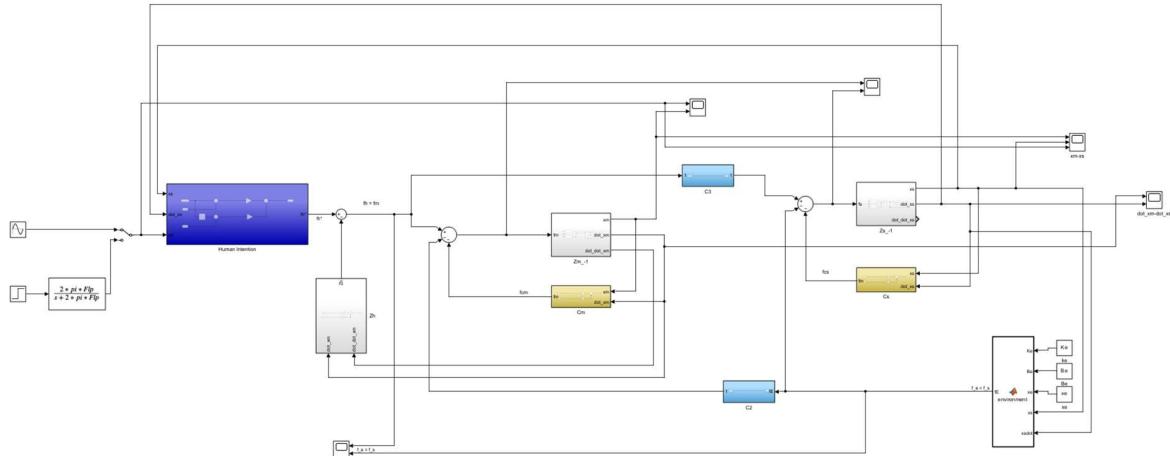


Figure 43 Two-Channel Force-Force teleoperation schema in Simulink.

As stated earlier, due to force signal to the slave robot, the slave robot did not converge to the master robot position as it tries to converge to the force of the master robot. The Figure 44 shows the position of the slave and master robot, as it can be seen that the slave robot did not converge to the master robot as expected. Similar to the position, the velocity of the slave robot did not converge to the master robot shown in Figure 45. Moreover, the Force of the master robot try to act upon the desired position while the slave is in free motion thus do not have any force available. This phenomena can be seen in Figure 46.

When the slave robot interacts with the environment, the environment restricts the motion of the slave robot as it can be seen in Figure 47. The velocity of the slave and master robot is also shown in Figure 48. Furthermore, when the slave robot transfers the force signal to the master

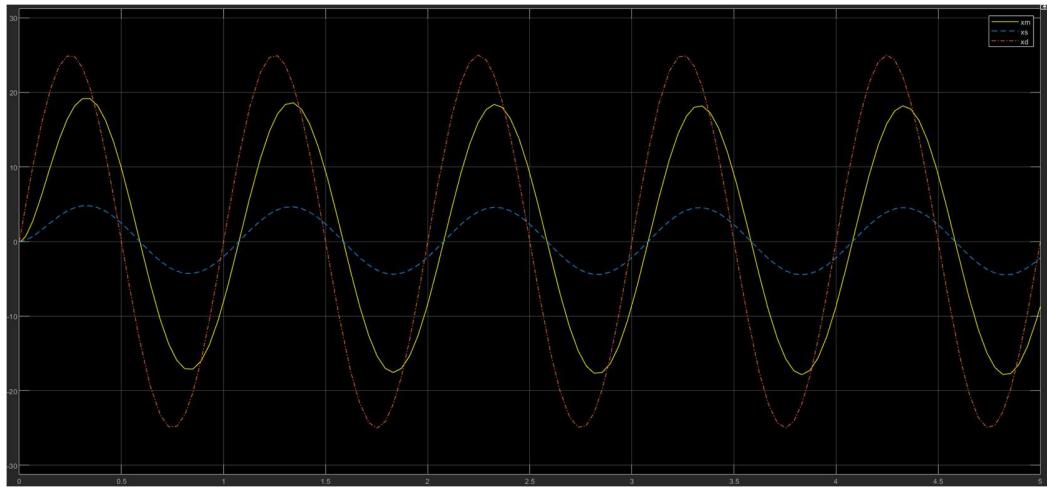


Figure 44 Position graph of master and slave robot for Two-Channel Force-Force teleoperation.

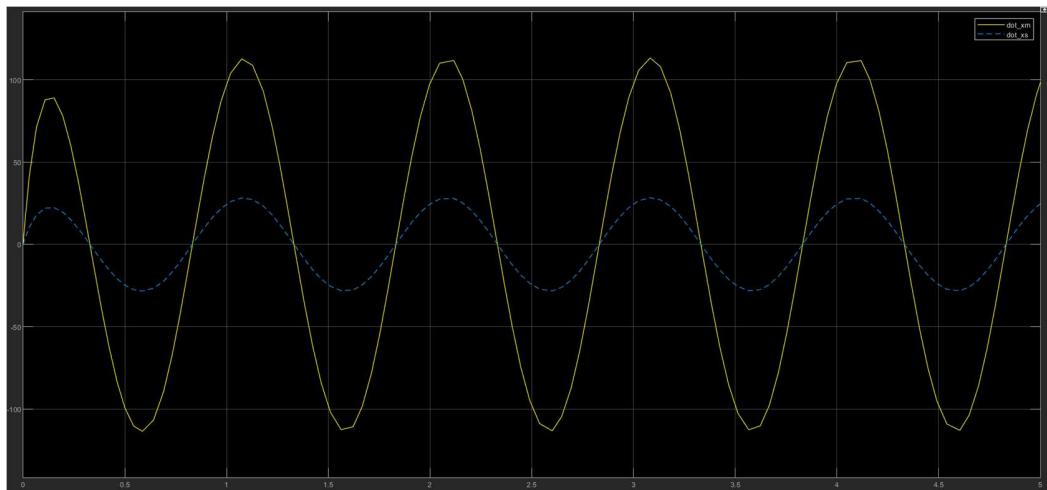


Figure 45 Velocity graph of master and slave robot for Two-Channel Force-Force teleoperation.

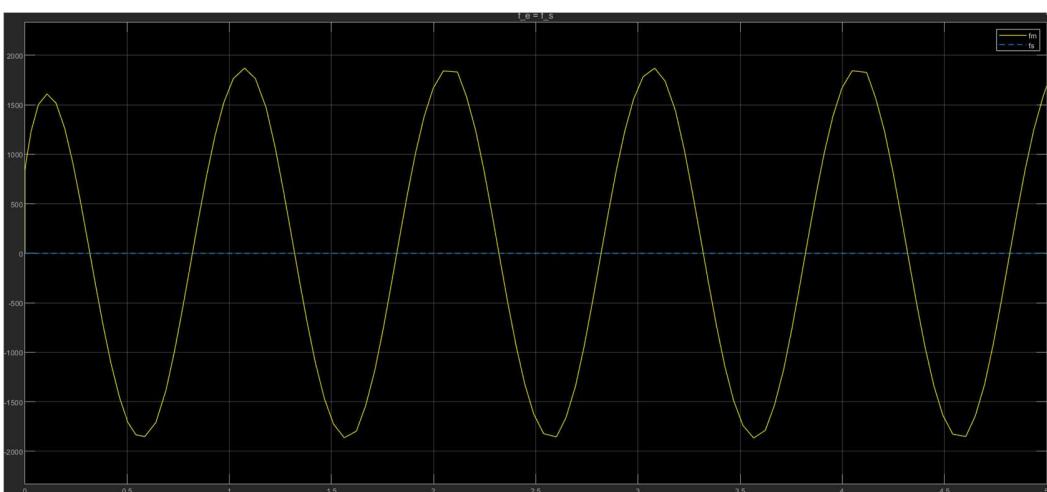


Figure 46 Force graph of master and slave robot for Two-Channel Force-Force teleoperation.

robot, the controller of the master robot responds to the force signal to follow the human intention force as shown in Figure 49.

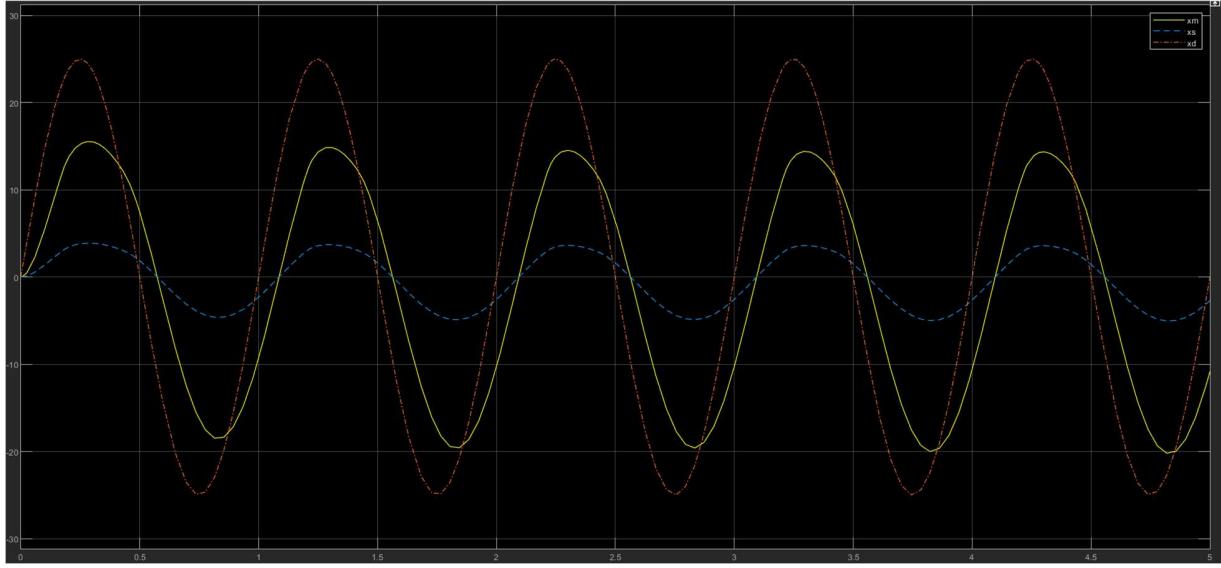


Figure 47 Position graph of master and slave robot for Two-Channel Force-Force teleoperation with environment interaction.

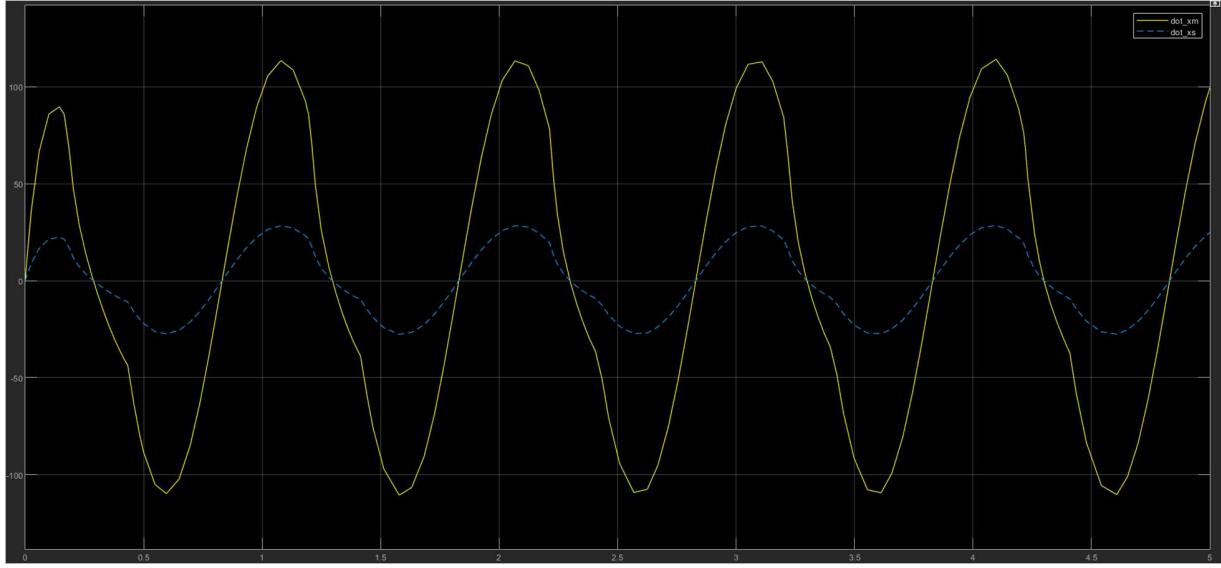


Figure 48 Velocity graph of master and slave robot for Two-Channel Force-Force teleoperation with environment interaction.

3.4.Three-Channel Position-Force and Position (PF-P) teleoperation

The three-channel architecture uses the three signals for teleoperation. These signals include the force and position. The position signal is sent by the master robot to the slave robot while the force and position signal is sent to the master robot by the slave robot during the teleoperation. The basic schema is shown in Figure 50 and the MATLAB Simulink implementation is shown in Figure 51. It is important to note that during the free motion, the slave robot achieves the position of the master robot. However, when the slave robot interacts with the environment, the

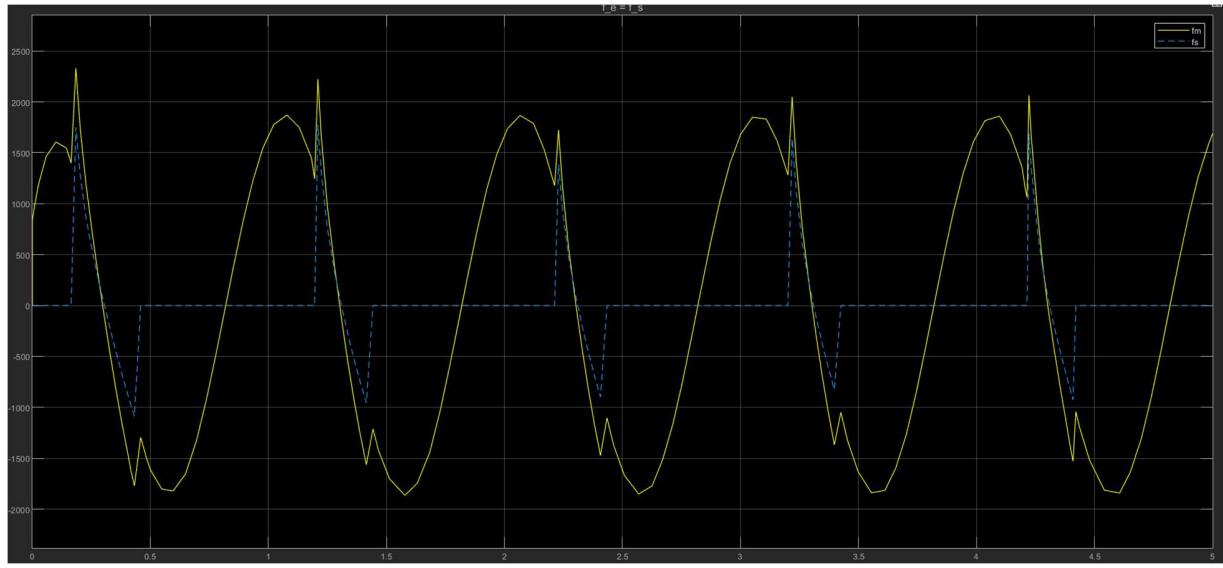


Figure 49 Force graph of master and slave robot for Two-Channel Force-Force teleoperation with environment interaction.

position of the slave robot is restricted by the environment and the master robot tries to achieve the slave position due to the position and force feedback loop of the slave robot.

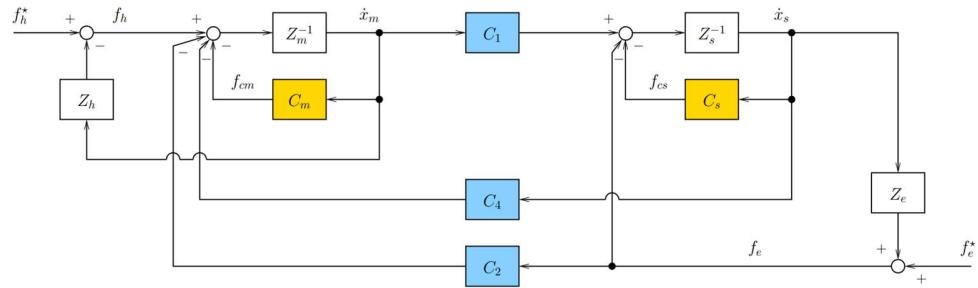


Figure 50 Three-Channel Position-Force and Position teleoperation schema.

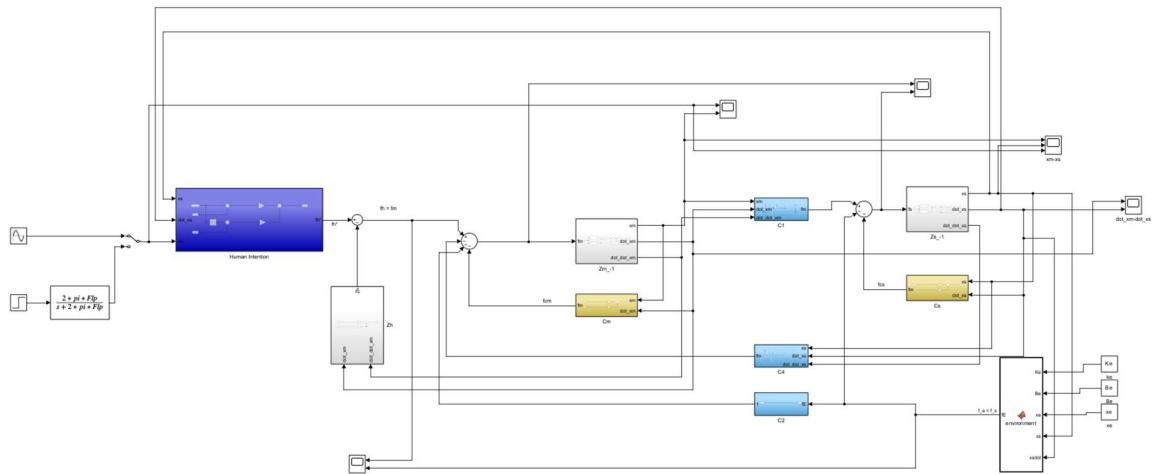


Figure 51 Three-Channel Position-Force and Position teleoperation schema in Simulink.

Figure 52-54 shows the position, velocity, and force graph for the three-channel teleoperation. The slave robot achieves the master robot in free motion as it can be seen in Figure 52. Furthermore, similar to the position, the velocity of the master robot is achieved by the slave robot. However, the slave robot is not interacting with the environment so its force is zero as well as the force at master size is too small and can be considered as zero.

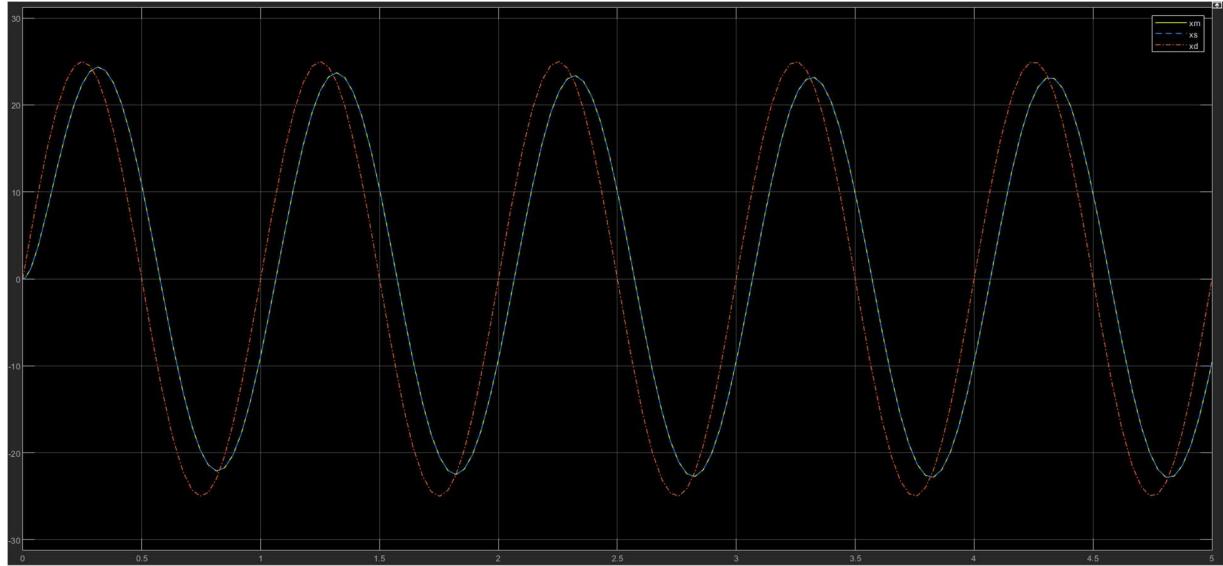


Figure 52 Position graph of master and slave robot for Three-Channel Position-Force and Position teleoperation.

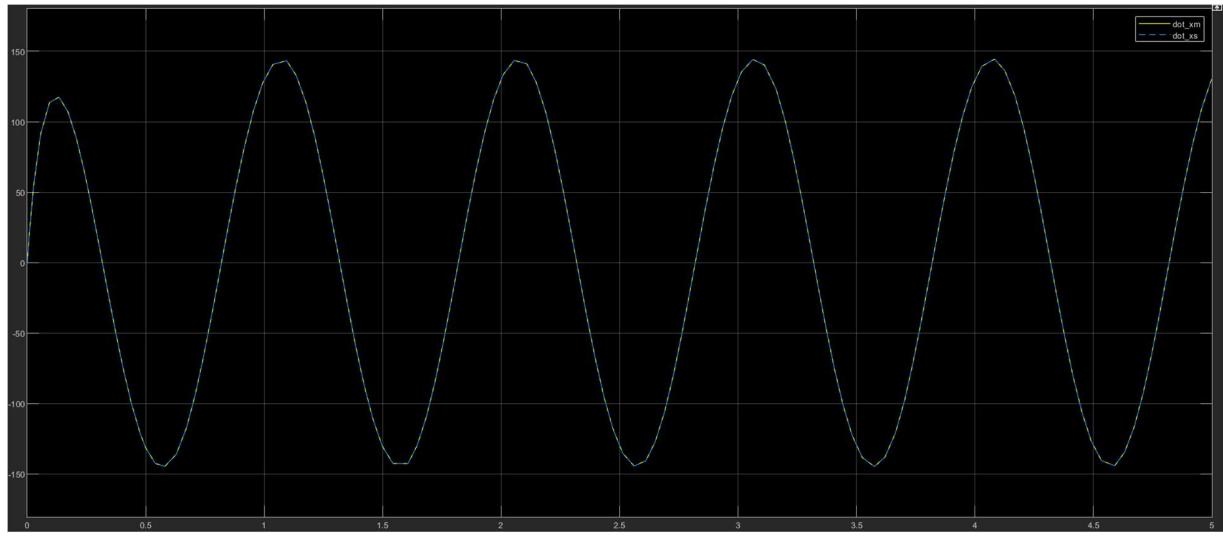


Figure 53 Velocity graph of master and slave robot for Three-Channel Position-Force and Position teleoperation.

When the slave robot starts interacting with the environment, the master robot shows very interesting behavior. The environment restricts the slave to achieve the desired master position while the master position tries to achieve back the slave position, the resultant is very small displacement between the slave and master robot as this is illustrated in the Figure 55. The velocity also has some small difference between the master and slave robot under the time when

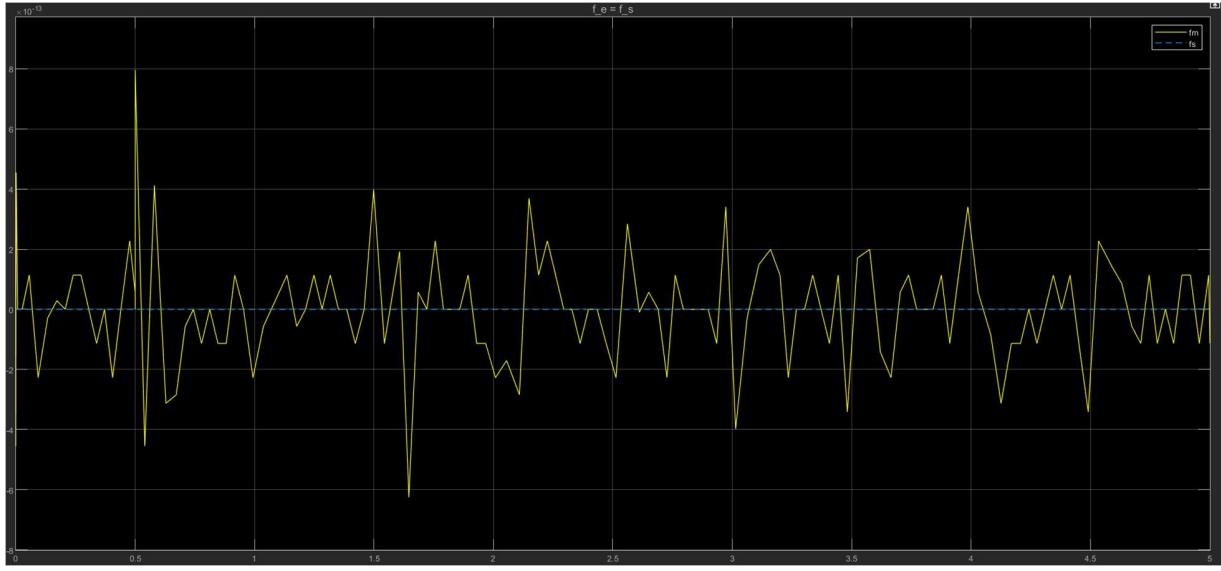


Figure 54 Force graph of master and slave robot for Three-Channel Position-Force and Position teleoperation.

the slave robot interacts with the environment. Figure 56 shows the velocity graph of the slave and master robot. Figure 57 shows the force of slave and master robot. It can be seen that the slave and master robot are almost equal and only small displacement can be seen in the amplitude of the graph.

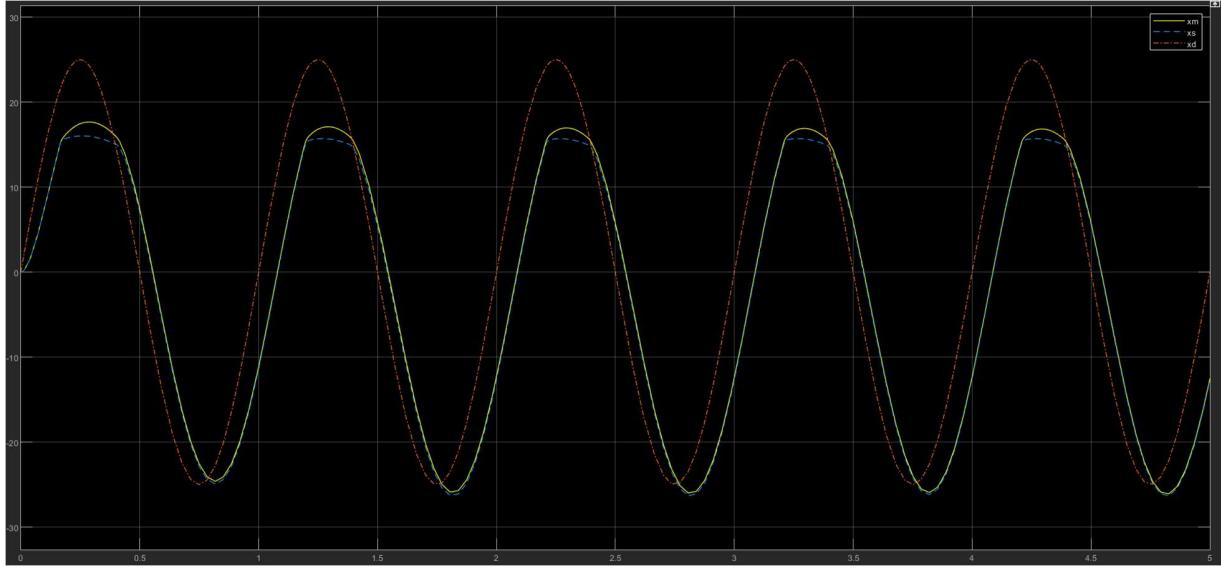


Figure 55 Position graph of master and slave robot for Three-Channel Position-Force and Position teleoperation with environment interaction.

4. Kalman Filter/Predictor

Teleoperation majorly depends on the position of the robot and its velocity. However, the sensor that estimates the position of the robot is not accurate and has a noise in them. These sensors

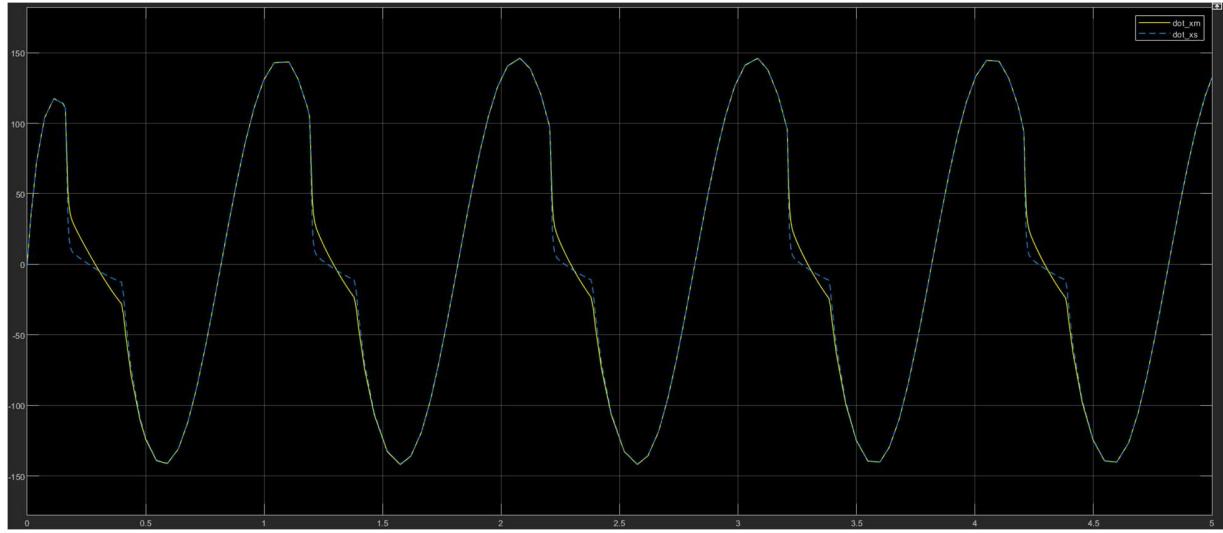


Figure 56 Velocity graph of master and slave robot for Three-Channel Position-Force and Position teleoperation with environment interaction.

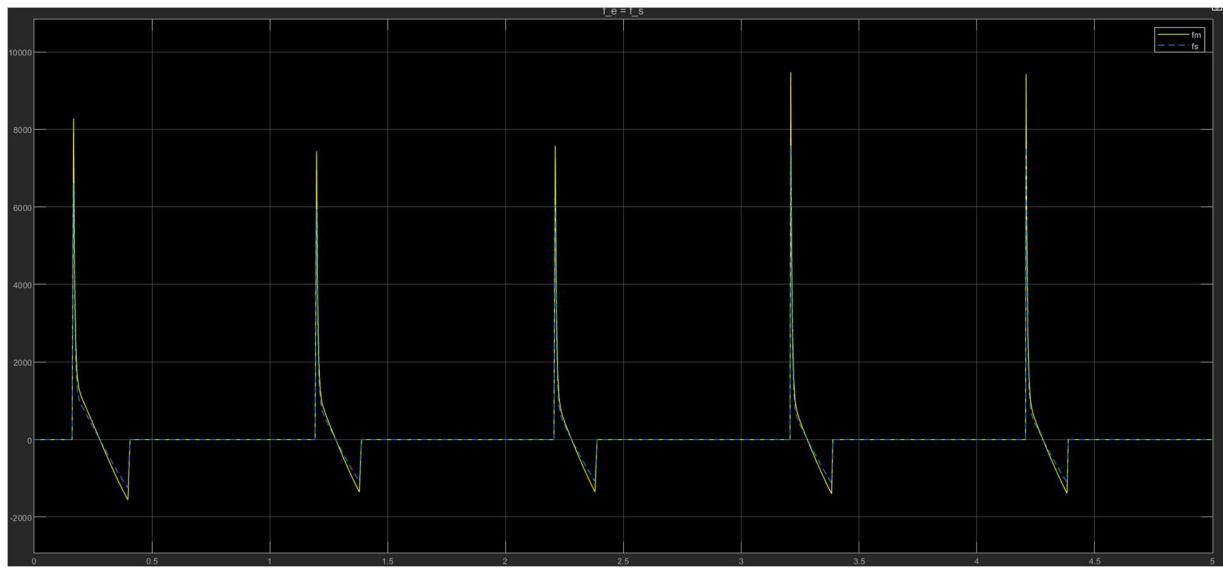


Figure 57 Force graph of master and slave robot for Three-Channel Position-Force and Position teleoperation with environment interaction.

may include GPS, proximity sensors etc. To model this noise, the discrete time system is assumed with the additive noise. This model is illustrated from following equation

$$y(t) = s(t) + n(t) \quad (18)$$

where $y(t)$ is the measurement of the position at the particular time t , $s(t)$ is the actual position and $n(t)$ is the additive measurement noise by the filer. The Kalman filter/predictor can be used to denoise the measurement as much as possible.

Kalman filter is a mathematical algorithm that uses a series of measurements observed over time, containing statistical additive noise, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone. The Kalman filter is an optimal estimator that minimizes the expected value of the squared error between the estimated values and the true values. It does this by combining a predicted state estimate with a new measurement using a weighted average. The weights are calculated based on the error covariance matrices of the predicted state estimate and the measurement.

It is important to note that, Kalman filter and Kalman predictor are almost the same concept, In Kalman filter the previous data is filtered while in the Kalman predictor by using the one previous time the next data is predicted. The difference between the results is almost neglectable.

4.1. Kalman Filter

As stated earlier the Kalman filter estimates the current value $\hat{x}_{k+1|k+1}$ using the previous estimates $\hat{x}_{k|k}$ and the current measurement y_{k+1} . The Kalman filter is composed of three primary equations which states that

$$\hat{x}_{k+1|k+1} = A\hat{x}_{k|k} + K_{k+1}(y_{k+1} - CA\hat{x}_{k|k}) \quad (19)$$

$$\hat{P}_{k+1|k} = AP_{k|k-1}A^T - AP_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1}A^T + Q \quad (20)$$

$$K_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1} \quad (21)$$

where A is the discrete time approximation, converting the continuous time system to the discrete time system. K_{k+1} is known as the gain matrix, which maps the estimation error into the correction of prediction state. Equation 20 is the Riccati equation which helps to compute the optimal gain for the estimation of the variable by using the covariance matrix of the state variables.

4.2. Kalman Predictor

On the other hand, the Kalman predictor uses the previous state variable $\hat{x}_{k|k-1}$ and current measurement y_k to estimate (predict) the future variable $\hat{x}_{k+1|k}$. Similar to Kalman Filter, the predictor has almost same equation which is illustrated below

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \quad (22)$$

$$\hat{P}_{k+1|k} = AP_{k|k-1}A^T - AP_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1}A^T + Q \quad (23)$$

$$K_k = AP_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1} \quad (24)$$

Similar to Kalman Filter, Equation 20 is the Riccati equation which helps to compute the optimal gain for the estimation of the variable by using the covariance matrix of the state variables. In other words, it is a mathematical equation that helps to calculate the optimal gain for estimating a variable by using the covariance matrix of the state variables. K_k is known as the gain matrix

which maps the estimation error into the correction of prediction state. A is the discrete time approximation that converts the continuous time system to the discrete time system.

As in the experiment, both the Kalman predictor and estimator is used to predict the velocity and acceleration using the noisy position. However, for this experiment following matrixes and vectors are defined

$$A = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{T_s^3}{6} \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix}$$

Where T_s is the discrete time of the measurement interval. In other words, it is a time elapsed until the next measurement is obtained. In this experiment, the $T_s = 0.001s$ which means ever defined second the measurement is obtained. B matrix is the control input matrix that maps the control input $u(t)$ to the state space. It is used when there is a control input that affects the state of the system.

$$C = [1 \ 0 \ 0]$$

$$Q = qBB^T$$

However, the q and R is the constant which is set to 1000 and 0.001 respectively. Figure 58 shows the velocity prediction and filtering using the noisy measurement of the position. The graphs shows the actual velocity with the predicted and filtered velocity. It can be clearly see that the prediction and filtering both have a similar results and have very less error with respect to actual velocity. It is important to note that during the changing of the intensity of the velocity i.e. sharp changes the error occurs due to the change in the variance between the state variable.

Similar to the velocity, the acceleration graph shown in Figure 59. In this graph, the actual acceleration is compared with the predicted and filtered acceleration by the Kalman mathematical algorithm. It can be observed that the acceleration is map to the actual acceleration but has error greater than the velocity on the sharp changes. This is due to the very sharp fall of the acceleration at the particular time.

It is very hard to decide the better results of the Kalman predictor and filter by simply looking at the comparative graph. However, the Root Mean Square Error (RMSE) is used to optimize the solution as much as possible. The RMSE stated as

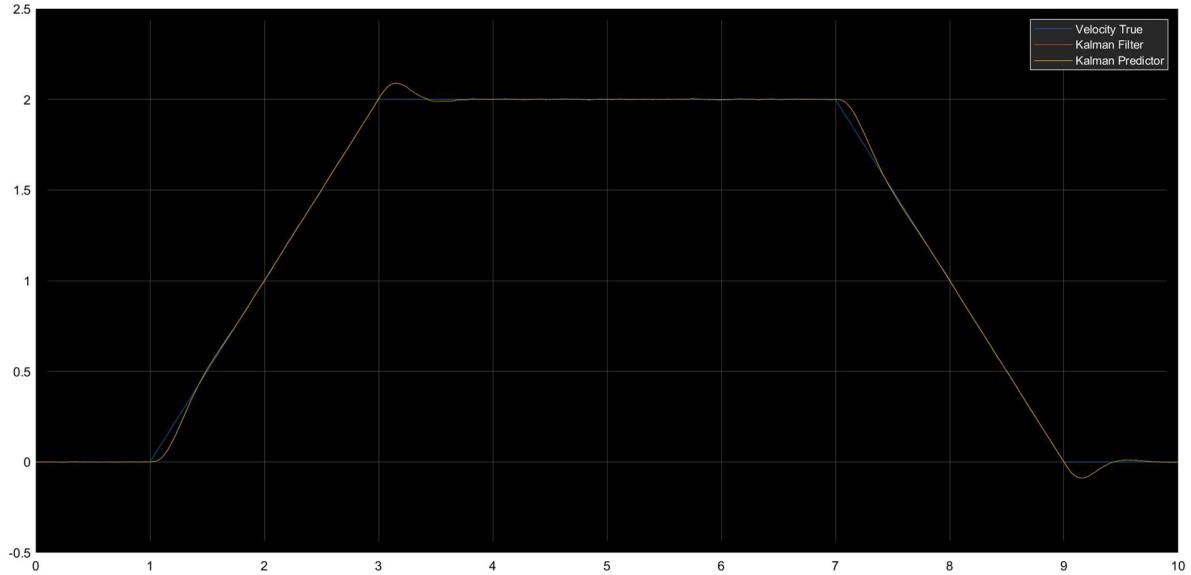


Figure 58 Velocity prediction and filtering using Kalman algorithm.

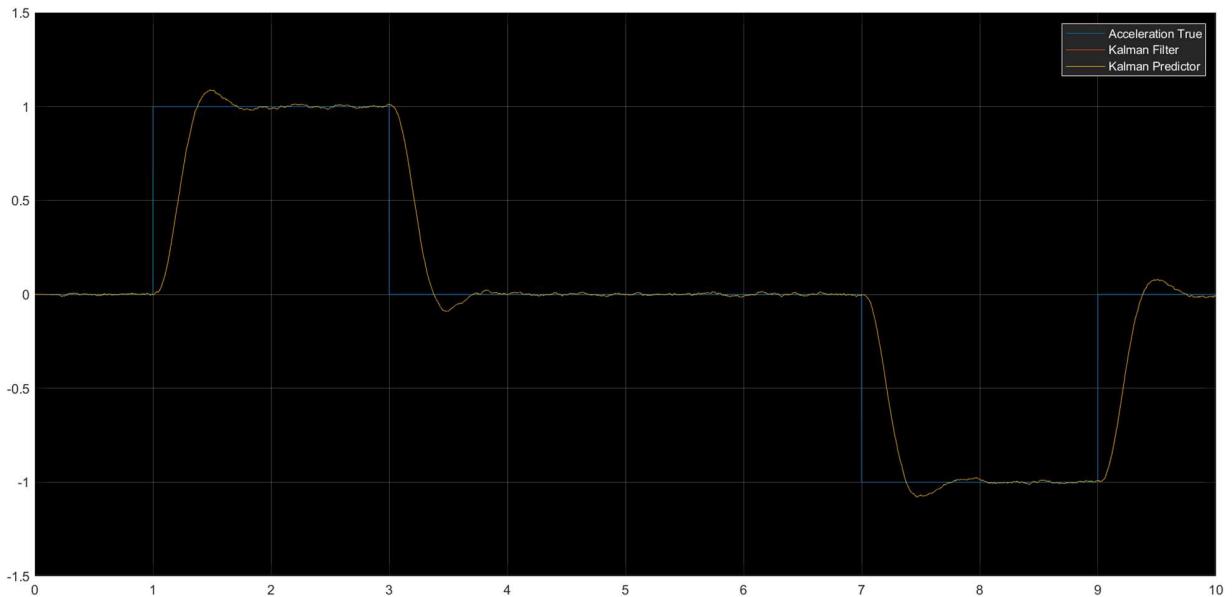


Figure 59 Acceleration prediction and filtering using Kalman algorithm.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (25)$$

Where the n is the total number of observations, \hat{y}_i is the predicted or filtered i_{th} variable and y_i is the actual i_{th} variable. The RMSE of the velocity and acceleration is shown in the following table.

Table 1 Root Mean Square Error of Kalman Filter and Kalman Predictor for Velocity and Acceleration. (Lower is better)

	Velocity	Acceleration
Kalman Filter	0.0244	0.2584
Kalman Predictor	0.0245	0.2584

From Table 1, it can be seen that both Kalman predictor and filter produce almost the same results. AS stated earlier, the acceleration has much larger RMSE then the velocity.

4.3. Kalman Steady-State

The Kalman steady-state assumes the Kalman gain and Riccati equation as a Linear Time Invariant System (LTI). Hence the Riccati and the Kalman gain became time invariant and constant. For this, these two equations do not need to be calculated in every iteration. However to convert to steady space for both the predictor and filter, the Equation 20 and 23 for Riccati equation needed to be changed to

$$P_{\infty} = AP_{\infty}A^T - AP_{\infty}C^T(CP_{\infty}C^T + R)^{-1}CP_{\infty}A^T + Q \quad (26)$$

and for the Kalman gain, the Equation 21 and 24 needed to be modified to work with the particular version of the Kalman i.e. filter and predictor. The Kalman gain looks like

$$K_{\infty} = \begin{cases} AP_{\infty}C^T(CP_{\infty}C^T + R)^{-1}, & \text{Predictor} \\ P_{\infty}C^T(CP_{\infty}C^T + R)^{-1}, & \text{Filter} \end{cases} \quad (27)$$

where infinity shows the LTI of both the gain and Riccati equation. Figures 60 and 61 show the velocity and acceleration graph for the steady-state Kalman filter. According to the graphs, both graphs i.e. steady-state Kalman and time variant Kalman is almost same, but there is a little difference can be seen in the start of the acceleration graph. Steady-state shows a little large error in the start then in the time variant Kalman. This is due to the constant Riccati equation and Kalman gain do not manage to change in the beginning of the filter and predictor of steady-state model.

Table 2 Root Mean Square Error of Kalman Filter and Kalman Predictor in Steady-State for Velocity and Acceleration. (Lower is better)

	Velocity	Acceleration
Kalman Filter	0.0244	0.2584
Kalman Predictor	0.0245	0.2584

However, according to Table 2, the RMSE are the same of both standard and steady-state Kalman filters and predictor. As it can be seen from Figure 61, the acceleration error at the start is much greater than the start of the acceleration in standard Kalman in Figure 59. Despite this behavior, the error of both variants is same due to the very little error and hence due to the number precision this behavior cannot be seen.

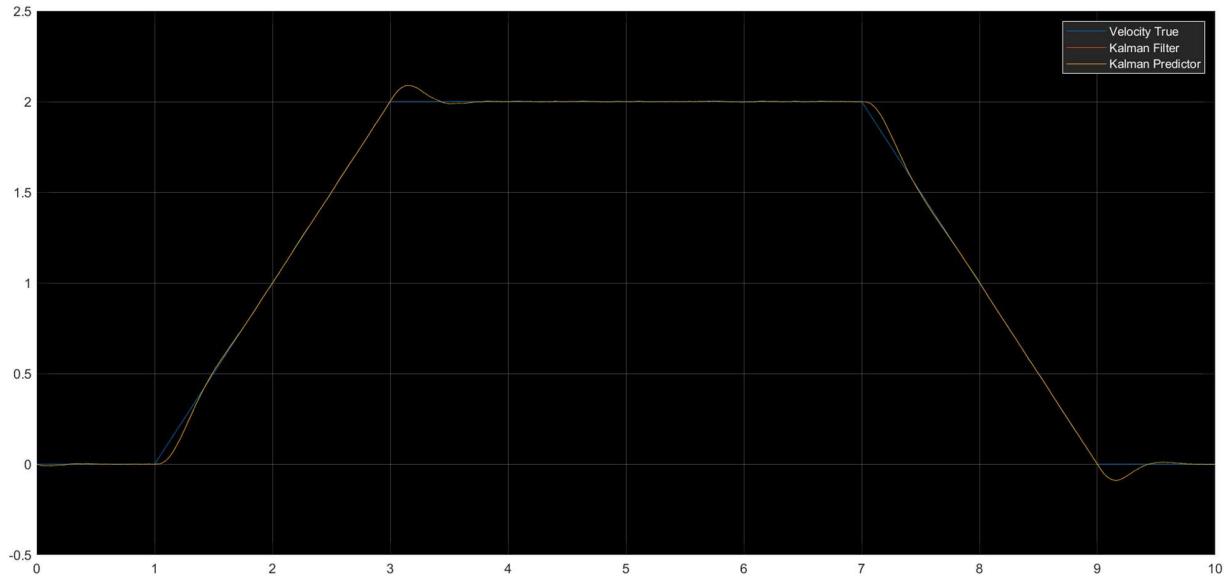


Figure 60 Velocity graph of Kalman Filter and Predictor for Steady-State.

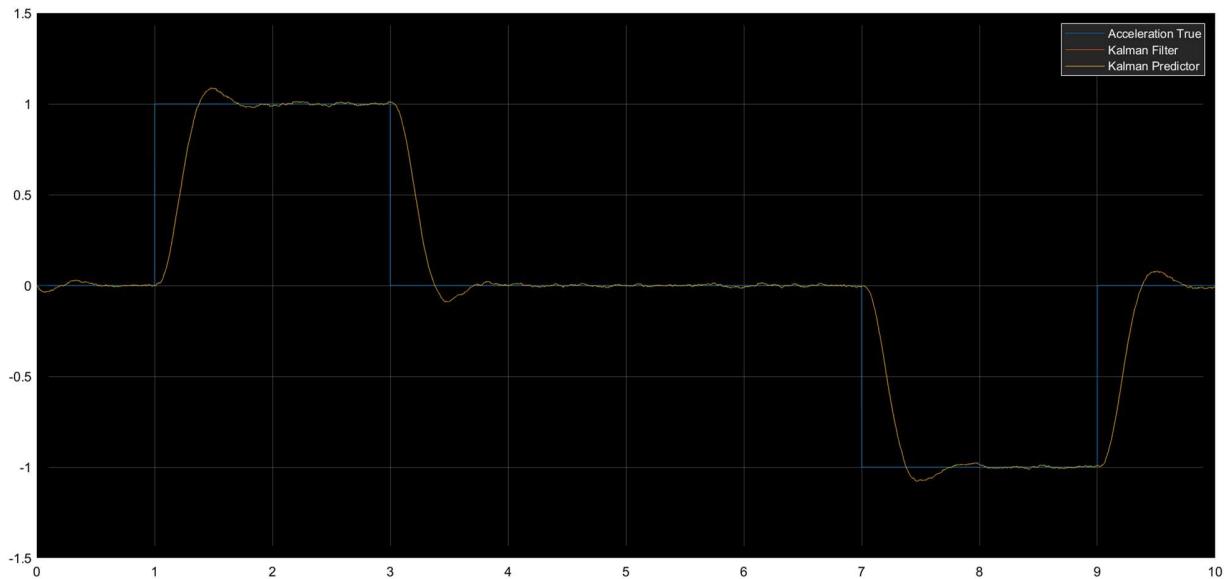


Figure 61 Acceleration graph of Kalman Filter and Predictor for Steady-State.

5. Kalman Smoother

Kalman smoothing is very similar to the standard Kalman filter, however, the Kalman smoother uses all data to move backward to smooth the previous data. Let assume the Kalman filter is used to estimate the velocity and acceleration up to N data points. The Kalman smoother uses all the N data points to smooth the previous estimated results of Kalman filter. However, by this technique the Kalman smoother produces more accurate results than the standard Kalman filter. Kalman Smoother is applied in two steps which are defined below.

1) Forward Step

Forward step uses the standard Kalman filter to estimate the velocity and acceleration. It uses the same equation as Kalman filter which is defined by Equations 19-21. However, it is necessary to store some information from the Kalman filter to use it in the backward step for smoothing the estimated data.

2) Backward Step

As the name suggests, the backward step estimate the previous velocity and acceleration by using the current estimated point, thus as stated earlier it needed a complete estimation of data to smooth the data points. Following Equation can be used for the smoothing process.

$$\hat{x}_{k|N}^s = \hat{x}_{k|k}^f + \tilde{K}_k(\hat{x}_{k+1|N}^s - \hat{x}_{k+1|k}^f) \quad (28)$$

$$P_{k|N} = P_{k|k}^f + \tilde{K}_k(P_{k+1|N} - P_{k+1|k}^f) \quad (29)$$

$$\tilde{K}_k = P_{k|N} A^T (P_{k+1|k}^f)^{-1} \quad (30)$$

where N is the number of total estimated data by Kalman filter. The super script f states the filter parameter and the super script s states the smoother parameter.

The Kalman smoother shows promising results for the velocity and acceleration as shown in Figure 62 and Figure 63. AS the velocity graph did not show any significant difference between the Kalman smoother and filter, but the acceleration graph shows some interesting trends. According to Figure 63, the acceleration of the Kalman smoother is much smoother and close to its actual value then the Kalman Filter itself. This is due to the fact that the backward step uses the recursion to estimate again the data point using its next estimated points.

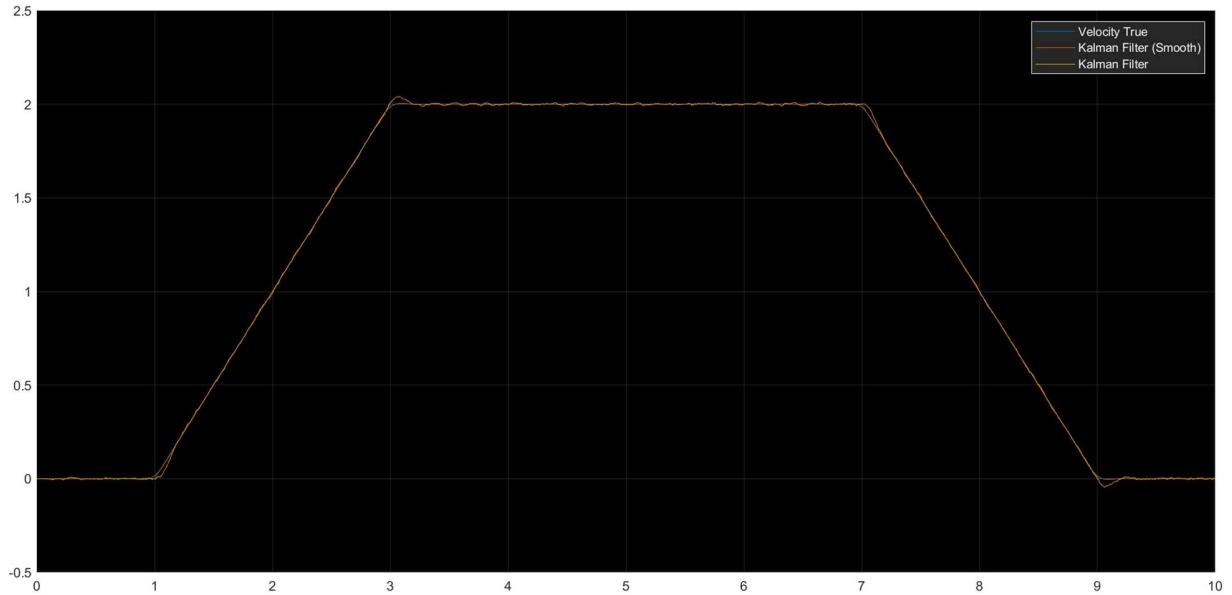


Figure 62 Velocity graph for the Kalman smoother.

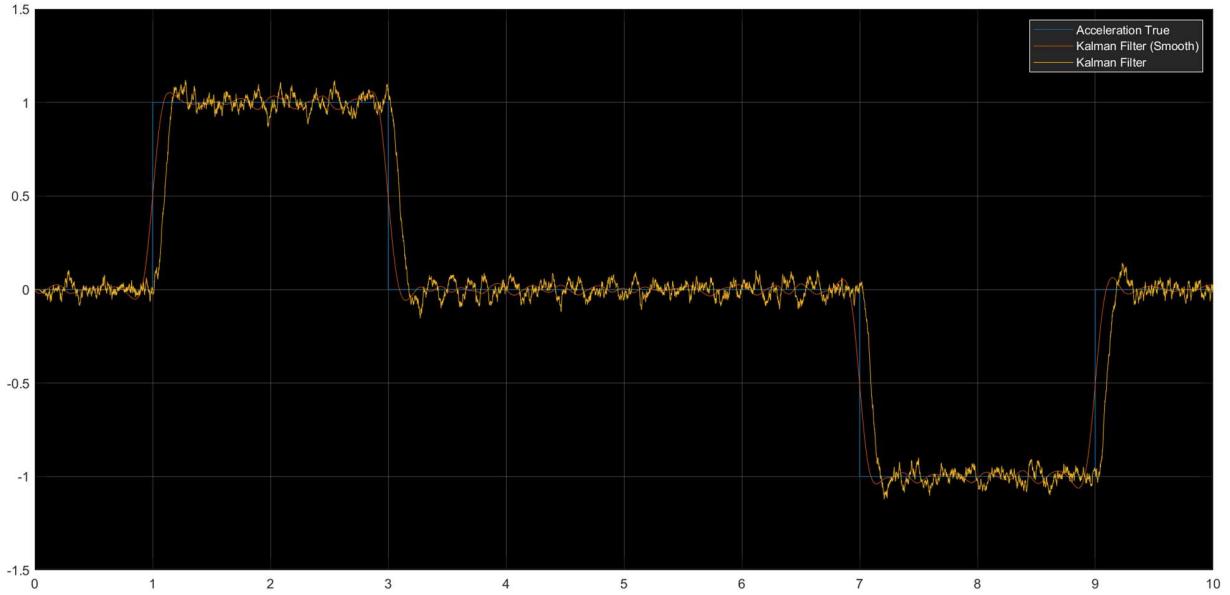


Figure 63 Acceleration graph for the Kalman smoother.

Table 3 Root Mean Square Error of Kalman Filter and Kalman Smoother for Velocity and Acceleration. (Lower is better)

	Velocity	Acceleration
Kalman Filter	0.0085	0.1804
Kalman Smoother	0.0020	0.0728

In Table 3, the results of Kalman smoother and Kalman Filter show in RMSE. It can be clearly seen that the Kalman smoother outperform as compared to the Kalman filter in acceleration as well as velocity estimation.

6. Linear Methods for Regression

Linear methods of regression are for modeling the relationship between one or more independent variables and a dependent variable. It uses statistical concept to model between the dependent and independent variables. Linear regression models use a straight line to describe the relationship between two scalar values: the input variable x and the output variable y . The model assumes that y is a linear function of the input variable x , which implies that the y will decrease or increase linearly when the x input changes. In this literature, the Least Square (LS) and Recursive Least Square (RLS) linear methods for regression is used to model the τ and K using the voltage, acceleration and velocity. Where τ is the motor torque and K is the torque constant. However, to model the motor the following equation can be used.

$$\frac{\tau}{K} \dot{\omega}(t) + \frac{1}{k} \omega(t) = V(t) \quad (31)$$

the above equation became in matrix form

$$[\dot{\omega}(t) \ \omega(t)] \begin{bmatrix} \frac{\tau}{K} \\ 1 \\ \frac{1}{k} \end{bmatrix} = V(t) \quad (32)$$

where the identifier to each matrix can be assigned as

$$x \triangleq [\dot{\omega}(t) \ \omega(t)] \quad (33)$$

$$y \triangleq V(t) \quad (34)$$

$$\theta \triangleq \begin{bmatrix} \frac{\tau}{K} \\ 1 \\ \frac{1}{k} \end{bmatrix} \quad (35)$$

the Equation 32 became

$$x\theta = y \quad (36)$$

6.1. Least Square

The least squares method is a way of finding the line of best fit that minimizes the sum of the squared distances between the observed data points and their corresponding values on a line. For example, let's say you have a set of data points that you want to fit a line to. The least squares method would find the line that minimizes the sum of the squared distances between each data point and its corresponding fitted value on the line. This line would be the line of best fit for that set of data points. To calculate the optimal line which fit all the data points normal equation is used to which is defined as

$$\beta = (X^T X)^{-1} X^T Y \quad (37)$$

where β is the coefficient matrix, X is the input matrix and Y is the output matrix. In this experiment, the Kalman filter is used to estimate the input matrix i.e. X which contains acceleration $\dot{\omega}$ and velocity ω . However, the Y is provided as a Voltage in this scenario.

Figure 64 shows the voltage graph between the true value and the predicted values. The accuracy between the predicted and actual value determines the accuracy of the model. The figure illustrates that the predicted value has very little difference than the actual value. More precisely if take account of the RMSE then the error is 3.1548. However, to extract the τ and the K relationship of Equation 36 can be used with the learned parameter β .

$$\beta_{2,1} = \frac{1}{K} \quad (38)$$

$$\beta_{1,1} = \frac{\tau}{K} \quad (39)$$

hence the equations became

$$K = \frac{1}{\beta_{2,1}} \quad (40)$$

$$\tau = \beta_{1,1} K \quad (41)$$

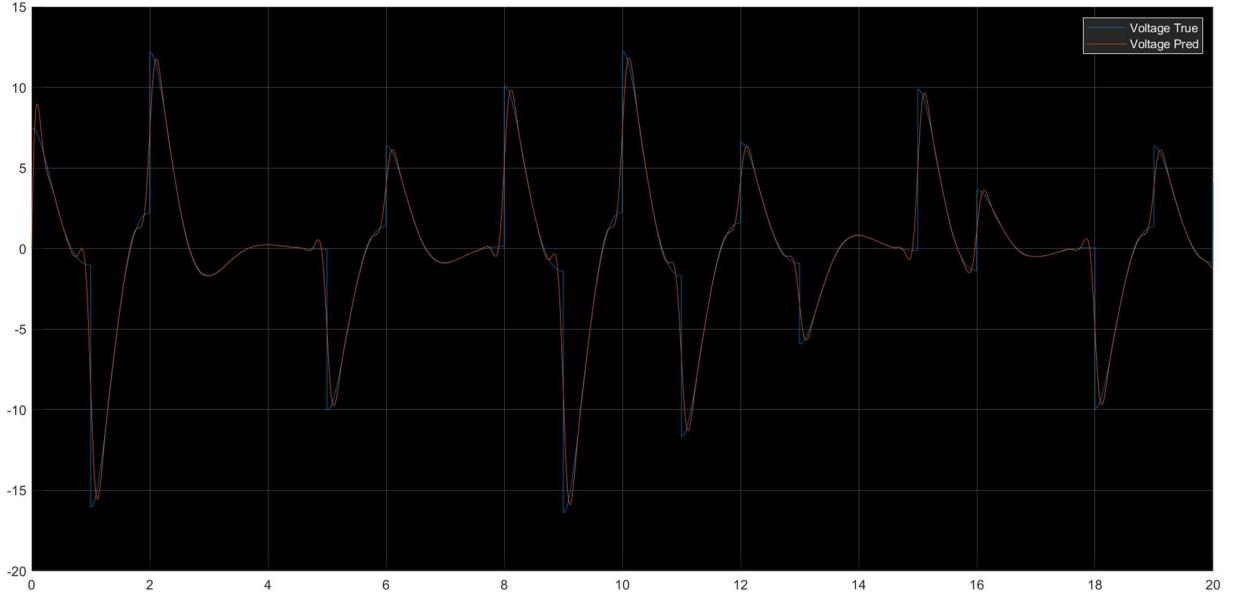


Figure 64 Voltage graph of prediction and actual values for Least Square estimator.

6.2. Recursive Least Square

The recursive least square estimator (RLSE) is a method for estimating the parameters of a linear model in an online fashion, that is, without storing all the past data. The RLSE updates the parameter estimates at each time step by using the new data point and the previous estimates. The RLSE minimizes the sum of squared errors between the observed outputs and the predicted outputs of the model. However, the forgetting factor can also be used to provide the less weights to the older estimates, and it is denoted by the λ . The recursive least square estimator can be defined using couple of equations, these equations are defined as

$$\hat{Y}(k) = X(k)\hat{\beta}(k) + e \quad (42)$$

$$\hat{\beta} = \hat{\beta}(k-1) + K(k)e \quad (43)$$

$$K(k) = P(k)X^T(k) \quad (44)$$

$$e = Y(k) - X(k)\hat{\beta}(k-1) \quad (45)$$

$$P(k) = \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)X^T(k)X^T(K)P(k-1)}{\lambda + X(k)P(k-1)X^T(k)} \right] \quad (46)$$

According to equation 42, the target value \hat{Y} can be predicted using the current data point $X(k)$ where k is the iterator. $\hat{\beta}$ is the learnable parameter and it is dependent on the gain K and the

estimation error e . Moreover, $P(k)$ is the covariance matrix of the current data point with the forgetting factor λ . As stated earlier, λ helps the estimator to provide the less weight to the previous estimation hence act as a forgetting factor.

Figure 65 shows the Voltage graph of the recursive least square error. According to the graph, there is very little estimation error. However, if the graph of Least Square in Figure 64 is compared with the Recursive Least Square graph, dramatic improvement can be seen. Furthermore, if the RMSE is considered for RLS, the error is as low as 0.0868. However, to extract the τ and the K relationship of Equation 40 and 41 can be used with the learned parameter β .

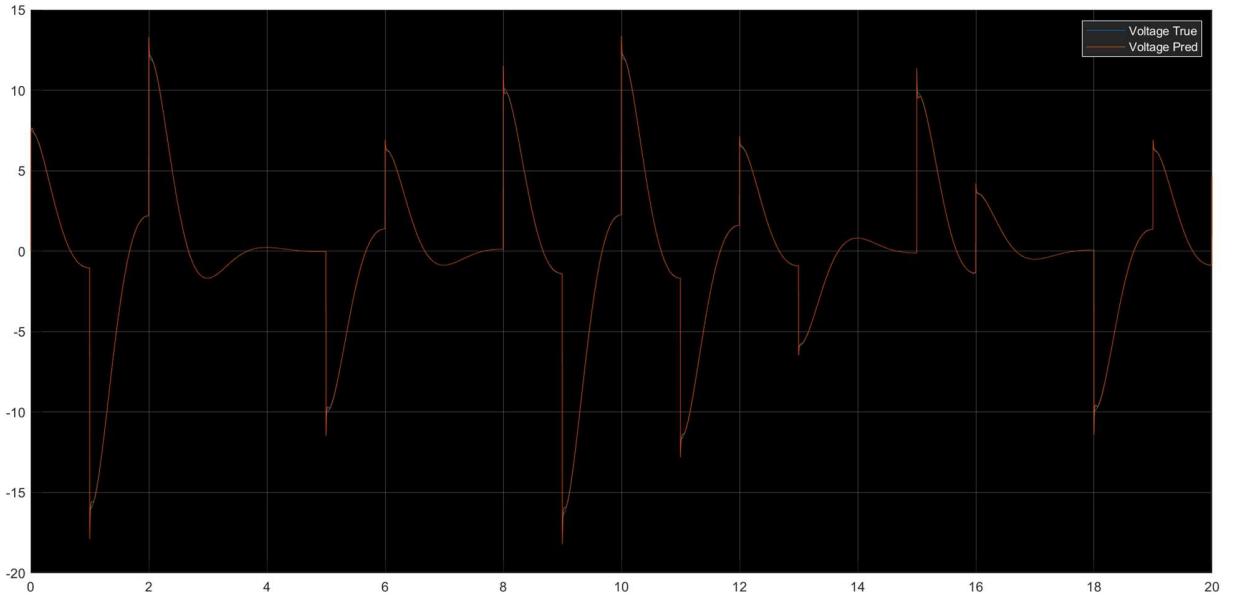


Figure 65 Voltage graph of prediction and actual values for Recursive Least Square estimator.

7. Scattering-based Bilateral Teleoperation Architecture

Scattering-based bilateral teleoperation architecture is a framework for designing and implementing teleoperation systems that can achieve stability and transparency under various communication conditions [6]. The main idea of this architecture is to use scattering operators to model the interaction forces between the master and slave devices, and between the slave device and the master. Scattering operators are mathematical tools that can capture the energy exchange and dissipation in physical systems. By applying scattering operators to both ends of the communication channel, the teleoperation system can be decoupled into two passive subsystems that are interconnected by a power-conserving channel. This way, the stability of the system is guaranteed regardless of the time delay. Moreover, passivity is the phenomenon in which the system does not generate any energy by itself. Hence this phenomenon leads to the stability of the whole system.

As stated earlier, the power variables i.e. Forces and Velocities into wave variables and then wave variables are used into transportation channel. After receiving the wave variables it is then

converted back to the power elements as shown in Figure 66. In the transportation of the variables the delay can happen which leads to undesired behavior as already discussed in Section 3. However the waves variables are unvulnerable to the transportation channel delay and does not cause any uncertainties. According to the figure, the A and B block are systems that convert the power variables to the wave variables and vice versa.



Figure 66 Flow diagram of the scattering wave-based system.

As in the current teleoperation architecture, the power variables are of two types i.e. Force (F) and Velocity (V), and it is important to design the specific system that converts the particular power variable to the particular wave variable. There are three types of architecture, that converts the power variables to its respective wave variables. These three architectures are discussed below

1) Forces

In this architecture, the forces are input i.e. F_l and F_r and the output will be the velocities i.e. \dot{x}_l and \dot{x}_r . The system can be defined as

$$\text{Left Port} = \begin{cases} u_l = \sqrt{\frac{2}{b}} F_l - v_l \\ \dot{x}_l = \frac{1}{b} (F_l - \sqrt{2b} v_l) \end{cases} \quad (47)$$

$$\text{Right Port} = \begin{cases} u_r = \sqrt{\frac{2}{b}} F_r - v_r \\ \dot{x}_r = -\frac{1}{b} (F_r - \sqrt{2b} v_r) \end{cases} \quad (48)$$

2) Velocities

In this architecture, the velocities are input i.e. \dot{x}_l and \dot{x}_r and the output will be the forces i.e. F_l and F_r . The system can be defined as

$$\text{Left Port} = \begin{cases} u_l = \sqrt{2b} \dot{x}_l - v_l \\ F_l = b \dot{x}_l + \sqrt{2b} v_l \end{cases} \quad (49)$$

$$\text{Right Port} = \begin{cases} u_r = -\sqrt{2b} \dot{x}_r + v_r \\ F_r = b \dot{x}_r + \sqrt{2b} v_r \end{cases} \quad (50)$$

3) Mixed

It is possible to mix the power variable e.g. one port can have force (F_l) and other port can have the velocity (\dot{x}_r) and vice versa. It is useful in the situation if the bilateral FP teleoperation architecture is used. This system can be stated as

$$\text{Left Port} = \begin{cases} u_l = b\dot{x}_l + \frac{1}{\sqrt{2b}}F_l \\ F_l = b\dot{x}_l + \sqrt{2b}v_l \end{cases} \quad (51)$$

$$\text{Right Port} = \begin{cases} u_r = \frac{1}{\sqrt{2b}}(2F_r - v_r\sqrt{2b}) \\ \dot{x}_r = \frac{1}{b}(\sqrt{2b}v_r - F_r) \end{cases} \quad (52)$$

7.1. Force-Position (F-P)

In this sub section, bilateral force position architecture defined in Section 3.2 modified to generate the scattering based bilateral teleoperation architecture. In this this architecture, the whole system is transferred from the continuous system to the discrete time system with the time interval T_s . It is important to note that only the communication channel is modified, and all other components are the same as the original architecture. It is also important to note that the full transparency communication channel is used for this experiment.

Figures 67-69 shows position, velocity, and force graph. According to the position graph, the slave position converges to the master position accurately. It is important to note that the delay use in this experiment is quite high, which is 10 steps. It means that the packet the slave received is 10

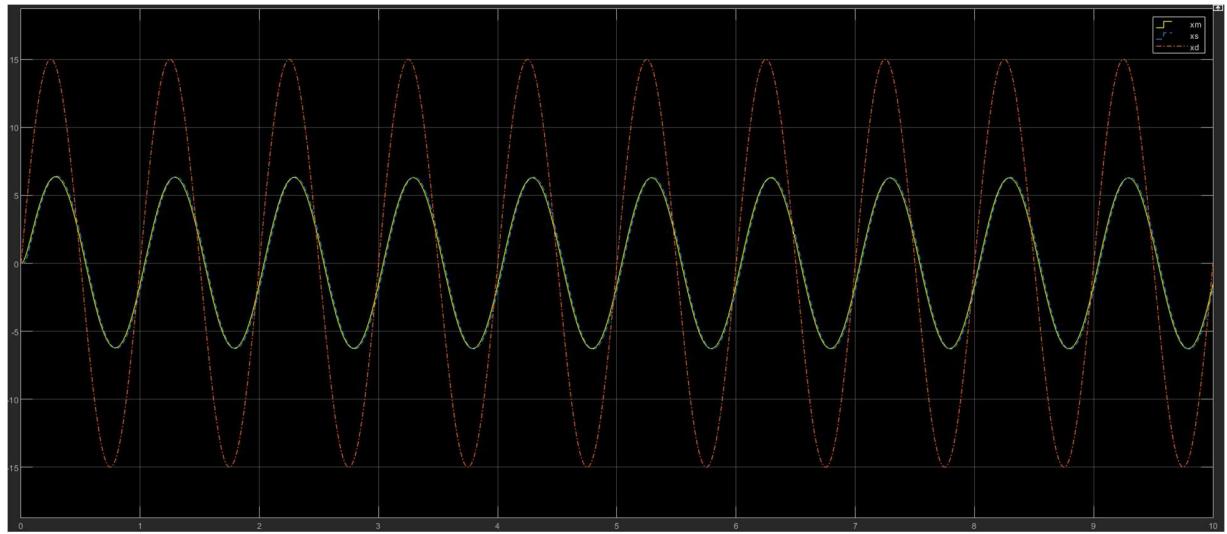


Figure 67 Position graph of scattering based bilateral teleoperation in free motion.

steps delayed and the packet master received is 10 steps delayed. However, with the presence of the large delay the positions of the slave and master robot converge to each other due to the

scattering wave teleoperation architecture. Similarly, the velocity graph shows the same behavior, the slave velocity easily converges to the master velocity. Furthermore, the slave robot is not interacting with the environment, the force of the slave robot is zero. This behavior is as same as discussed in the Section 3.2.



Figure 68 Velocity graph of scattering based bilateral teleoperation in free motion.

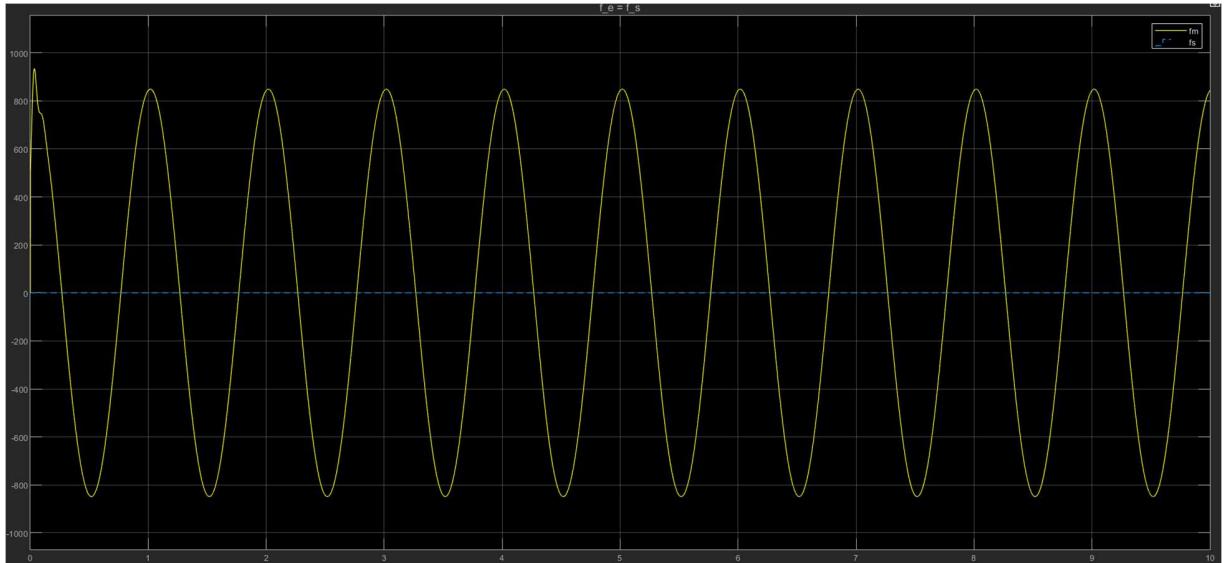


Figure 69 Force graph of scattering based bilateral teleoperation in free motion.

According to Figure 70, the position of the slave robot is restricted due to the environment and could not converge to master robot. However, this behavior is same as Section 3.2, which implies that the delay is not affecting the output of the teleoperation controller. Similar phenomena is observed in the velocity graph. According to Figure 71, the velocity does not converges to the

master robot at the point where the slave robot is interacting with the environment. Moreover, the Figure 72 shows the Force graph of the Slave and Master robot, the master robot experiences the force from the slave robot and act to reduce the impact of the force.

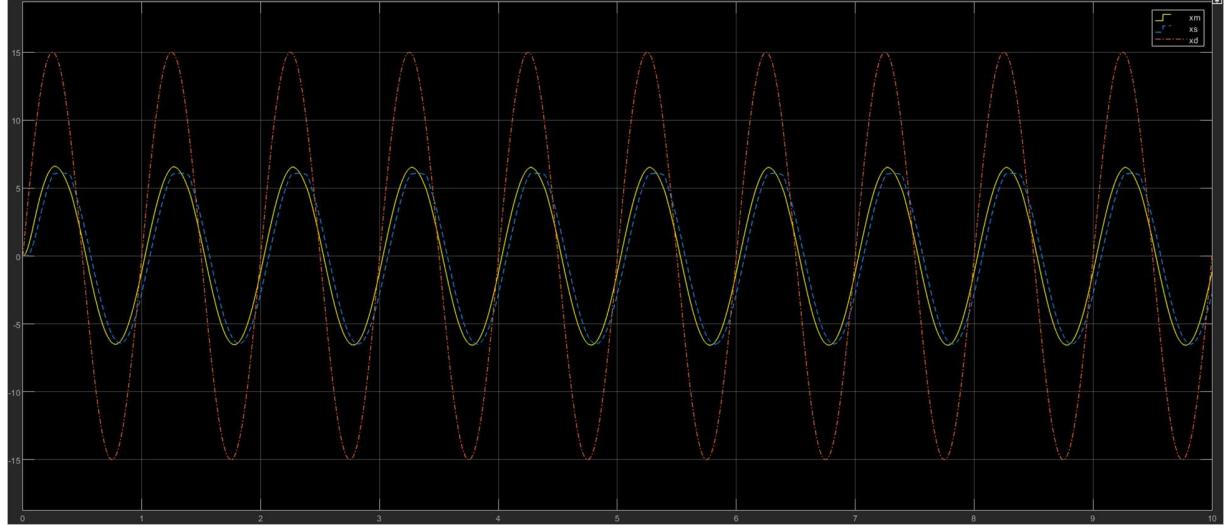


Figure 70 Position graph of scattering based bilateral teleoperation with environment interaction.

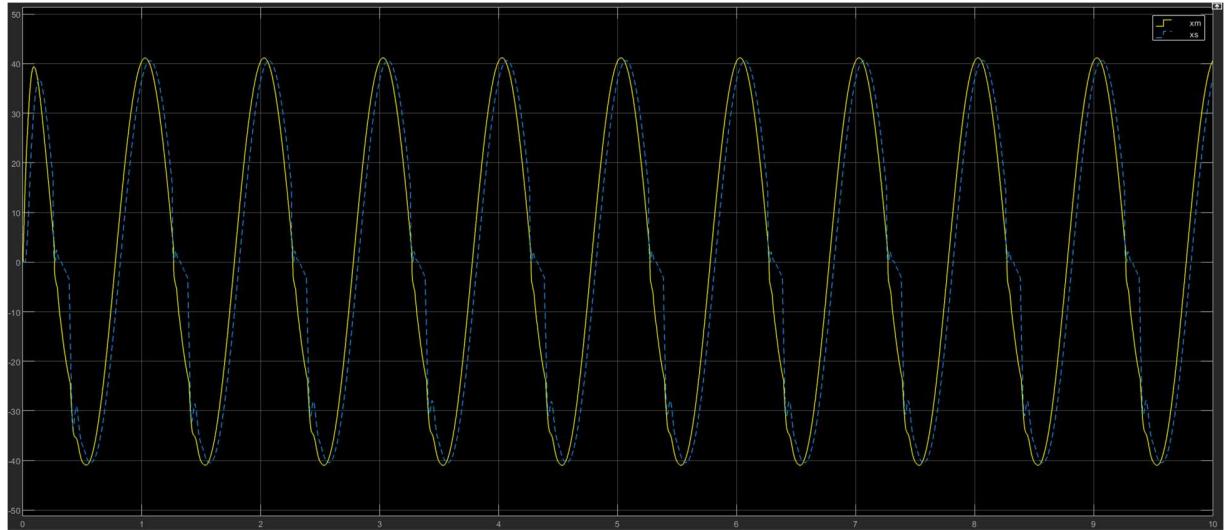


Figure 71 Velocity graph of scattering based bilateral teleoperation with environment interaction.

If the position sensor, i.e. GPS produces noisy results, the controller might perform indeterministic way. In this experiment, this situation is mimicked where the noise is added to the position. In Section 4.1, it is already discussed that the Kalman filter can be used to estimate the correct velocity and position with the help of noisy position. In this experiment, the Kalman filter is used to estimate the velocity and position using the noisy position. As it can be seen from Figure 73, Despite the noise, the slave robot accurately converges to the master robot. However,

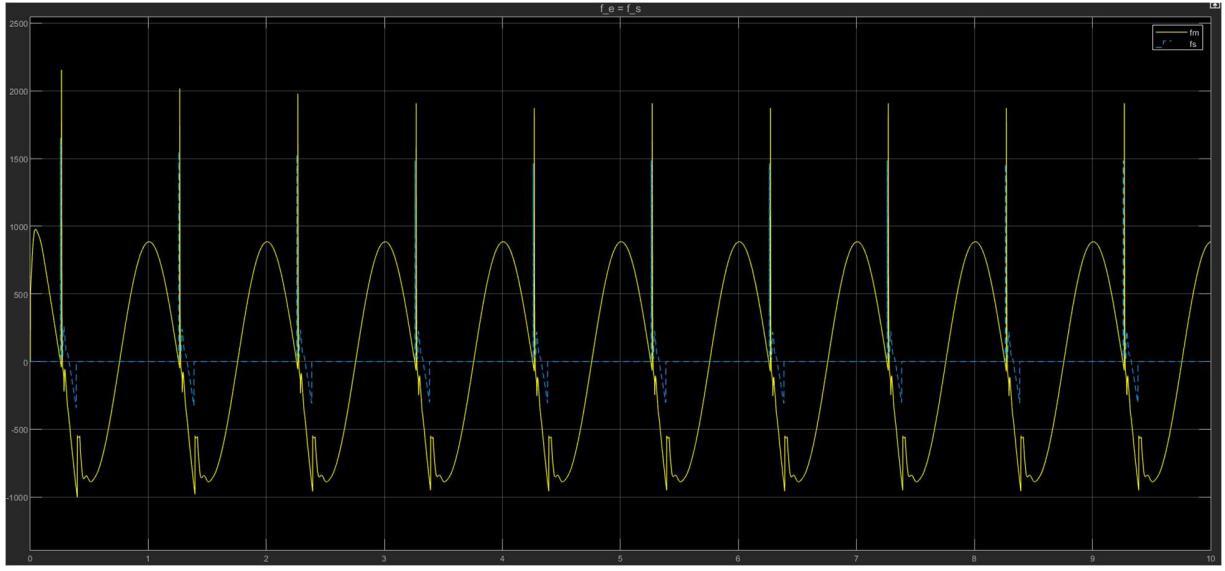


Figure 72 Force graph of scattering based bilateral teleoperation with environment interaction.

the slave robot and master robot show some disturbances in the beginning of the velocity graph due to the little inaccurate estimations as shown in Figure 74. As force is dependent on the acceleration, and the acceleration depends on the velocity, the force shows some disturbances on the master side because of the estimation of the velocity from the noisy position using the Kalman filter as shown in Figure 75.

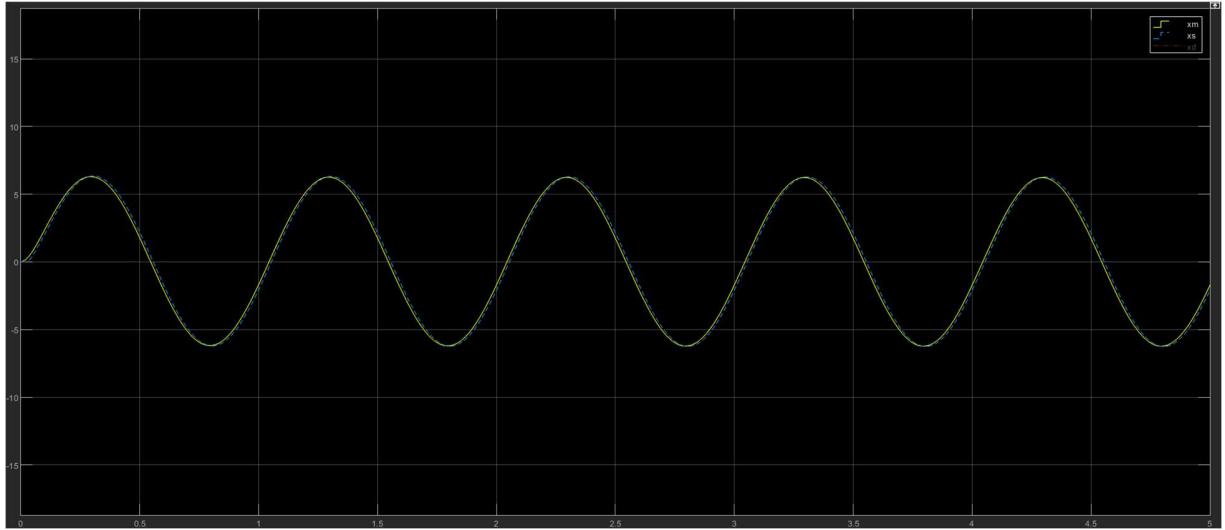


Figure 73 Position graph of scattering based bilateral teleoperation in free motion with Kalman filter to predict velocity using noisy position.

When the position of both robot i.e. master and slave robot is noisy and the slave robot is interacting with the environment then at the point of interaction in the position graph shows the great disturbance due to the changing of the variance from the previous points. This phenomenon is visible in Figure 76. Similarly, the velocity is estimated from the noisy position

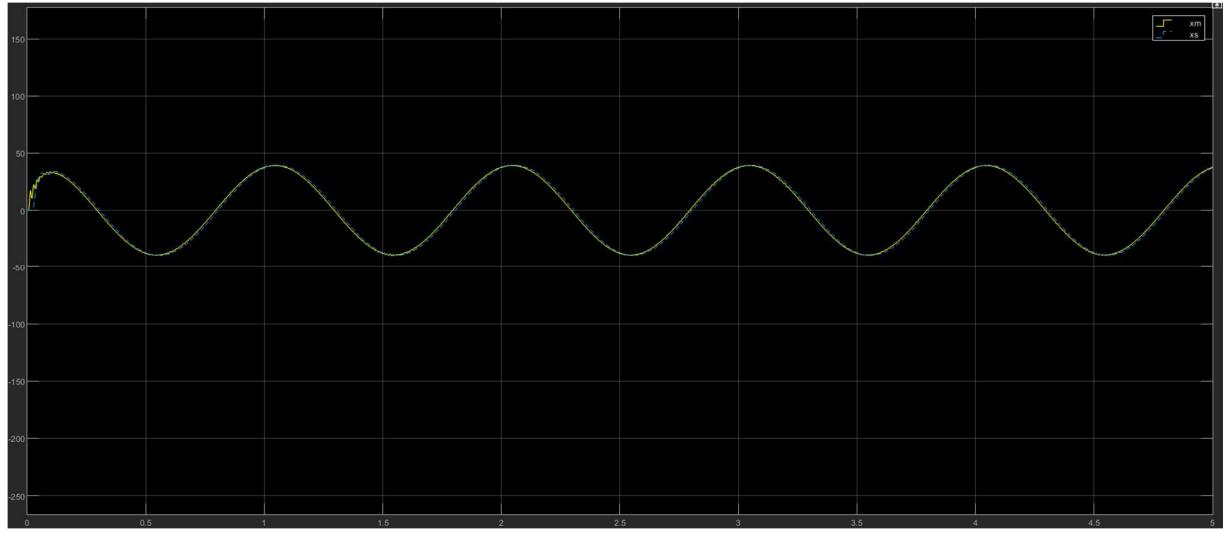


Figure 74 Velocity graph of scattering based bilateral teleoperation in free motion with Kalman filter to predict velocity using noisy position.

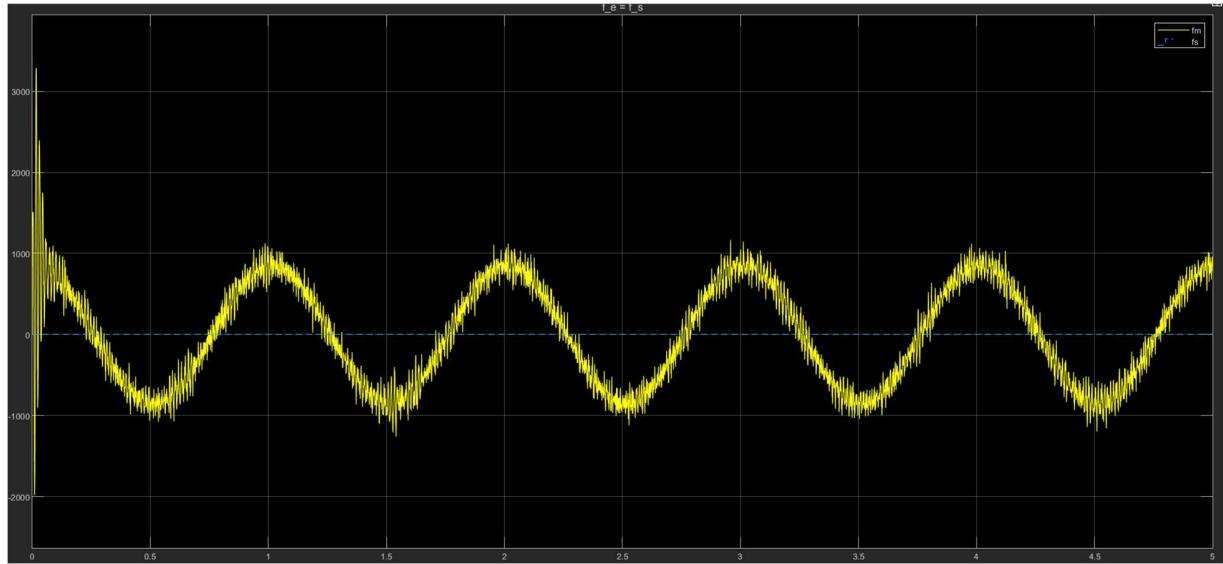


Figure 75 Force graph of scattering based bilateral teleoperation in free motion with Kalman filter to predict velocity using noisy position.

and acceleration is dependent on the velocity, the great disturbances can be found in the velocity and force graph in Figure 77 and Figure 78.

7.2. Position-Position (P-P)

This sub section presents a modified version of the bilateral force position architecture from Section 3.1, which incorporates scattering-based bilateral teleoperation. In this architecture, the entire system is discretized with a sampling time T_s . The only component that changes from the original architecture is the communication channel, which uses scattering-based approach for bilateral teleoperation. All other components remain unchanged and full transparency

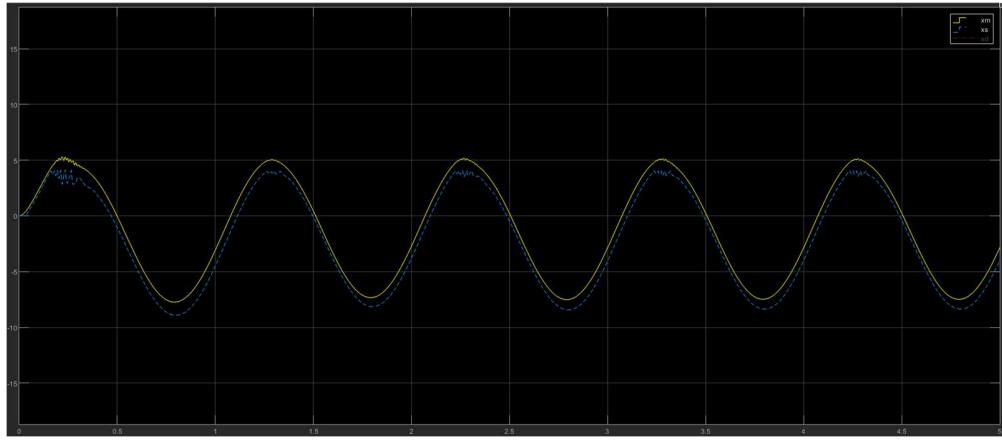


Figure 76 Position graph of scattering based bilateral teleoperation under environment interaction with Kalman filter to predict velocity using noisy position.

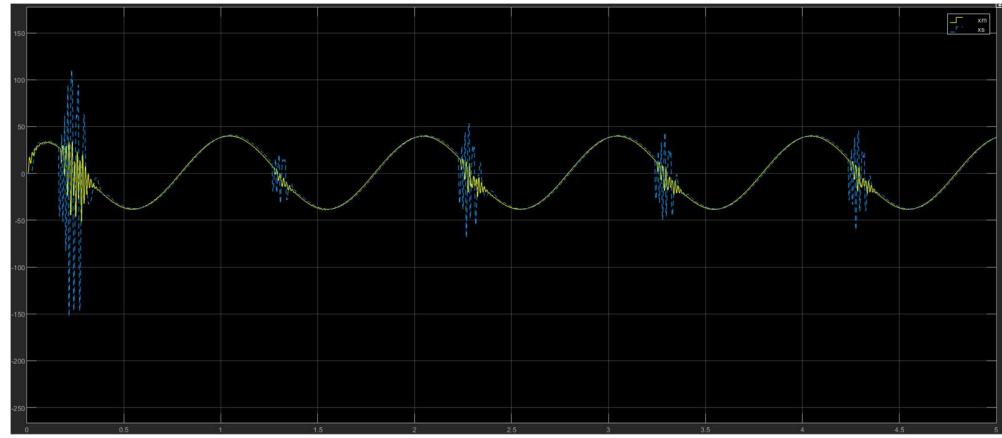


Figure 77 Velocity graph of scattering based bilateral teleoperation under environment interaction with Kalman filter to predict velocity using noisy position.

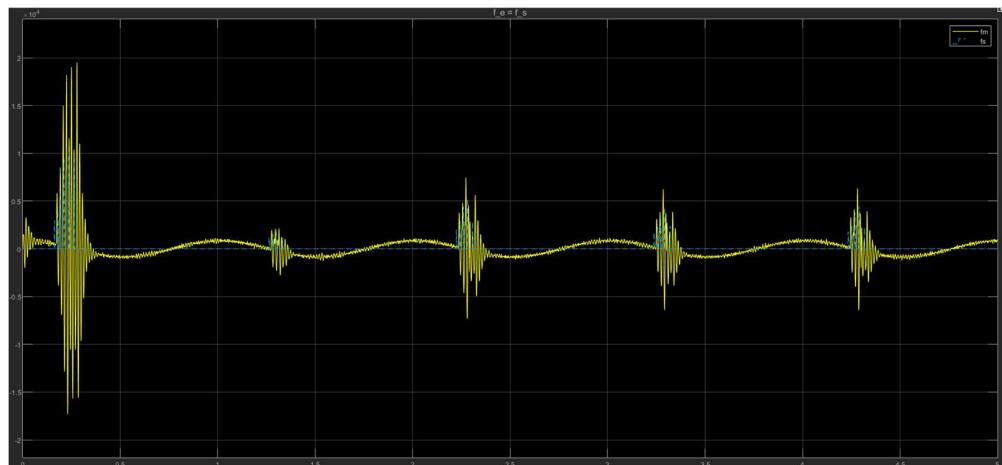


Figure 78 Force graph of scattering based bilateral teleoperation under environment interaction with Kalman filter to predict velocity using noisy position.

architecture is used. The purpose of this modification is to experiment with the scattering-based approach for bilateral teleoperation.

The position, velocity, and force graph are shown in Figures 79-81. The position graph indicates that the slave position accurately matches the master position. This is remarkable considering that the experiment uses a high delay of 10 steps. This means that both the slave and the master receive packets that are 10 steps outdated. Nevertheless, the scattering wave teleoperation architecture enables the slave and master positions to converge with each other. The velocity graph exhibits a similar pattern, as the slave velocity quickly aligns with the master velocity. Moreover, since the slave robot does not interact with the environment, its force is zero. This is consistent with the discussion in Section 3.1.

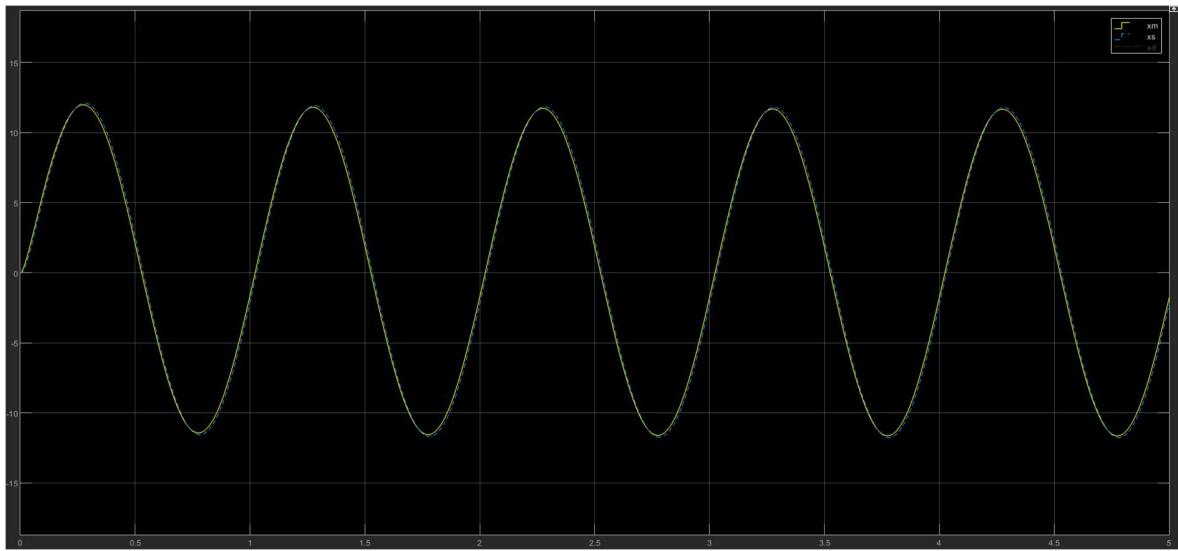


Figure 79 Position graph for Position-Position scattering based bilateral teleoperation without environment interaction.

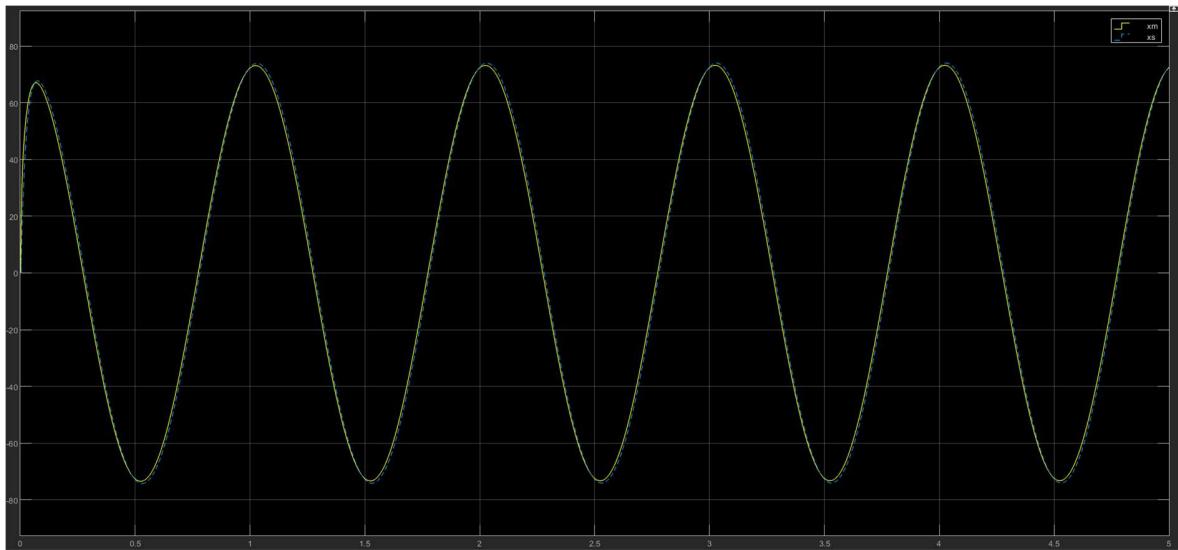


Figure 80 Velocity graph for Position-Position scattering based bilateral teleoperation without environment interaction.

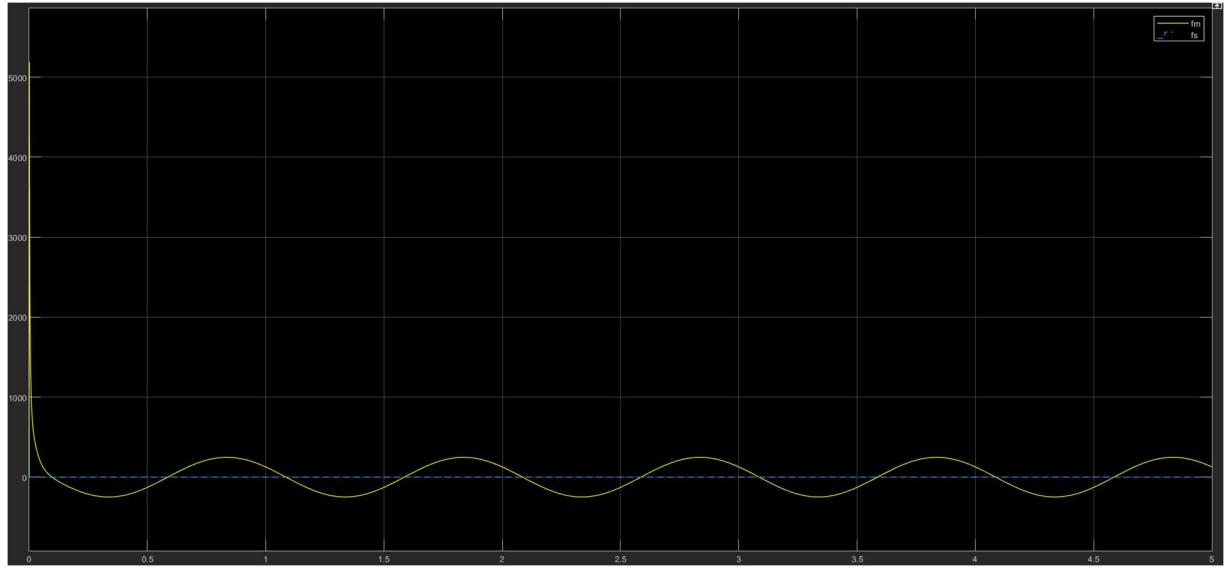


Figure 81 Force graph for Position-Position scattering based bilateral teleoperation without environment interaction.

As shown in Figure 82, the slave robot's position is limited by the environment and does not match the master robot's position. This is consistent with the results in Section 3.1, which indicate that the teleoperation controller's output is not affected by the delay. The velocity graph in Figure 83 also shows a similar pattern, where the slave robot's velocity diverges from the master robot's velocity when it encounters the environment. In addition, Figure 84 displays the Force graph of the Slave and Master robot, where the master robot feels the force from the slave robot and reacts to minimize its impact.

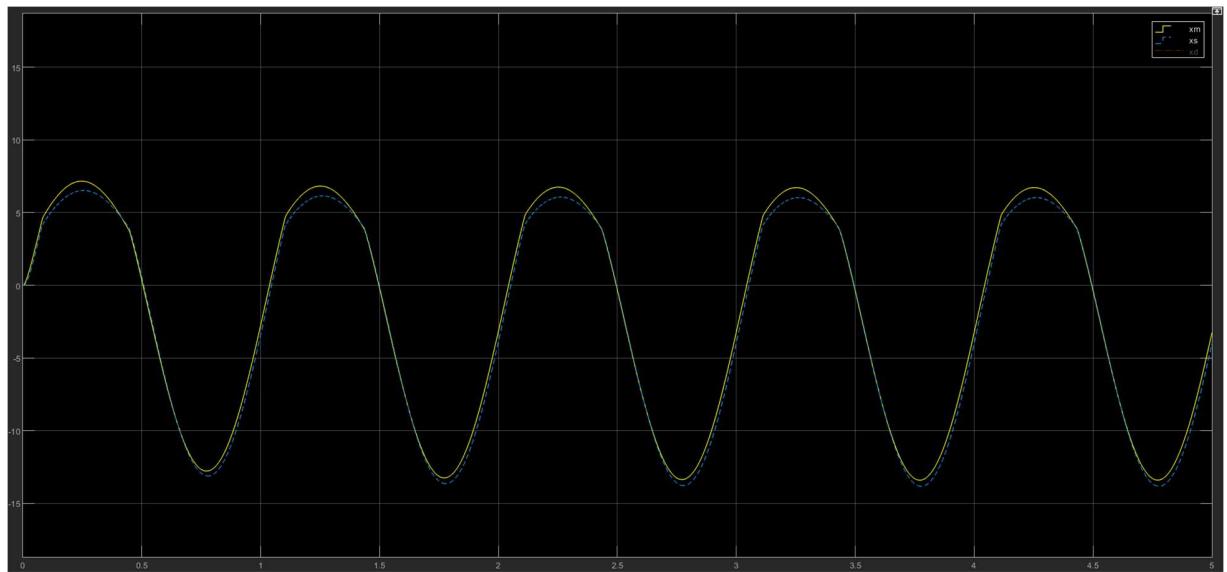


Figure 82 Position graph for Position-Position scattering based bilateral teleoperation under environment interaction.

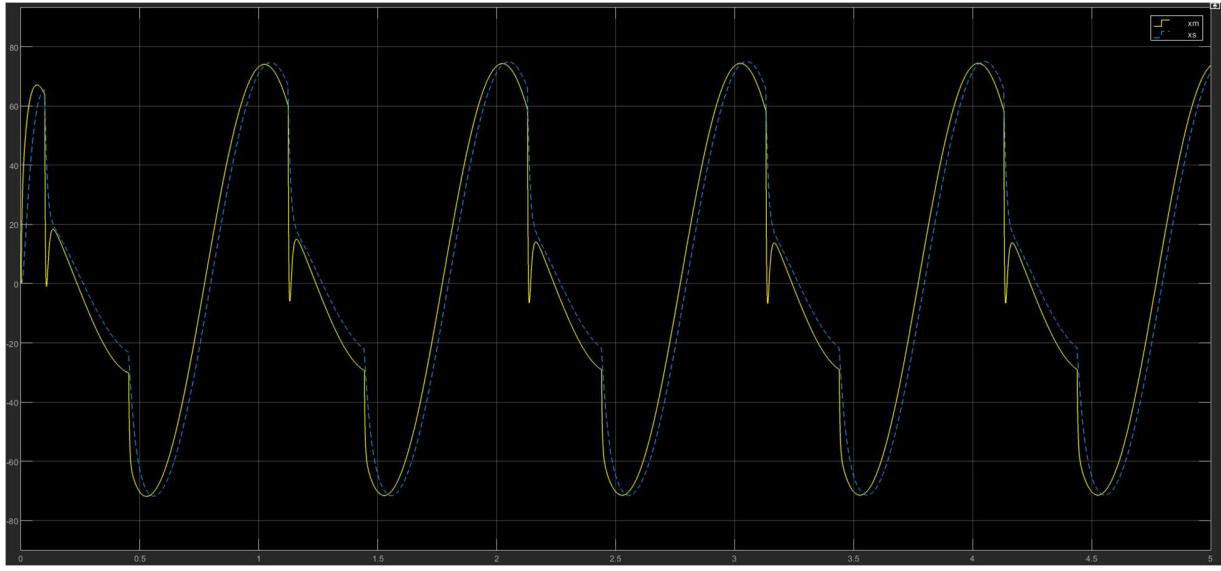


Figure 83 Velocity graph for Position-Position scattering based bilateral teleoperation under environment interaction.

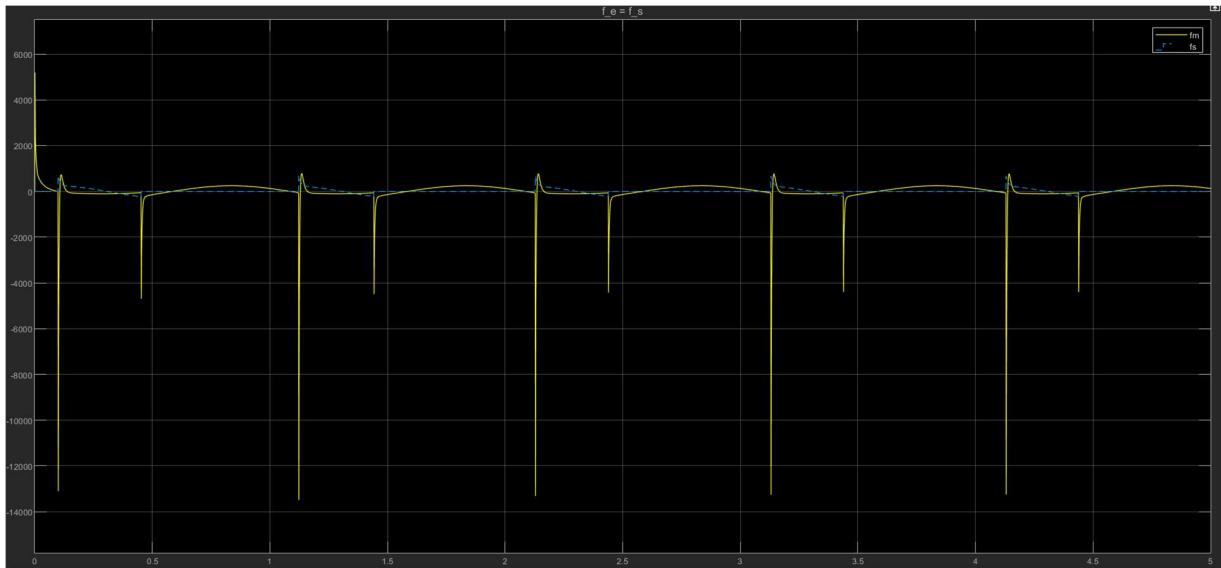


Figure 84 Force graph for Position-Position scattering based bilateral teleoperation under environment interaction.

The position sensor, such as GPS, may produce unreliable results that affect the controller's performance. This experiment simulates this scenario by adding noise to the position. The Kalman filter is a technique that can estimate the correct velocity and position from the noisy position, as explained in Section 4.1. This experiment applies the Kalman filter to estimate the velocity and position from the noisy position. Figure 85 shows that the slave robot follows the master robot accurately despite the noise. However, there are some fluctuations in the velocity graph at the start, as Figure 86 illustrates, due to the slightly inaccurate estimations. Since the force depends on the acceleration, which in turn depends on the velocity, the force also exhibits

some fluctuations on the master side, as Figure 75 demonstrates, due to the estimation of the velocity from the noisy position using the Kalman filter.



Figure 85 Position graph for Position-Position scattering based bilateral teleoperation without environment interaction with Velocity estimate using Kalman filter.



Figure 86 Velocity graph for Position-Position scattering based bilateral teleoperation without environment interaction with Velocity estimate using Kalman filter.

The position graph in Figure 88 shows significant fluctuations at the point of contact between the slave robot and the environment, when both the master and slave robots have noisy positions. This is because the variance changes suddenly from the previous points. Likewise, the velocity and force graphs in Figure 89 and Figure 90 also exhibit large disturbances, since the velocity is derived from the noisy position and the acceleration depends on the velocity.

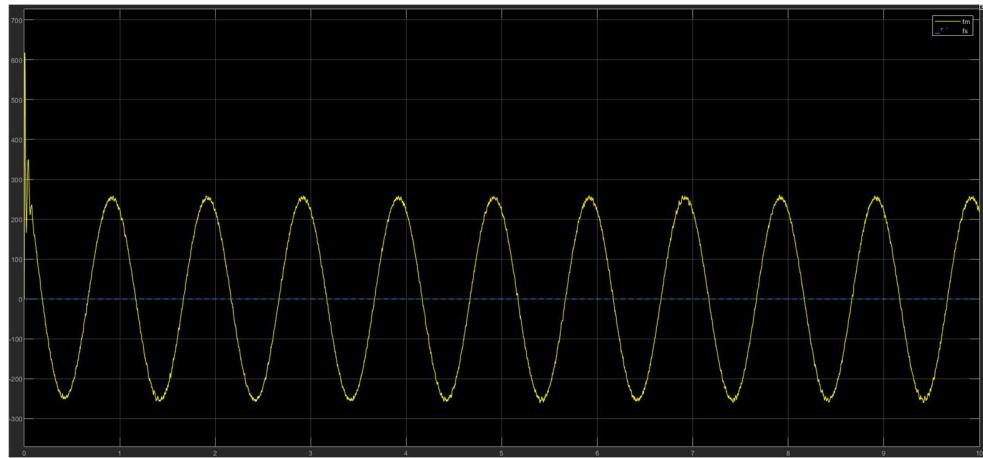


Figure 87 Force graph for Position-Position scattering based bilateral teleoperation without environment interaction with Velocity estimate using Kalman filter.

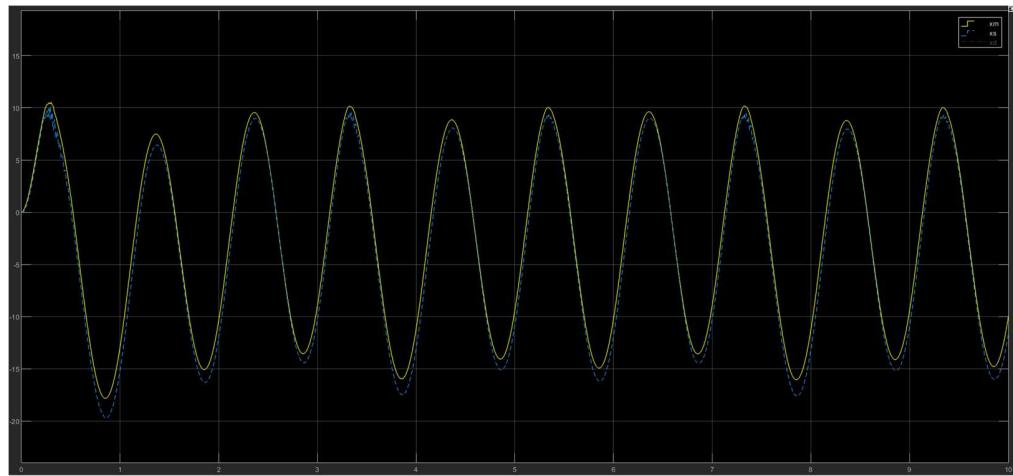


Figure 88 Position graph for Position-Position scattering based bilateral teleoperation under environment interaction with Velocity estimate using Kalman filter.

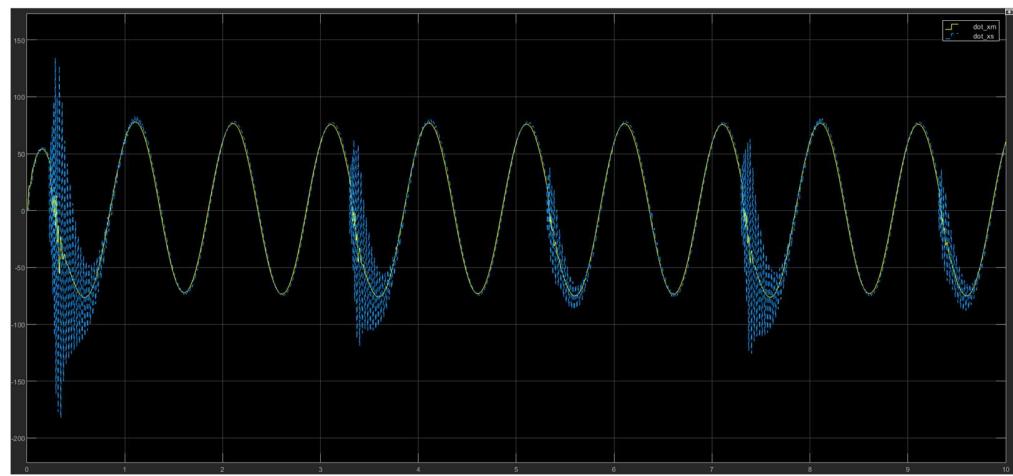


Figure 89 Velocity graph for Position-Position scattering based bilateral teleoperation under environment interaction with Velocity estimate using Kalman filter.

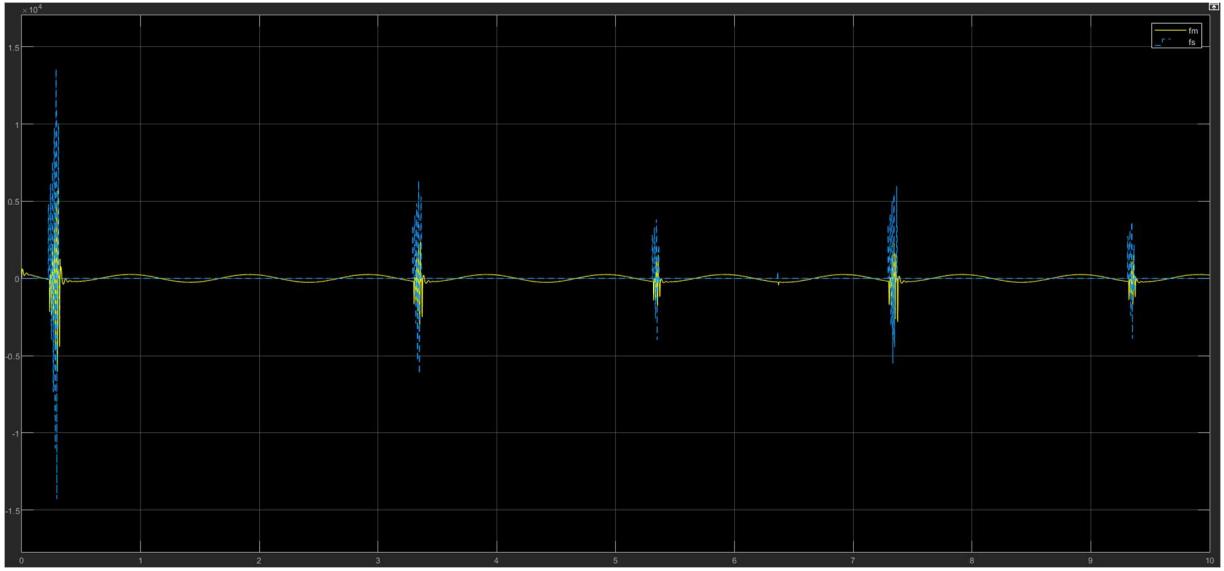


Figure 90 Force graph for Position-Position scattering based bilateral teleoperation under environment interaction with Velocity estimate using Kalman filter.

8. Tank-based bilateral teleoperation architecture

Bilateral telemanipulation is a challenging problem that involves the coordination of two robotic systems connected by a communication channel. The goal is to enable a human operator to manipulate a remote environment as if he or she were physically present there [7]. However, the presence of time delays and communication losses on the channel can compromise the stability and performance of the teleoperation system. To address this issue, the tank based bilateral teleoperation architecture uses a two-layer approach that combines passivity and transparency techniques.

The first layer is a transparency-based controller that compensates for the degradation of performance caused by the passivity layer. Transparency is a measure of how well the teleoperation system reproduces the dynamics of the remote environment and the human operator at both ends. Ideally, a transparent system should make the operator feel as if he or she were directly interacting with the remote environment, without any interference from the teleoperation system. However, passivity controllers tend to introduce damping and inertia effects that reduce the transparency of the system. To overcome this limitation, the transparency layer estimates the states of the remote environment and the human operator and applies a feedforward compensation to reduce the apparent impedance at both ends of the teleoperation system.

The second layer is a passivity-based controller that ensures the stability of the system under arbitrary time delays and communication losses. As stated earlier in Section 7, Passivity is a property that guarantees that a system does not generate more energy than it receives from its inputs. By enforcing passivity at both ends of the teleoperation system, the controller prevents the occurrence of oscillations or instabilities due to the delayed and unreliable feedback. The

passivity layer also preserves the physical interaction between the human operator and the remote environment, by applying force feedback that reflects the contact forces at both ends.

To achieve passivity, the tank-based communication channel is used. In the tank-based approach, two virtual energies tanks are used to power the slave and master robot. An energy tank is a virtual storage element that can absorb or supply energy to the system. The energy tank can compensate for the energy dissipation or injection caused by the time delay, and thus maintain passivity and stability. However, the total energy of the whole system is the summation of energy present in each components

$$H_T(t) = H_M(t) + H_C(t) + H_S(t) \quad (53)$$

where H_M , H_C , and H_S are the energies present in the master side, slave side and communication channel. Assuming the initially stored energy is zero, the passivity condition of the system is

$$H_T(t) \geq 0 \quad (54)$$

or if the robots physical side needed to be considered i.e. interaction with the environment and human operator interacts with the master robot then the following information should be preserved

$$\dot{H}_T(t) \leq P_M(t) + P_S(t) \quad (55)$$

where P_M and P_S are the flow of power from the master and slave robot to the master and slave controller respectively. \dot{H}_T is the rate of change of the energy balance of the total system.

Moreover the energy level of the tank $H(k)$ at the sampling time k can be calculated by

$$H(k) = H(\bar{k}) + H_+(k) - \Delta H_I(k) \quad (56)$$

where $H(\bar{k})$ is the energy level of the tank before any operation or exchange of energy. However, there is an energy queue which stores the packets of energy that have arrived at it. The $H_+(k)$ are the total arrived energy packets in the energy queue. It is defined as

$$H_+(k) = \sum_{i=1}^{Q(k)} \bar{H}(i) \quad (57)$$

where $Q(k)$ is the number of energy packets present in the energy queue and $\bar{H}(i)$ represents the i_{th} energy packet. Furthermore, $\Delta H_I(k)$ is the exchange of the energy between controller and physical world in time interval k and $k - 1$. It is denoted by

$$\Delta H_I(k) = \tau_r(\bar{k})\Delta q(k) \quad (58)$$

where τ_r is the torque of the robot at time interval \bar{k} and $\Delta q(k)$ is the difference in the position of the robot. The $H(\bar{k} + 1)$ is the energy that is remaining energy when all operation is performed, it is defined as

$$H(\overline{k+1}) = H(k) - H_-(k) \quad (59)$$

where $H_-(k)$ represents the energy that is leaved from the system.

The total energy that is present in the communication channel H_C at the sample time k can be defined as

$$H_C = \sum_{i=1}^k (H_{-M} - H_{+M} + H_{-S} - H_{+S}) \quad (60)$$

At the master side, the τ_{TLC} is calculated. Tank level control (TLC) main function is to monitor the master side tank $H_M(\overline{k+1})$ with respect to desired H_D . Whenever the energy of the tank is lower than the desired energy, a small amount of the energy is extracted from the user and stored in the energy tank. The energy extracted by the master controller is

$$\tau_{TLC} = -d(k)\dot{q}_m(k) \quad (61)$$

the τ_{TLC} is the torque for tank level control, it is defined by the viscous damper system. The d is the damping constant which is defined by

$$d(k) = \begin{cases} \alpha(H_D(k) - H_M(\overline{k+1})), & \text{if } H_D(k) > H_M(\overline{k+1}) \\ 0, & \text{otherwise} \end{cases} \quad (62)$$

the τ_{TLC} prevents the master controller to end up in the deadlock when all energy is used from the master tank. α is the constant and it is strictly positive. The H_C can be derived with the help of simple energy transfer protocol (STEP) which states that

$$\bar{H}_C = 2d\beta H_D \quad (63)$$

where d is the delay, β is the fraction constant and strictly positive. To calculate the torque on the master or slave side, the torque is calculated using the following formulas

$$\tau_{max1} = \begin{cases} 0, & \text{if } H(\overline{k+1}) \leq 0 \\ \tau_{TL}(k), & \text{otherwise} \end{cases} \quad (64)$$

$$\tau_{max2} = \frac{H(\overline{k+1})}{\dot{q}(\bar{k})T_s} \quad (65)$$

and

$$\tau_{PL}(k) = sign(\tau_{TL}(k))\min [\tau_{max1}, \tau_{max2}] \quad (66)$$

where τ_{max1} and τ_{max2} are the lower and upper bound torques, and τ_{TL} is the torque that is calculated by the master/slave side transparency layer and τ_{PL} is the torque that needs to be calculated by the passivity layer. However, the final torque for the master and slave side can be explained by

$$\tau(\overline{k+1}) = \begin{cases} \tau_{PL}(k) + \tau_{TLC}(k), & \text{for master} \\ \tau_{PL}(k), & \text{for slave} \end{cases} \quad (67)$$

8.1. Force-Position (F-P)

The bilateral force position architecture defined in Section 3.2 was modified to generate the tank based bilateral teleoperation architecture. In this architecture, the whole system is transferred from the continuous system to the discrete time system with the time interval T_s . Only the communication channel is modified, and all other components are the same as the original architecture. The full transparency communication channel is used for this experiment.

The graphs of position, velocity, and force are shown in Figures 91-94. The position graph indicates that the slave robot follows the master robot precisely, despite the large delay of 10 steps in the communication channel. This delay means that both the master and the slave receive packets that are 10 steps old. However, the tank-based teleoperation architecture ensures that the positions of the master and slave robots converge to each other. The velocity graph exhibits a similar pattern, with the slave velocity matching the master velocity. Moreover, since the slave robot does not interact with any environment, its force is zero. This behavior is consistent with the discussion in Section 3.2.

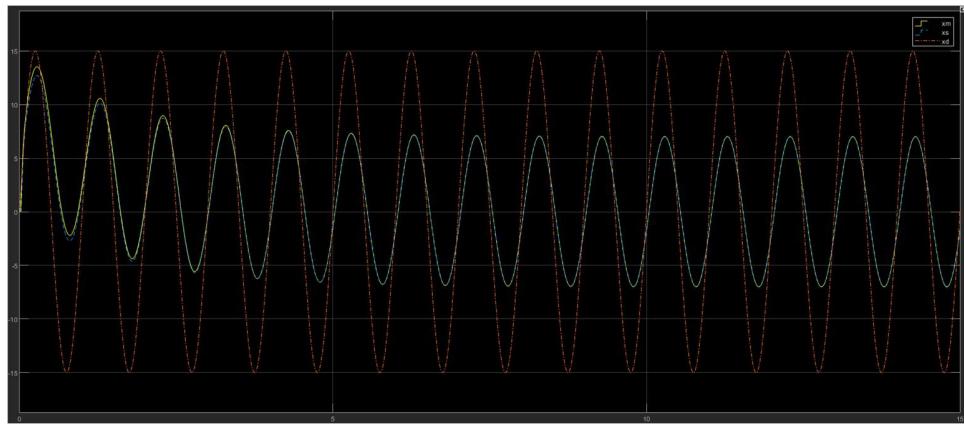


Figure 91 Position graph of the tank-based architecture in free motion.

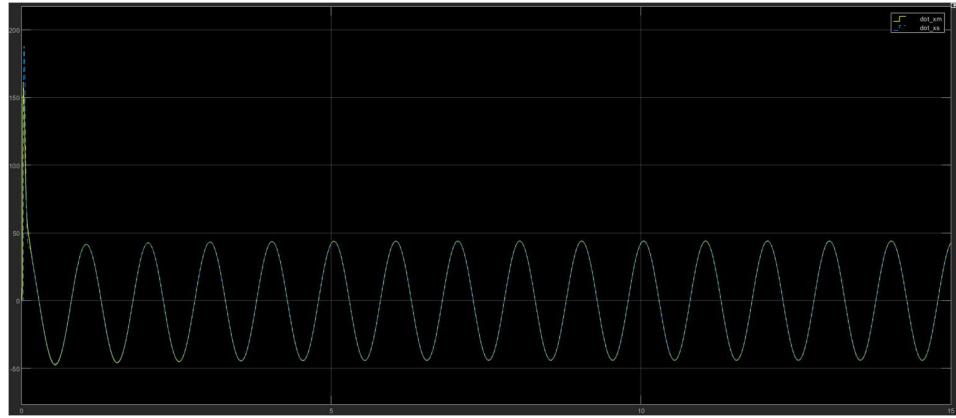


Figure 92 Velocity graph of the tank-based architecture in free motion.

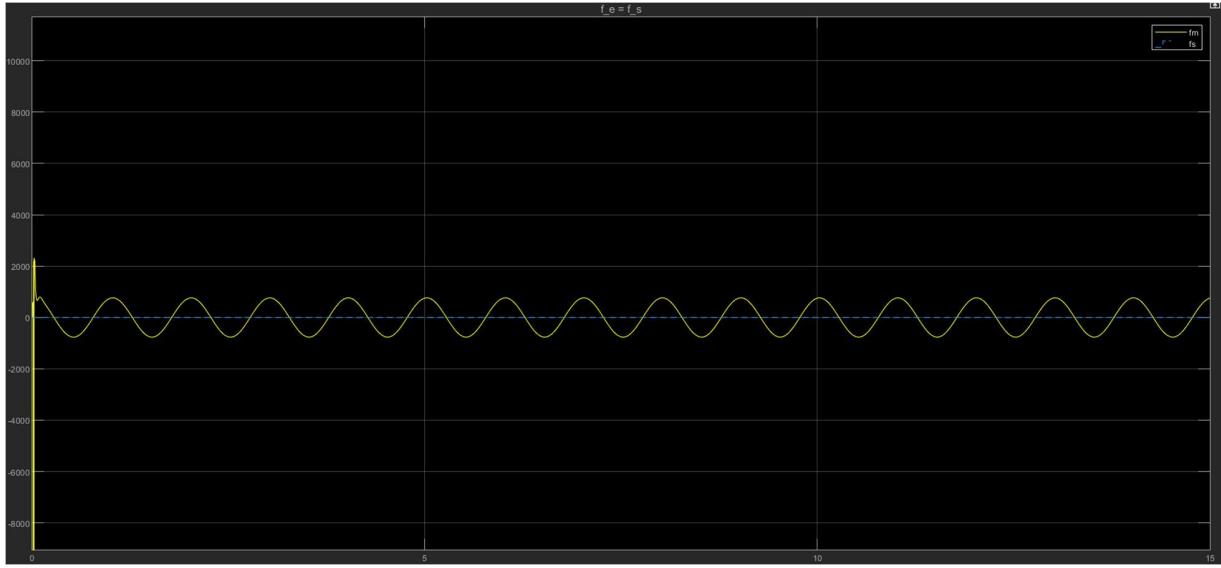


Figure 93 Force graph of the tank-based architecture in free motion.

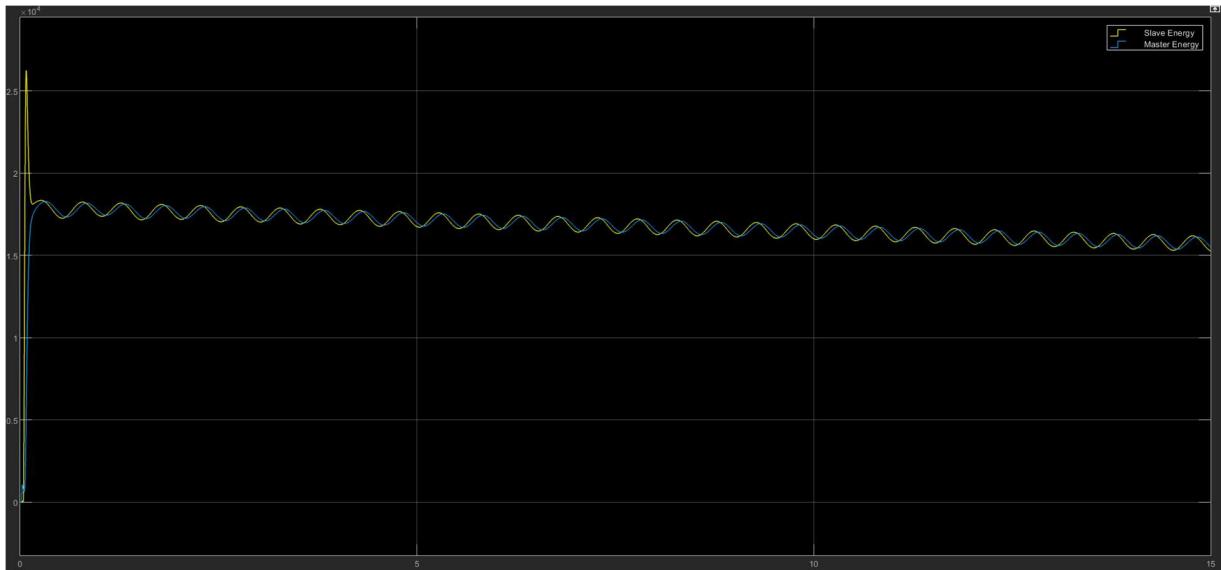


Figure 94 Energy levels of master and slave robot

As shown in Figure 70, the slave robot's position is limited by the environment and does not match the master robot's position. This is consistent with the results in Section 3.2, which indicate that the teleoperation controller's output is not affected by the delay. The velocity graph in Figure 71 also shows a similar discrepancy between the slave and master robots when the slave robot encounters the environment. In addition, Figure 72 displays the Force graph of both robots, where the master robot feels the force from the slave robot and reacts to minimize its impact.

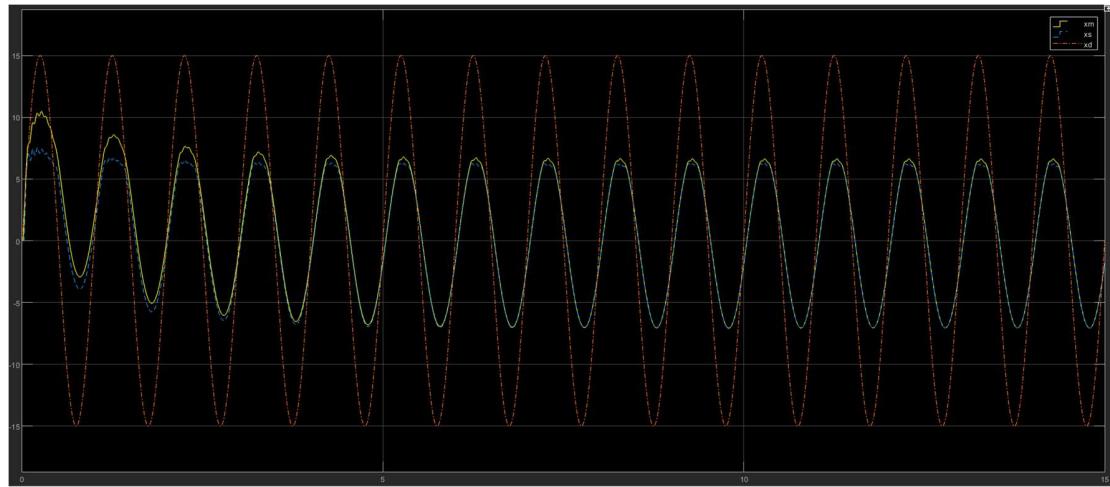


Figure 95 Position graph of the tank-based architecture in constraint motion.

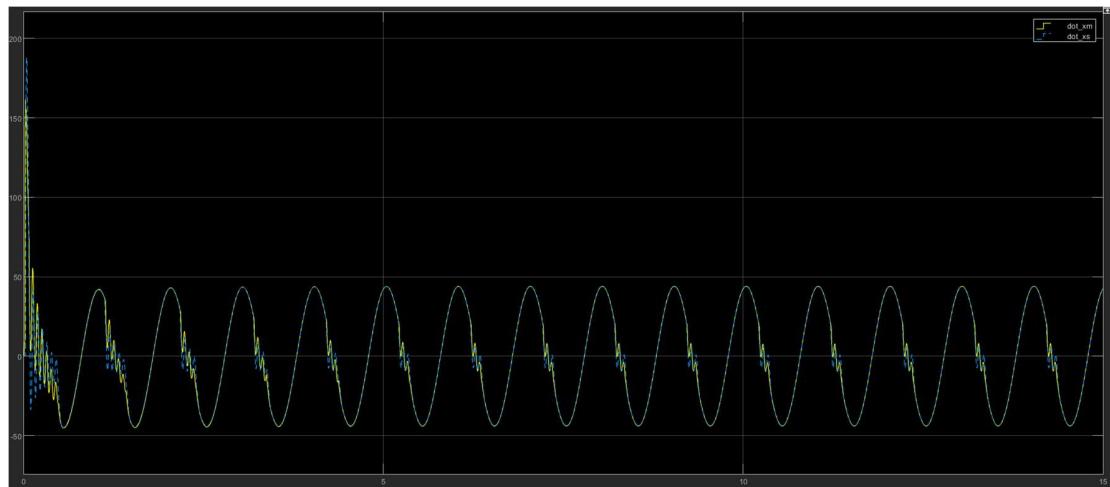


Figure 96 Velocity graph of the tank-based architecture in constraint motion.

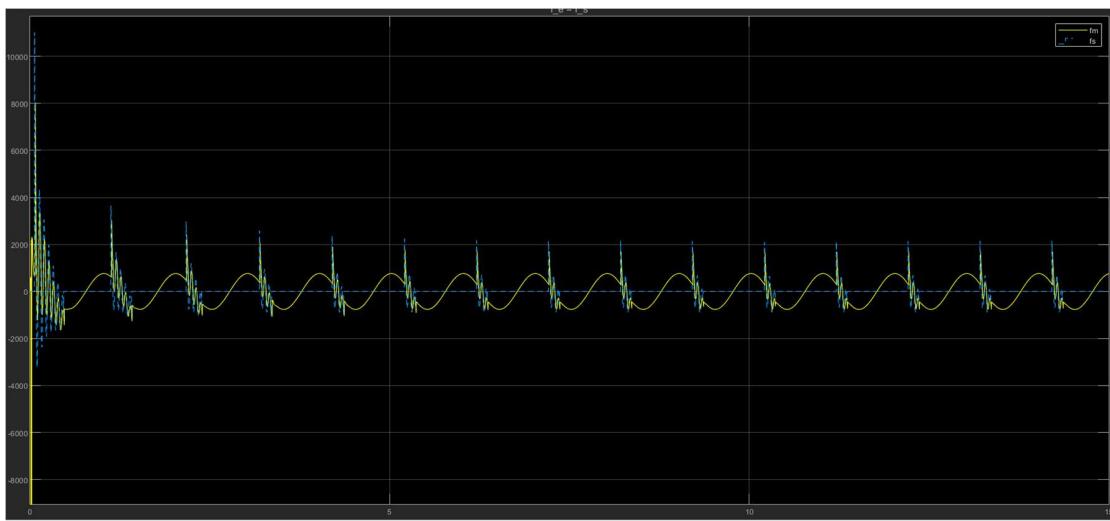


Figure 97 Force graph of the tank-based architecture in constraint motion.

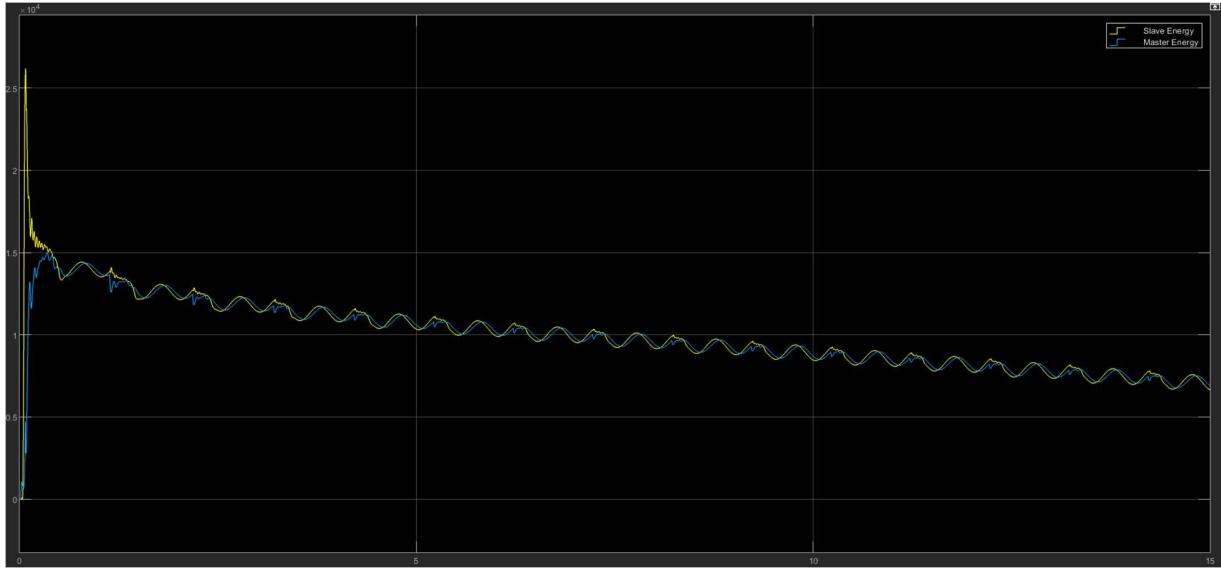


Figure 98 Energy levels of master and slave robots.

8.2.Position-Position (P-P)

A modified version of the bilateral force position architecture from Section 3.1 is presented in this sub section. This version uses the tank-based approach for bilateral teleoperation, which is applied to the communication channel. The rest of the components are the same as in the original architecture, which uses full transparency. The system is discretized with a sampling time T_s . The aim of this modification is to test the performance of the tank-based approach for bilateral teleoperation.

Figures 99-102 show the graphs of position, velocity, and force. The graphs reveal that the slave robot follows the master robot's position precisely, even with a large delay of 10 steps. This

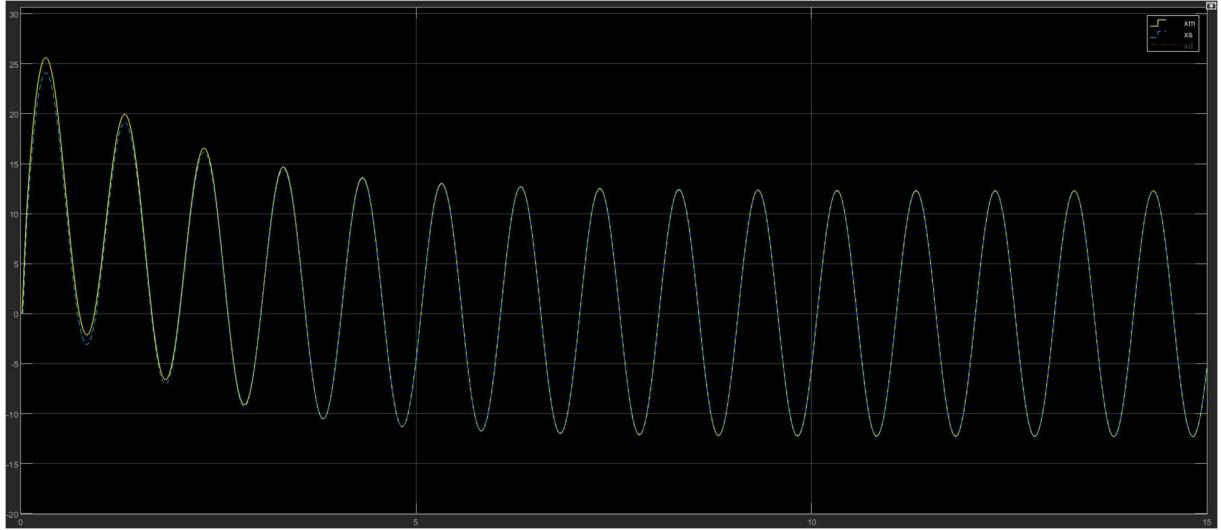


Figure 99 Position graph of the tank-based architecture in free motion.

implies that the packets received by both robots are 10 steps old, but the tank-based teleoperation architecture still ensures the convergence of the positions. The velocity graph also demonstrates a similar trend, as the slave robot quickly adjusts its velocity to match the master robot's. Furthermore, since there is no contact between the slave robot and the environment, its force is zero, as explained in Section 3.1.

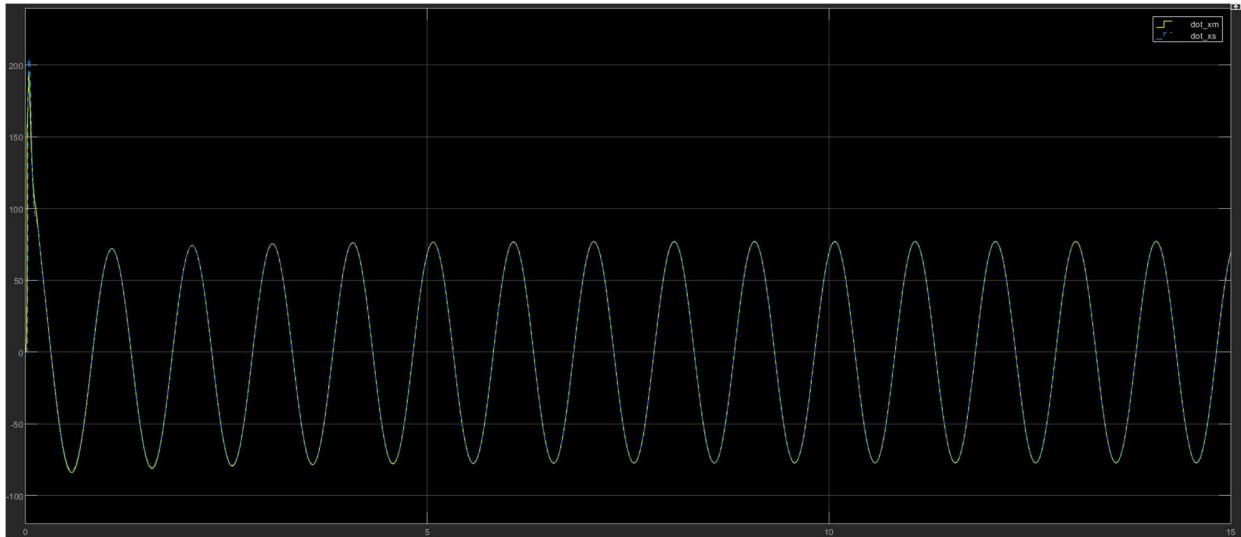


Figure 100 Velocity graph of the tank-based architecture in free motion.

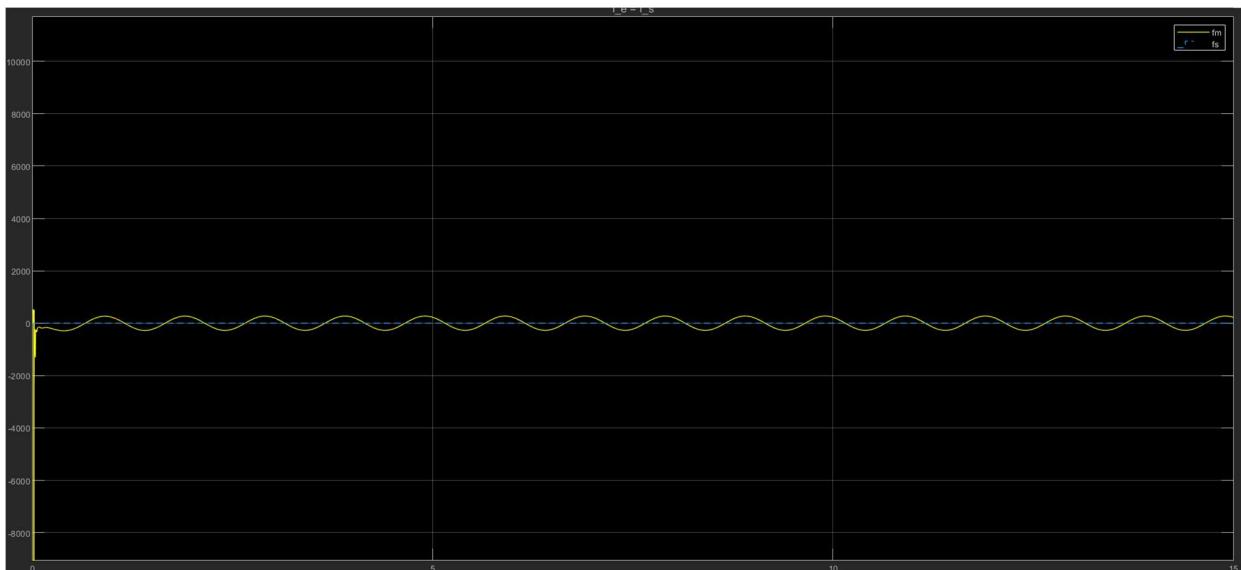


Figure 101 Force graph of the tank-based architecture in free motion.

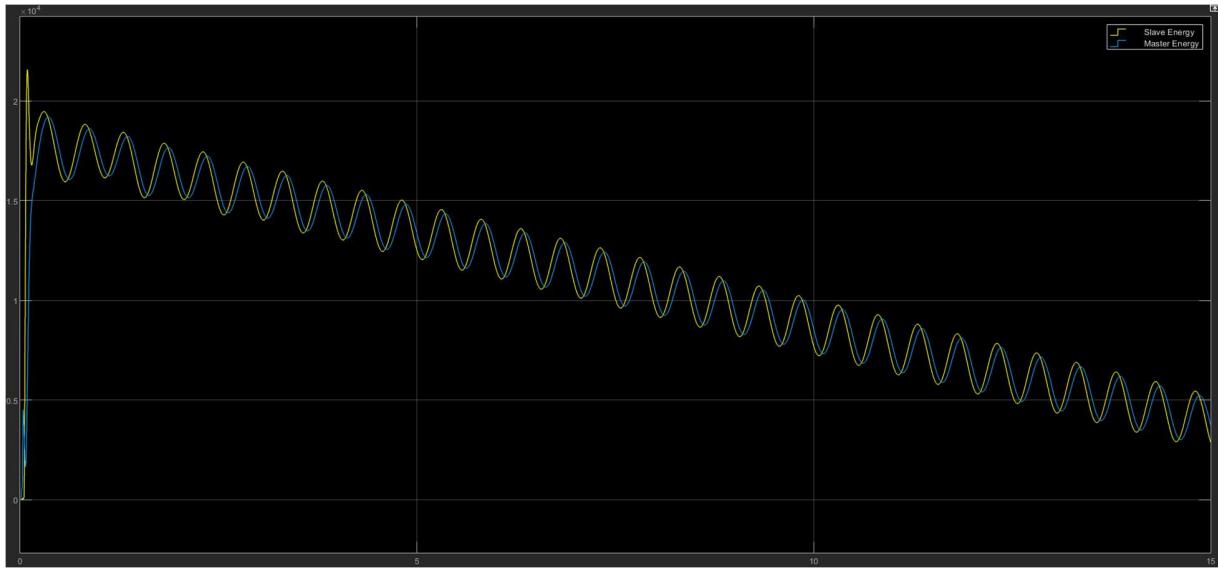


Figure 102 Energy graph of the master and slave robot.

9. Conclusion

Physical Human Robot Interaction and its applications in teleoperation is a hot topic among the researchers. Teleoperation refers to the remote control of a robot by a human operator, allowing the operator to perform tasks in environments that may be dangerous or inaccessible to humans. A detailed analysis of various teleoperation architectures, including Four-Channel Bilateral Teleoperation, Two-Channel Teleoperation, and Three-Channel Position-Force and Position (PF-P) teleoperation is conducted. These architectures are designed to enable effective communication between a human operator and a robot, allowing the operator to control the robot's movements and actions. Four-Channel Bilateral Teleoperation is an architecture based on four signals, two from the master side and two from the slave side. These signals are in the form of force and velocities of the slave and master robot. The goal was to design a controller and teleoperation channel for the slave and master robot that achieves transparency in a master-slave robotic system. As from the results perfect transparency was achieved. Two-Channel Teleoperation uses two communication channels for communication, one for transporting signals from the master robot to the slave robot, and another for transporting signals back from the slave to the master robot. The type of signal can be force or position for each channel and depends on the architecture of the teleoperations. Three-Channel Position-Force and Position (PF-P) teleoperation uses three signals for teleoperation, including force and position. The position signal is sent by the master robot to the slave robot while the force and position signal is sent to the master robot by the slave robot during teleoperation.

The use of Kalman Filter/Predictor and Kalman Smoother in teleoperation is discussed in this literature. These techniques are used to improve the accuracy of position, velocity, and force estimation in teleoperation systems. The Kalman Filter/Predictor is a mathematical algorithm that uses a series of measurements observed over time to produce estimates of unknown

variables that tend to be more accurate than those based on a single measurement alone. The Kalman Smoother is like the standard Kalman filter but uses all available data to smooth previous estimates. Also, the scattering wave architecture and tank-based architecture with different two channel architecture is implemented to eliminate the delays in communication channels.

Overall, the article provides valuable insights into the field of Physical Human Robot Interaction and its applications in teleoperation. The results demonstrate the effectiveness of various teleoperation architectures and techniques, providing a solid foundation for future research in this field.

References

- [1] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, ‘An atlas of physical human–robot interaction’, *Mech. Mach. Theory*, vol. 43, no. 3, pp. 253–270, 2008.
- [2] P. F. Hokayem and M. W. Spong, ‘Bilateral teleoperation: An historical survey’, *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.
- [3] S. Lichiardopol, ‘A survey on teleoperation’, *Tech. Univ. Eindh. DCT Rep.*, vol. 20, pp. 40–60, 2007.
- [4] D. A. Lawrence, ‘Stability and transparency in bilateral teleoperation’, *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 624–637, Oct. 1993, doi: 10.1109/70.258054.
- [5] S. E. Salcudean, M. Zhu, W.-H. Zhu, and K. Hashtrudi-Zaad, ‘Transparent Bilateral Teleoperation under Position and Rate Control’, *Int. J. Robot. Res.*, vol. 19, no. 12, pp. 1185–1202, Dec. 2000, doi: 10.1177/02783640022068020.
- [6] ‘Stable Adaptive Teleoperation | Semantic Scholar’. <https://www.semanticscholar.org/paper/Stable-Adaptive-Teleoperation-Niemeyer-Slotine/73767c873c49572df627a446d0ca92be434e3329> (accessed Apr. 10, 2023).
- [7] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, ‘Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparency’, *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 741–756, Aug. 2011, doi: 10.1109/TRO.2011.2142430.