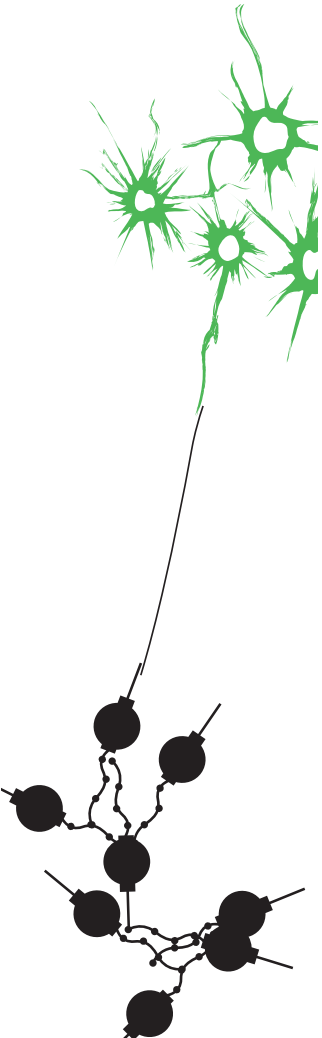


RAM ROBOTICS AND MECHATRONICS

MODEL-MEDIATED TELEOPERATION OF A FLEXIBLE OBJECT WITH A GRIPPER-MOUNTED TELEROBOT

A.S. (Akhilesh Singh) Naurathae

MSC ASSIGNMENT

Committee:
dr. ir. D. Dresscher
C. van der Walt
dr. ir. R.G.K.M. Aarts

January, 2022

002RaM2022
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Contents

Model Mediated Teleoperation Of a Flexible Gripper Mounted Telerobot

Akhilesh Singh Naurathae

November, 2021

***Abstract*- This paper introduces a novel concept of breaking flexible objects dynamics into two sub-models. One for approximating the grasping (contact) force and the second for approximating the forces felt due to motion after the object is grasped. This involves designing model-specific estimation techniques in an MMT setup. The sub-models are approximated at the master end and the estimation is carried out at the robot end. The results obtained are comparatively good at high frequencies for the inertial model. However, the flexible model is not descriptive enough to capture the object dynamics.**

Keywords- Flexible model, inertial model, MMT, RLS estimation

1 Introduction

Human beings interact with the objects present in the surrounding environment with the help of two extensively used senses, "vision" and "haptics". This helps them identify and/or alter the location of an object present in their environment. This ideology is applied in various "teleoperation" systems. The major goal of a teleoperation system is to transport a human operator virtually in sensation to a remote environment. Teleoperation systems allow humans to perform complex dexterous tasks remotely. Thus finding its applications in difficult and dangerous fields e.g. in the nuclear industry for the remote manipulation of radioactive hazardous environments, in space robot manipulation, subsea level exploration, medical teleoperation, etc.

A typical teleoperation system consists of a master system, slave system, and a communication channel connecting the master and slave. For stable teleoperation, the slave end must track the motion data sent by the teleoperator at the master side accurately. And the interaction force between the slave and the remote environment should be rendered to the operator accurately. In such a setup, the communication channel over which the motion and/or force data is sent introduces inevitable time delays. This affects the system's stability and transparency[1]. Many methods have been introduced over the years to tackle performance issues due to time-delayed communication. All of these methods suffered from a contradictory trade-off between system stability and transparency [1]. This means that stability would be achieved at the cost of transparency and vice-versa. Model Mediated Teleoperation (MMT) tries to solve this contradictory issue to some extent. In the MMT approach, a local model is employed on the master side to approximate the slave environment [1]. This local model receives the model parameters of a remote environment continuously. These parameters are estimated in the slave environment in real-time and are transmitted back to the master end. After receiving the updated parameters, the local model at the master end is updated and the force is rendered to the operator. Hence, haptic feedback can be achieved without any noticeable delay. This is achievable if the parameters correctly describe the remote environment model and the model does not change

drastically over time.

Many Model-Mediated Teleoperation systems present in the literature treat the interaction between telerobot and environment as an interaction at a specific point. However, dexterous interaction involves the use of hands which interacts with the environment at several points. Human beings feel sensations at different parts of their hand while manipulating objects. Intuitively speaking, the gripping sensation is felt by fingers and palm, while the object manipulation is more dominant at the wrist. The gripping sensation is also felt during manipulation. The hypothesis for this research work is that an object's dynamics can be broken down into two different dynamic sub-models: one for approximating the gripping force and the other for approximating the forces felt due to object manipulation. The gripping/contact dynamics model is called the flexible dynamics model or the flexible model. And the object manipulation model is called the inertial dynamics model or inertial model. We plan to estimate flexible model parameters using data from the sensors mounted on a gripper attached to the telerobots end-effector, and the inertial model parameters using motion and force data from the end-effector. Due to splitting an object's dynamics, there can be performance issues in model-specific estimations. One sub-model can introduce parasitic effects in the other and/or vise-versa. This intuition leads to the following research question:

- What are the types of parasitic effects associated with flexible and inertial dynamics and to what extent do they hinder system performance?

2 Related Work

In practice, several methods have been designed for bilateral teleoperation systems. These methods provide a robust and stable haptic rendering under considerable time-delayed communications. The methods described in [2,3,4,5,6] are passivity-based approaches that incorporate the use of wave–variable transformations and time domain passivity control schemes . These methods are prone to a trade-off between system stability and transparency.

Model Mediated Teleoperation (MMT) provides significant improvements over the classical approaches by addressing both the stability and transparency issues. In the MMT approach, approximate model of the slave environment is employed at the master end and haptic feedback is obtained based on this local model. Hence, if the approximated local model is accurate both stable and transparent bilateral teleoperation can be achieved[1]. However, MMT has its own set of challenges: accurate object modeling and timely local model updates with respect to the changing environment [1]. This determines how effective is the designed MMT architecture.

A variety of MMT architectures exist in literature that makes use of different modelling structures and environment types. In [7] Huang et al proposed a recurrent neural network (RNN) – based predictive control scheme for a position-position teleoperation architecture. This RNN estimator is able to model the non-linear behavior of the slave environment. The non-linear slave-environment behavior is assumed to contain an invariant linear part. This invariant linear part is used on the master side to predict the slave behavior [7]. However, this method is suitable for systems with invariant time delays and gives performance issues under varying time delays. An improved version of this control scheme was obtained by replacing RNN with a neural network (NN). It provided good results under delay and time-varying environments[8]. Its performance is limitted due to larger training time and is computationally heavy. In [9] Tzafestas et al proposed an impedance reflection scheme. His work was improved further by Verscheure et al. in [10], and Xu et al. in [11] to support multiple degrees of freedom, cope with time-varying environments and geometric uncertainties. However, complex geometric models introduce additional parameters which can affect the system performance[9]. Xu et al proposed a point cloud based MMT (pcbMMT) in [12], which uses

a ToF camera to capture a high-resolution point cloud model of the surface of the object, and the physical properties of the object are measured at the slave side and is sent to the master side. Hence, making this approach suitable for approximating complex and uncertain geometric environments.

All of the methods mentioned above follow a generic flow. where the slave environment is approximated at the master end. However, these methods have different techniques for identifying the slave environment. This depends upon if the slave environment model is known or unknown. For unknown environments, no model assumption is made. Instead, there is a direct mapping between the input and the output for the slave environment. Hence, non-parametric methods can be used for uncertain environments [1]. The works mentioned [7,8] are based on non-parametric methods, If the slave environment is known or assumed based on pre-knowledge, parametric methods can be used [1]. This involves an online estimation of the model parameters. The works mentioned in [9,10,11,12] are parameter-based methods. In parametric methods, several environment models have been proposed based on the object's geometry and physical properties. The models can be linear or non-linear. Models like elastic models, Maxwells model, Kelvin-Voigt model, and Kelvin-Boltzmann model are linear models[20]. These models are composed of linear springs and linear viscous dampers arranged in different configurations. The Kelvin-Voigt model is a very popular linear model[21], with spring and damper arranged in parallel. However, the Kelvin-Voigt model or other linear models tend to show noticeable inconsistencies for flexible objects[22]. To account for this the non-linear models like the Hunt-Crossley model, Quasi-linear model, point cloud based were proposed. The Hunt-Crossley model is an extension of the Kelvin-Voigt model[21]. In this model, the spring and the viscous damper terms are expressed by exponential displacement, which considers the contact geometry. It has been proven in [21] that the Hunt-Crossley model works well with the non-linear objects. The Quasi-linear model is accurate for off-line estimation. But it is complex and difficult to realise for online estimation [22].

There are no modeling schemes in literature that separate a single object dynamics into two sub-dynamic models (Section 1). In this research work, we will use the Hunt-Crossley model due to its simplicity to represent our flexible dynamics model. The parameters of both of our models are estimated using Recursive Least Squares (RLS) estimator. More complex estimators exist in practice. We chose RLS estimator due to its simplicity. The goal of this research is to design model-specific estimators, validate our hypothesis, and identify the effects of mutual/parasitic dynamics between the two models.

3 Method

In this section, we provide a detailed explanation of the approach and design concepts related to our research work. Firstly, a few assumptions were made regarding the object with which the slave will be interacting. The remote environment consists of a flexible object grasped by the gripper. The following assumptions were made:

- The object with which the slave will interact is isotropic, i.e the object's properties do not vary with direction. Hence the force measured by the grippers in any direction will have the same value. This assumption will help us to simplify our flexible dynamics model.
- The gyroscopic effects are assumed to be negligible due to low velocities.
- The effect of gravity is included and is added to the inertial dynamics model.

To re-iterate the flexible dynamic model is used to describe the contact and gripping of the object by the slave . An inertial dynamic model is used to describe the end effector forces after gripping has taken place. Therefore, our MMT system consists of two estimators at the slave side which synchronously estimates model specific parameters. The inter-

action forces are calculated from the individual local models at the master side based on these estimated parameters and sensor data.

This section involves a sub section explaining the MMT approach. We provide a detailed explanation on flexible and inertial dynamics modelling (Section 3.3.2 – 3.3.3). Finally, the complete system is explained in Section 3.3.4.

3.1 Model Mediated Teleoperation (MMT)

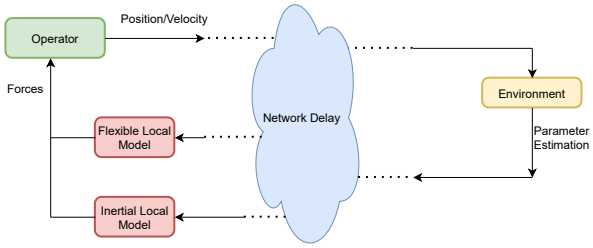


Figure 1: Process Flow for our MMT architecture.

In our research, we follow the MMT flowchart as shown above. Instead of creating a single model of an object, the object dynamics are modelled into two different dynamic models called as the flexible dynamics model and the inertial dynamics model.

The operator inputs a certain motion/position command which is sent to the slave end via a tracking controller. At the slave end environment parameters are estimated which are model specific.These estimated parameters are sent to their respective local models at the master end to compute the estimated forces. These forces are then sent back to the operator. The models and estimation algorithms for flexible and inertial models are explained in the following sub sections.

3.2 Modelling

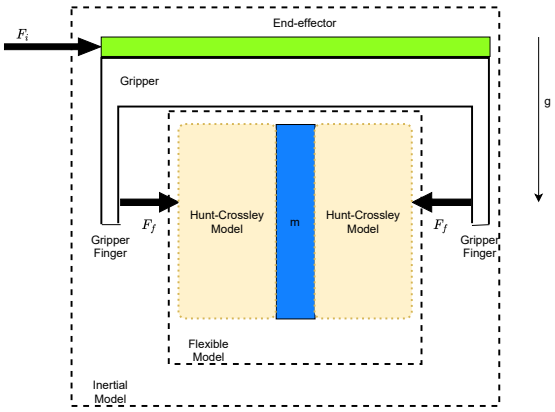
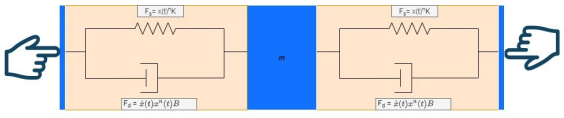


Figure 2: Complete Dynamic Model

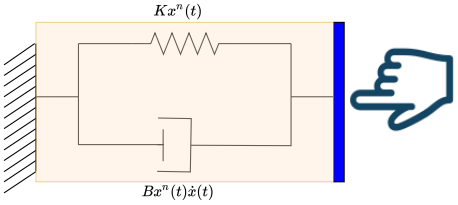
The above figure represents the complete model of our remote environment. The flexible model consists of two Hunt-crossly models. These models are arranged as shown in figure 2. This arrangement will help us identify the grasping force applied by the gripper fingers. This force is measured by using a force/torque sensor mounted on the gripper. Also, the gripper force is independent to the force applied by the end-effector of the robot arm. The inertial model is represented by the exterior dotted box. This indicates that the inertial model requires the object to be in the grippers grasp before estimating the model parameters. The force is measured using the sensors present at the end-effector. The reason for such a model is to study the effect of any parasitic effects as we manipulate the object in free space. Since we are manipulating the object at low velocities, there is a possibility to encounter parasitic effects at high frequencies. This will be explored more later.

The subsections below provides more details and depth to individual model design.

3.2.1 Flexible Dynamics Modelling



(a) Grasping of a flexible object



(b) Equivalent model

Figure 3: Flexible model for two finger gripper interacting with a flexible object.

The Hunt – Crossley Model is a non-linear model as explained in section 2, and is well suited for modelling most of the real environments[1], and can be represented mathematically as,

$$F(t) = \begin{cases} Kx^n(t) + Bx^n(t)\dot{x}(t), & \text{if} x(t) \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Where K is the stiffness, B is the damping, x(t) is the penetration by the gripper finger, n is a constant which typically lies between 1 and 2 [15]. Figure 3(a) shows the grasping of a flexible object using two Hunt-Crossley models. The idea behind using two Hunt-Crossley models is to represent the object being grasped by the two-finger gripper. Here each Hunt-Crossley model approximates the interaction force between the flexible object and the respective finger. Since we assume our object to be isotropic, the object's parameters are the same when measured in any direction i.e. K, B and n are constant when measured from all directions. Along with isotropic object assumption, both the fingers should deform the object equally (i.e. $x_1(t) = x_2(t)$). This implies that the force applied by one finger is equal in magnitude and opposite in direction to the force applied by the second finger. With these assumptions our model can be further simplified and can be reduced to a single model as shown in Figure 3(b).

Based on the Hunt-Crossley model the slave contact force can be modelled locally and can be calculated using equation (1),

$$F(t) = Kx^n(t) + Bx^n(t)\dot{x}(t) \quad (2)$$

The non-linear relationship in equation (2) can be linearised by taking natural logarithm on both sides [15]. This yields the following equation,

$$lnF(t) = ln(K) + nln(x(t)) + \frac{B\dot{x}(t)}{K} \quad (3)$$

The above equation is linear and can be solved linearly to minimise the error between the measured and the estimated force.

3.2.2 Inertial Dynamics Modelling

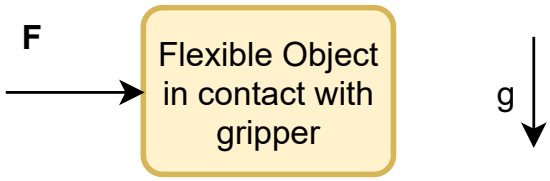


Figure 4: Inertial Model once the slave is in contact with the deformable object.

Inertial dynamics come into the picture once the robot gripper has grasped the flexible object. The effects due to friction are neglected as the object is being manipulated in free space and not along any horizontal surface. The gyroscopic effects and Coriolis effects are neglected as object manipulation is carried out at low velocities. While moving an object in free space action is performed against gravity and hence the gravitational effect is added to our inertial model. Also in free space environmental damping is negligible and hence is not considered in our model. From the assumptions and considerations of the physical effects mentioned above, the inertial model can be described by the following equation and is represented as in figure 4:

$$\vec{F_i} = m\vec{a} + m\vec{g} \quad (4)$$

Where $\vec{a}$ is the end effector acceleration. It is the same as the object acceleration after grasping has taken place. The slave velocity signals can be used from the end effector for calculating the acceleration.The parameter $\hat{m}$ is the estimate. Applying it in equation 4, the estimated force F_i can be calculated. Equation 4 is linear and can be solved linearly to minimise the error between the measured and the estimated force.

3.3 Parameter estimation using RLS estimator

To estimate the dynamic parameters, we make use of RLS estimator as explained in [15] for both the flexible and inertial models for estimating the model specific dynamic parameters.

Equation (3) and equation (4) can be re-written in linear form as,

$$y_f = \phi_f^T \theta_f \tag{5}$$

$$y_i = \phi_i^T \theta_i \tag{6}$$

Where,
$y_f = ln(F)$, $\phi_f^T = [1, \dot{x}, ln(x)]$, $\theta_f = [ln(K), \frac{B}{K}, n]^T$ for the flexible model and $y_i = F_i$, $\phi_i^T = a + g$, $\theta_i = m$, for the inertial model.

Here in general terms, ϕ^T is called the regressor vector, and θ is called the parameter vector. After defining and linearising the models, parameter estimation is performed using the recursive least squared(RLS) parameter estimation algorithm. The estimated model parameters are $\hat{K}$, $\hat{B}$, $\hat{n}$ for the flexible model and $\hat{m}$ for the inertial model. Following calculations are iterated for both the models separately to achieve convergence:

$$L_{k+1} = \frac{P_k \phi_{k+1}}{\lambda + \phi_{k+1}^T P_k \phi_{k+1}} \tag{7}$$

$$P_{k+1} = \frac{P_k - L_{k+1} \phi_{k+1}^T P_k}{\lambda} \tag{8}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + L_{k+1}(F_{k+1} - \phi_{k+1}^T \hat{\theta}_k) \tag{9}$$

Where, L is the adaptation matrix, P is the covariance matrix, $\hat{\theta}$ is the estimated parameter vector. The model specific estimated parameter vectors along with their respective regressor vectors are then used in the local model at the master side to compute the interaction force as per equation (5) and (6).

3.4 Complete System

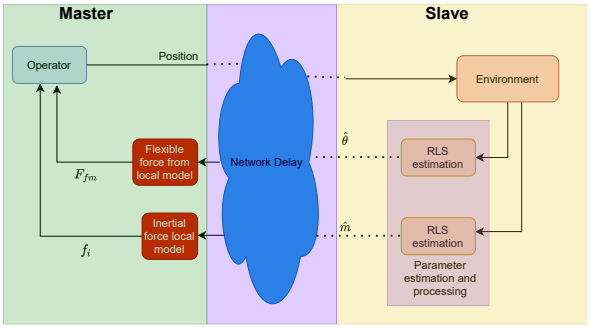


Figure 5: Complete System Flow.

The complete system flow is as shown in figure 5. The operator at the master end inputs an position command for the telerobot to contact the object.

The estimation begins when the object is in grasp of the gripper and is in free space. The measured position, velocity and force of the end-effector is sent to the inertial model RLS estimator. The gripper force, position and end-effector velocity is sent to the flexible model RLS estimator. The RLS estimators provides model specific parameter vectors. These parameter vectors are sent to their respective local models. Using the estimated parameter vectors and regressors model specific force is obtained.

4 Results and Discussions

This section explains the practical implementation of the complete system (Section 3.5) and the validation of our method by the means of an experiment.

4.1 Experimental Setup

The experimental setup involves a haptic devic (Omega 7), a robot arm (Franka Emika) and a flexible object. The communication between the two devices was implemented in ROS by the means of communication nodes (Appendix D.1.1), where every node is task specific. The user inputs a position command via the omega 7, which sends position information to a position tracking controller to ensure that the robot follows the position commands. The

ROS system flow is as shown in Figure 6.

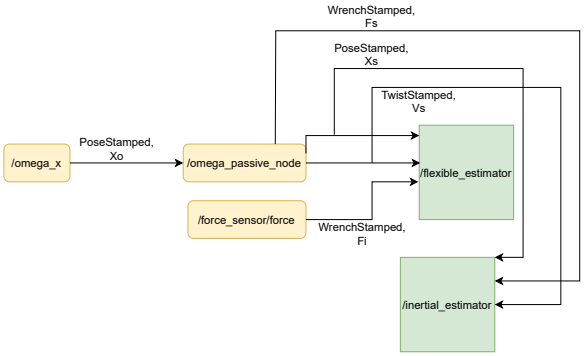
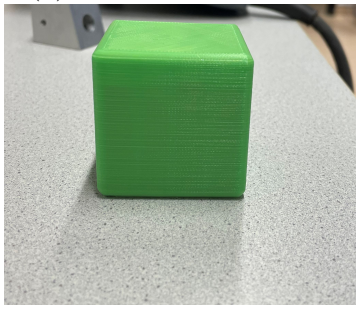


Figure 6: ROS System Flow.

A force torque sensor ATI-mini 40 [16] is attached to one of the gripper fingers to measure the interaction force with the flexible object. The robot interacts with a flexible object weighing 305 grams and is made of Dragonskin 30[17] (Figure 7). The interaction force from the force torque sensor from the gripper finger after the gripper has grasped the flexible object and the position and velocity of the end-effector are sent to the flexible model RLS estimator and the end-effector acceleration is sent to the inertial model RLS estimator for model specific parameter estimation.

(a):Flexible object used



(b):Rigid object used

Figure 7:Objects used

4.2 Experiment

The purpose of this experiment is to validate our method and to answer our research question mentioned in section 1. We are not detecting contact, instead we aim to determine the correctness of our models during object manipulation in free space. Therefore, before beginning the experiment the flexible object is grasped by the gripper and is suspended in free space (Appendix E: Figure E.2). The grippers are opend and closed using keys on the key board. The operator inputs sinusoidal position commands via the handle of the haptic device along the x-y plane. Over the course of the experiment the system is excited at different frequencies. The initial values for the flexible model RLS estimator were set as K = 50 N/m, B= 5 Ns/m, n=1.3 and for the inertial model RLS estimator the initial value used was m = 0.305 kg. It should be noted that, it takes nearly 10.56 seconds for the parameters to converge to a steady value (Appendix E: Figure E.7).

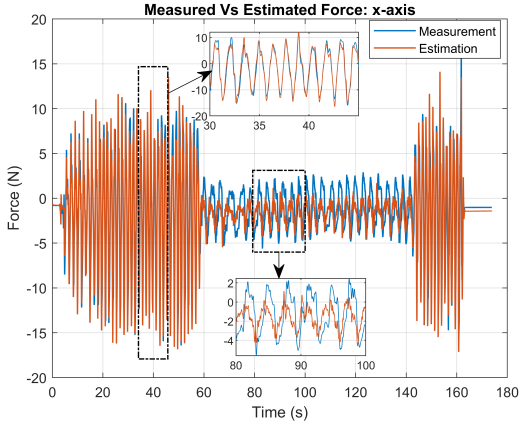


Figure 7:Measured and estimated force along x-axis for inertial model

Figure 8 shows the estimated and measured force along x-axis over time in seconds for the inertial model. The tracking is relatively good at higher frequencies and the estimation matches the measurement. This can be seen in the magnified plot from t=30s to t=45s. In this range, the relative error is approximately between 2-9%. And the mean tracking error is 2.1% of the RMS value of the measured force, which is good.

The tracking is poor at low frequen-

cies, and the inconsistencies in tracking is clearly noticeable in the plot between t=60s to t=140s. The tracking follows the measured force but doesn't match with the magnitude of the measured force (t=80s to t=100s magnified plot). In this range between t=60s to t=140s, the mean tracking error is 20.41% of the RMS value of the measured force. And the relative error can reach upto 100%. This is undesirable behaviour.

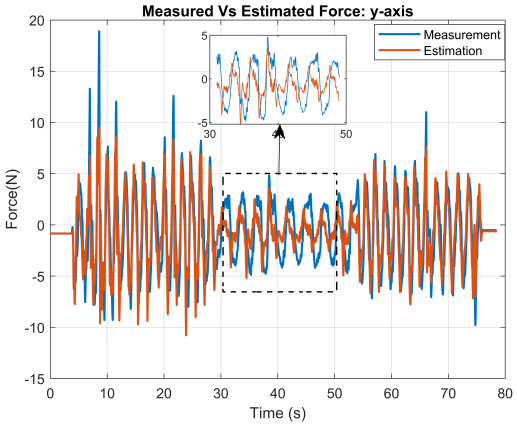


Figure 8:Measured and estimated force for rigid object along y-axis for inertial model

We expected poor tracking at higher frequencies. This is not the case based on the obtained results. The flexibility of the object does not introduce parasitic effects at high frequencies. To confirm this, the same experiment was carried out using a rigid object (Figure). The results obtained along y-axis can be seen in figure 9. The results indicate similar performance, where the tracking is poor at low frequencies (Figure 9: t=30s to t=50s magnified plot). However, we do not know the reason for poor performance at low frequencies, and more research is needed for the same.

The obtained results indicate that the inertial model is descriptive enough to capture inertial dynamics of flexible object manipulated in free space at high frequencies.

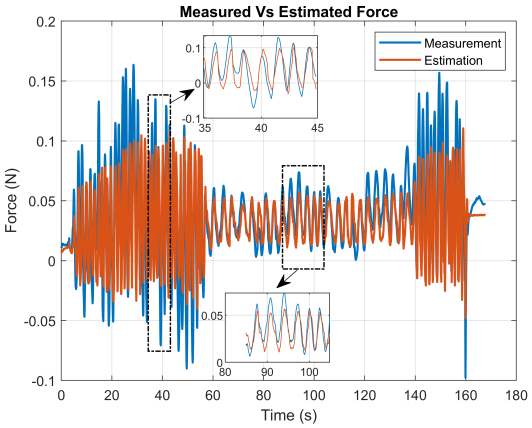


Figure 9:Measured and estimated force for flexible model

The measured and estimated forces over time for the flexible model is as shown in Figure 10. The tracking is poor at high frequencies and comparatively good at low frequencies. This can be clearly seen in the magnified plots from t=35s to t=45s, and from t=85s to t=110s. The tracking follows the measured force, but the force estimated using the flexible model parameter estimation does not match the magnitude of the measured force. This is undesirable behaviour.

At high frequencies (t=20s to t=45s) the mean tracking error is 18% of the RMS value of the measured force, with relative errors reaching to 100%. Similarly, at low frequencies (t=60s to t=100s) the mean tracking error is 7.4% of the RMS value of the measured force, with relative errors reaching to 96.33%. Also, the model parameters do not converge to a steady value and are inconsistent (Figure 11, 12, 13).

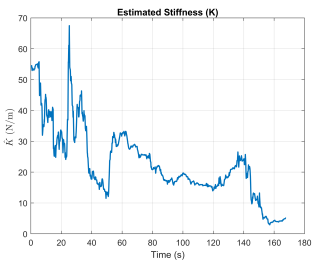


Figure 10:Measured and estimated force for flexible model

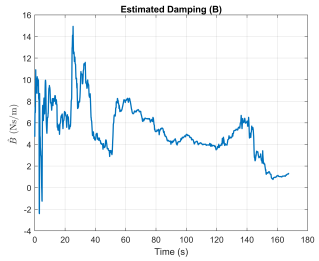


Figure 11:Measured and estimated force for flexible model

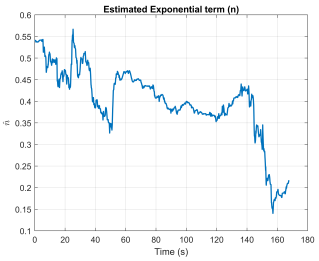


Figure 12:Measured and estimated force for flexible model

This indicates that our flexible model is not descriptive enough to capture the flexible objects dynamics. Also, the Hunt-Crossley model is a contact dynamics model which purely mimics the penetration. Intuitively speaking, the model does not consider the motion of the entire object as a whole and the level of penetration changes quickly at higher frequencies during free motion.This can be a reason for poor performance. In our experiment we are manipulating the object freely in space. There is a possibility that the objects mass and gravity also attribute in poor performance of our estimator. These are our intuitions and more research is needed to confirm or deny the same.

5 Conclusions and Future work

The goal of our research work is to prove or disprove the hypothesis of breaking an object dynamics and to identify the parasitic effects associated with both the models by designing model specific estimators. The performance of inertial model is better than the performance of flexible model at higher frequencies. However the flexible model suffers from inaccuracies at high and low frequencies, failing to match the peaks of the measured force. Therefore, the flexible model described in this work is not suitable for this work. Besides these problems, inertial model works as expected and is able to capture the object dynamics very well at high frequencies. This proves that it is possible to break an objects dynamic model into two sub-models, but more research is required for modelling the flexible dynamics.

This work can be improved further by investigating different models for modelling flexible dynamics. This can improve the tracking and the magnitude of the estimated force can nearly match that of the measured force. Also, since we mainly focused on the estimator design, future work may involve rendering these forces on a haptic device. A device which is capable of rendering forces from both the models to two different locations on a single haptic device. This would give a sense of force feedback to the operator at the wrist and the fingers.

6 Acknowledgements

I would like to thank my thesis supervisor Ir. Christophe van der Walt for his expert guidance and professional insights which helped me formulate my research questions. I would like to acknowledge his support, critical feedback and understanding which helped me throughout the course of my thesis and kept me in the right direction.

I would also like to thank my thesis chair Dr. Ir. Douwe Dresscher for giving me the wonderfull opportunity to carry out my thesis under his project group.

In addition I would like to thank the RaM (Robotics and Mechatronics) group for providing a wonderfull working atmoshpere during the practical part of my thesis for conducting experiments.

A Appendix: Model Mediated Teleoperation

Traditional bilateral teleoperation systems often suffered either from stability issues or transparency issues. This means that the system stability was achieved at the cost of system transparency and achieving a transparent operation affected the system stability. Model Mediated Teleoperation (MMT) was developed to tackle this contradicting issue between system stability and transparency. Impedance reflection systems were the first to incorporate the logic of MMT, in such systems the impedance parameters where estimated and then transmitted from the slave end over a time delayed communication network to the operator end and these impedance parameters where then used locally at the master end to compute the force. In the MMT approach the object model at the slave side is approximated at the master side, estimation algorithms provide estimates of the model parameters at the slave end, these estimated parameters are then transmitted over the time delayed communication network. These parameters are then used in the local model to reproduce the force, theoretically providing no delay in the force rendered to the operator as the force is being computed locally at the master end.

In this research, the dynamics of a single flexible object is broken down into two different dynamic models called the flexible model and the inertial model. The purpose of the flexible model is to provide an estimation of the contact forces when the gripper grasps the flexible object and the purpose of the inertial model is to approximate the forces due to the free motion of the flexible object once the flexible object has been grasped by the gripper.

A.1 Stability and Transparency

In an MMT architecture it is really important for the estimator to estimate the parameters accurately, inaccurate estimation can lead to stability issues. Also, the rate of change of parameters should be low. Transparency issues arise when the parameters change rapidly causing a mismatch between the local model and the dynamic environment.

Both of our models are designed based on certain assumptions mentioned in the specific model design sections (section 3), this means that our estimators will function as expected under certain frequency ranges. It is important to note that friction is assumed to be completely absent in our model design and hence dragging the flexible object on the horizontal surface may introduce unwanted behaviour due to friction. Also, exciting the system at higher frequencies in the range of GHz will also cause unwanted behaviour due to gyroscopic effect as it is also assumed to be absent in our model design. These parasitic effects may hinder the system stability and performance.

B Appendix: Parameter Estimation Algorithms

As mentioned earlier parameter estimation forms a major part of this research work. In our work we perform online parameter estimation at the slave side, these estimated parameters are sent to the master end for computing the interaction force from the local model at the master end. The parameter estimation technique used in this work is same for both the models but estimates model specific parameters, which means that they are designed to estimate parameters specific to a particular model.

B.1 Recursive Least Squared (RLS) algorithm for flexible model and inertial model parameter estimation

The aim of RLS algorithm is to minimize the mean square error, while finding the values of appropriate parameter variables which reduce the error to a minimum. For flexible model, the unknown parameters are the stiffness (K), damping (D) and an exponential term n and for the inertial model, the unknown parameter is the objects mass (m). The algorithm is said to have converged to the true values when the error between the estimated and measured force reduces to a minimum value. We are making use of a single stage RLS estimator which is capable of estimating all the three parameters (K, B and n) for the flexible model and the mass 'm' for the inertial model. The flexible model is designed based on the Hunt-Crossley model and hence forms a non linear relationship between force, velocity and position. This non-linear relationship has to be linearized. Taking natural logarithm on both sides of equation (2) gives,

$$ln(F(t)) = ln(Kx^n(t)(1 + K^{-1})Bx^n(t)\dot{x}(t)) \tag{10}$$

$$ln(F(t)) = ln(Kx^n(t)) + ln(1 + K^{-1}Bx^n(t)\dot{x}(t)) \tag{11}$$

$$ln(F(t)) = ln(Kx^n(t) + nln(x(t)) + ln(1 + K^{-1}Bx^n(t)\dot{x}(t) \tag{12}$$

Assuming that ln(a+1) $\approx$ a if and only if $|a| \ll 1$. Equation 14 boils down to,

$$lnF(t) = ln(K) + nln(x(t)) + \frac{B(t)}{K} \tag{13}$$

The algorithm for this is explained in section 3.4 and the process flow will be explained further.

C Appendix: The Complete System

The system architecture and the complete system flow is explained in this section

C.1 The environment

Before finalizing our model, few assumptions were made beforehand. The flexible object is assumed to be isotropic, physical effects like friction, Coriolis and gyroscopic effects are neglected (Appendix B.1.2). Our system consists of 2 dynamic models, 2 RLS parameter estimators, one for each model and one flexible object which is allowed to move freely in space.

The operator inputs a position command, which the Franka's end effector and gripper follow. At the franka end after grasping has taken place the information signals from the frankas end effector like force, position and velocity and from the gripper signals like position and force signals are used in the model specific RLS estimators to estimate the model parameters. These parameters are then stored in the parameter vector for individual models and are transmitted over the time delayed communication network towards the operator end. These estimaed parameters are then used in the local model at the operator end to compute the interaction force.

C.2 Estimator Process

This section explains model specific Estimation Processes.

C.2.1 Flexible model parameter estimation process flow

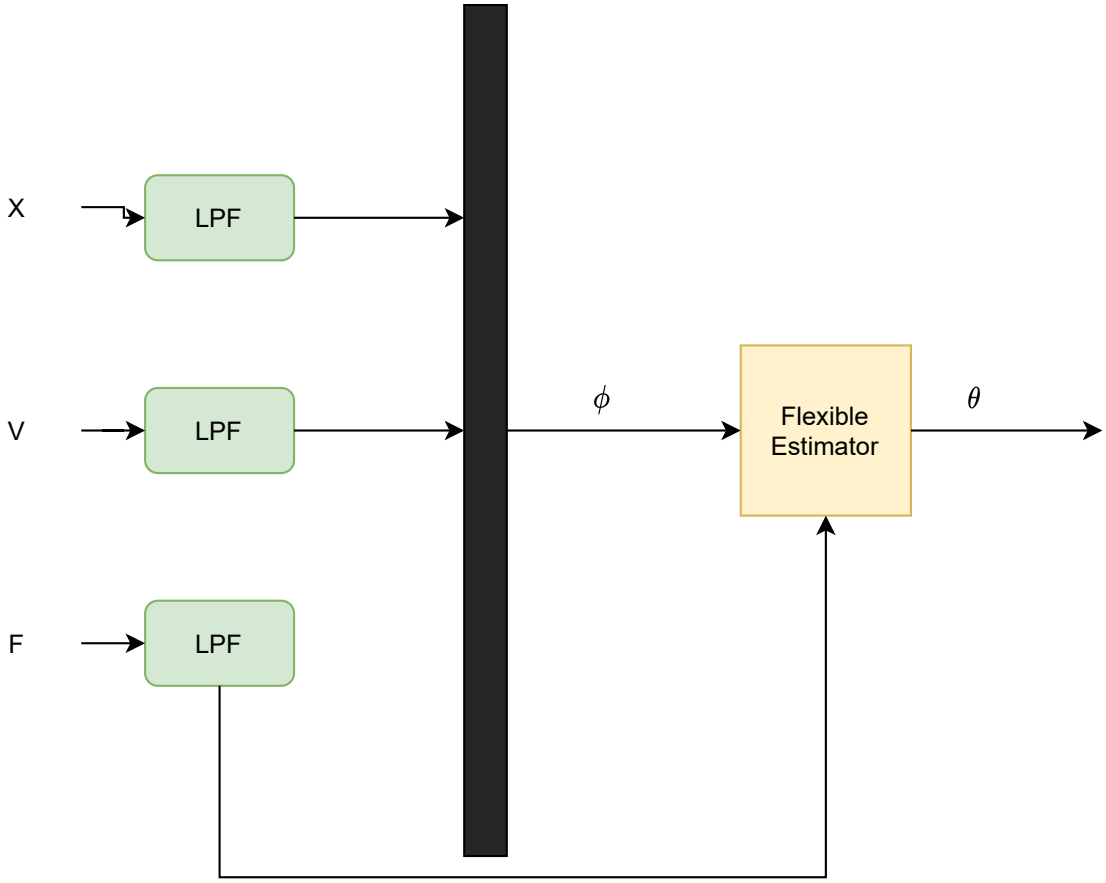


Figure C.1: Flexible Estimation Process Flow

In our work, we used the Franka Emika robotic arm as our telerobotic device. The end-effector of the Franka provides us with the end-effector force, position and velocity data and the gripper provides information about the gripper position and the force sensor attached to the gripper finger provides the gripping force information. The robots end-effector position and velocity are are required to from the regressor vector for parameter estimation. The position, velocity and force signals are passed through a simple low pass filter to reduce signal noise. The cutt-off frequency used was 10Hz. It is important to note that the cut-off frequency should filter sufficient noise and not distort the signals, if this happens then our estimator will produce distorted output which will cause undesirable behaviour. These filtered signals are concatenated to form the regressor vector ϕ. The regressor vector is then used in the RLS estimator which estimates the parameter vector θ. The parameter vector is a 3X1 vector consisting of log(K), $\frac{B}{k}$ and n terms. These parameters are then sent on by the estimator to compute the estimated force.

C.2.2 Inertial model parameter estimation process flow

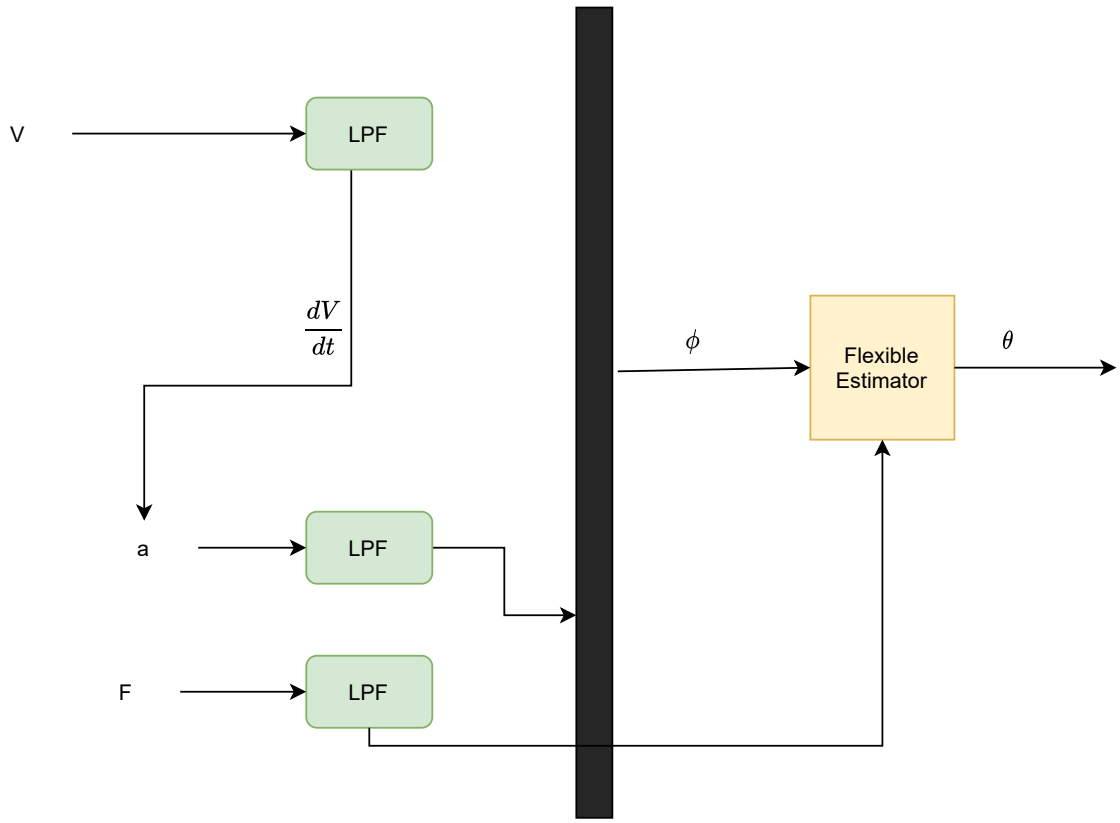


Figure C.2: Flexible Estimation Process Flow

The inertial model parameter estimator makes use of acceleration and gravity in a 2x1 regressor vector ϕ. The robots end-effector provides us with velocity information which is passed through a low pass filter and then differentiated to obtain the acceleration signal which is again passed through a low-pass filter to suppress noise. The force signals from the robots end effector are also passed through a low-pass filter to compensate for any phase lags. The regresessor vector is then passed to the RLS estimator which estimates the 2x1 parameter vector θ which comprises of the object mass m. The estimated parameter is then sent on by the estimator to computed the estimated force.

D Experimental Setup

This section provides more information about the software and the hardware used and the experiment setup.

D.1 Software

D.1.1 ROS Kinetic

The Robot Operating System (ROS) is a set of libraries and tools used for building robotic applications[19]. Over the years several different versions of ROS have come into picture. For our setup we are making use of ROS kinetic.

ROS establishes communication between different process to form a launchable single task specific application. Process here reffers to ROS nodes which are executable programs either in c++ or python. These nodes communicate by subscribing and publishing data with each other. A node can receive data from another node by subscribing to the node publishing the desired data.The data is subscribed ans/or published in the form of messages which defines a specific data format. To receive data, the message type of the subscriber must match that of the publisher.

D.2 Hardware

D.2.1 Omega 7

Omega 7 is a hapic device capable of imposing a 6 - DoF force on the handel of the device. It can also exert a 3 DoF output force on the handel [16].

Omega 7 is the input device, the user imposes a position command with certain force by the means of the handel, in the ROS environment this input is published as a message of type EndEffectorState which consists of message types Pose, Twist, Wrench. Pose message type consists of the end-effector position and orientation, Twist message type holds the linear and angular velocity data and Wrench message type holds the forces and torques. The EndEffectorState message type is then published by the means of omega_x node which is then subscribed by the omega_passive_node which is a passive controller for ensuring stable operation and to make sure the robots end-effector follows the input position command.

Figure D.1: ROS System Flow.

D.2.2 Franka Emika

The Franka Emika is a robotic arm having 7 - DoF [17], this will act as the telerobotic device in our system. The Franka communicates with the omega 7 via the omega_passive_node which

subscribes to Pose, Twist and Wrench information from the omega_x node and publishes the robots end-effector state which include the pose, twist and wrench. The end-effector state is subscribed by our RLS estimators for the desired signals.

Figure D.2: ROS System Flow.

D.3 Set up

At the beginning of the experiment the flexible object is grasped by the gripper and suspend slighlty above the horizontal surface so that the object is not touching the horizontal resting surface. There is always contact with the flexible object throughout the course of our experiment. The complete system flow in ROS environment and ROS interfaces is as shown in the following figure and tables

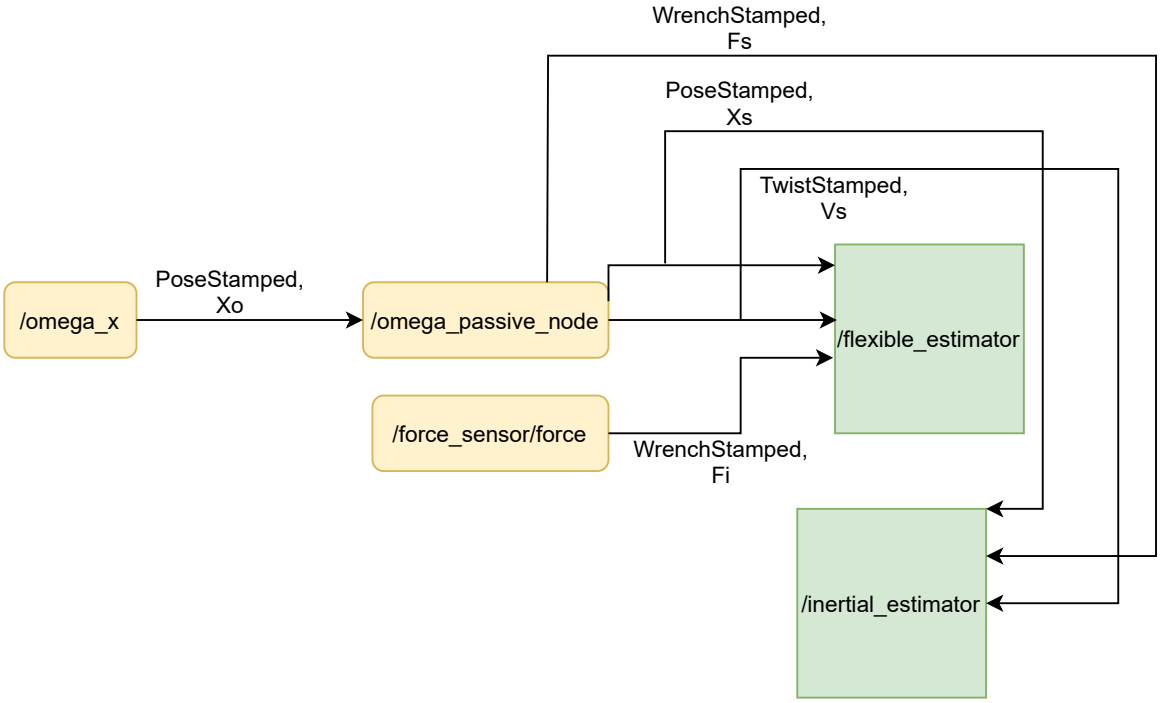


Figure D.3: ROS System Flow.

Node	/omega_x
Description	Omega State Publisher
Subscription	geometry_msgs/WrenchStamped
Publication	geometry_msgs/PoseStamped Xo

Node	/omega_passive_node
Description	Passive controller to ensure position tracking
Subscription	geometry_msgs/PoseStamped Xo
Publication	ibotics_msgs/EndEffectorState Es(Xs, Vs, Fs)

Node	/force_sensor_ros/Force
Description	Force torque sensor ROS interface
Publication	geometry_msgs/WrenchStamped Fi

Node	/flexible_estimator
Description	RLS estimator to estimate flexible dynamic parameters
Subscription	ibotics_msgs/EndEffectorState Es, geometry_msgs/WrenchStamped Fi
Publication	geometry_msgs/Vector3Stamped θ_f

Node	/inertial_estimator
Description	RLS estimator to estimate inertial dynamic parameters
Subscription	ibotics_msgs/EndEffectorState Es
Publication	geometry_msgs/Vector3Stamped θ_i

E Results and Discussions

This section dives deeper into the obtained results from the experiment. The estimation process begins when the flexible object is already grasped by the franka gripper and the object is suspended above the horizontal surface as shown in figure,

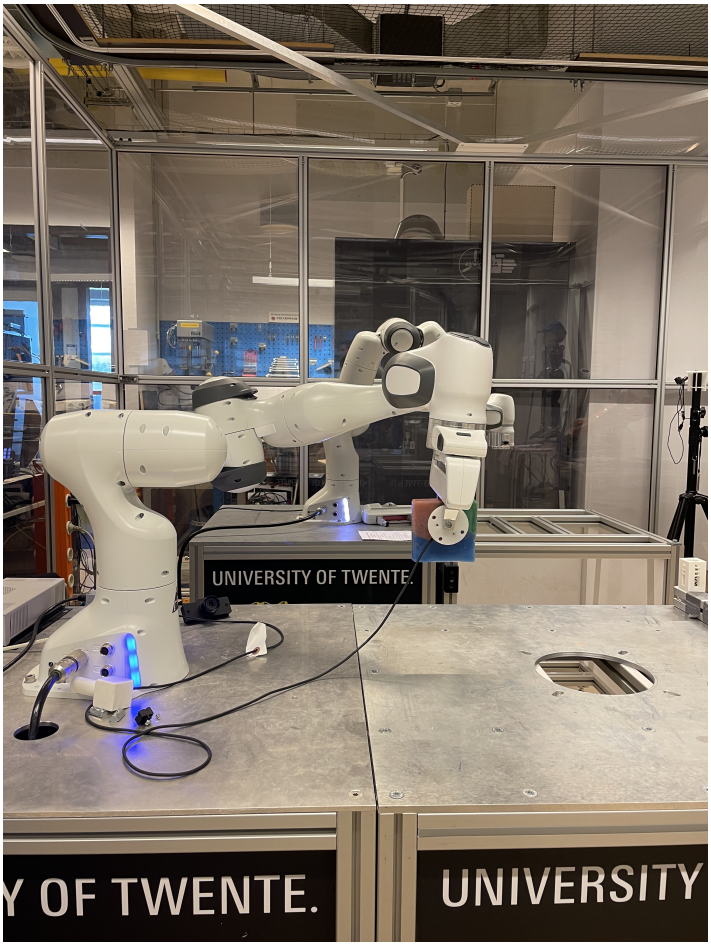


Figure E.1: Robot State before beginning the experiment

The goal of this experiment is to validate our hypothesis and to identify if any parasitic effects are associated with the inertial and flexible dynamics model. The experiment is carried out by moving the flexible object freely in space and the system is excited at different frequencies i.e the operator provides input positions by moving the handle of the haptic device at different speed.

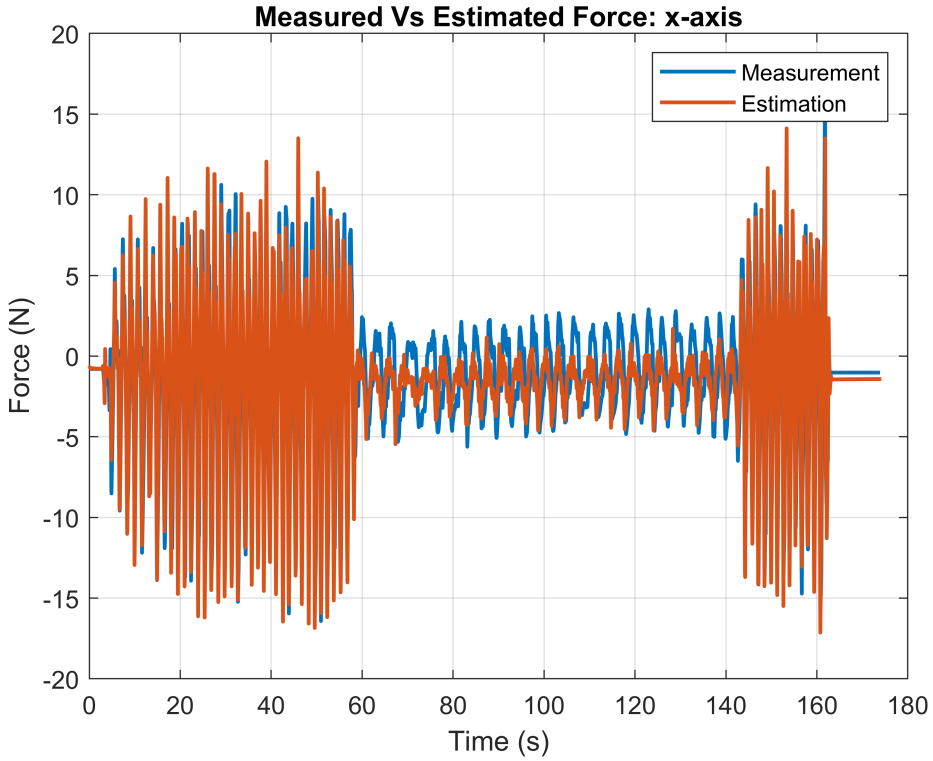


Figure E.2: Measured Vs Estimated Force along x-axis for inertial estimator

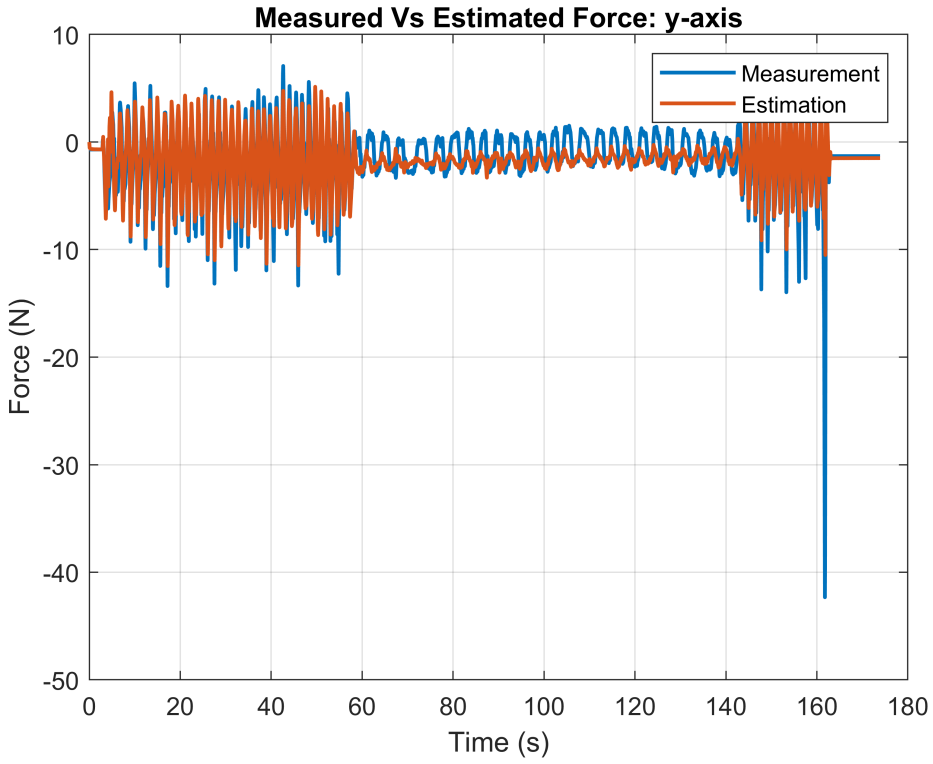


Figure E.3: Measured Vs Estimated Force along y-axis for inertial estimator

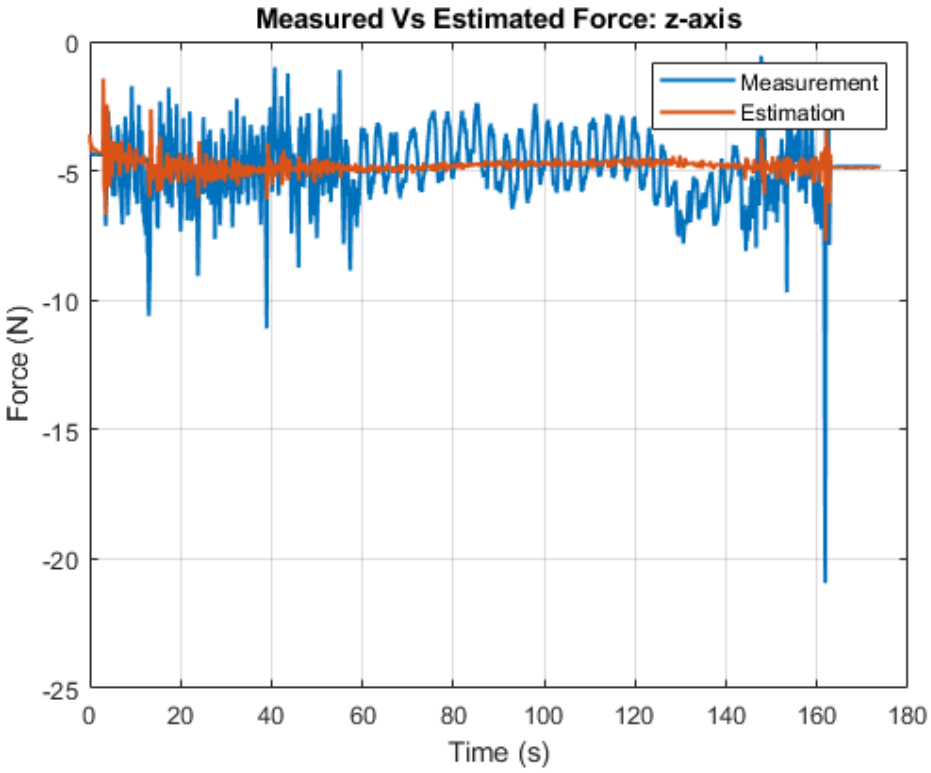


Figure E.4: Measured Vs Estimated Force along z-axis for inertial estimator

It can be seen from the plot for z-axis that the estimator doesn't converge to the measured values. This is because along z-axis the system is not excited enough making it difficult for the estimator to track the measurement. At higher frequencies the flexibility of the object does not introduces parasitic effect in our inertial model parameter estimation. Along y and x axis it can be seen that the estimates follow the measurements quiet well at high frequencies, however at low frequencies we see similar results as we saw for the z-axis, the estimator fails to track the measurement. At high frequencies the tracking is much better and the peaks of the estimated force match with the measured force. It takes nearly 10.56 sec for the estimator to converge, this could potentially be a problem in a real world application. The frequency spectrum of the tracking error along x-axis in log-log scale is as shown in the figure E.5. It can be seen that the error is nearly 0.3 N at approximately around 0.002 Hz, after which the estimation error reduces and stays below 0.1 N between 0.04Hz to 1.4 Hz, after which the error shows a very little change and is mostly noise. This matches the order of magnitude of the measured force at same frequencies as shown figure F.6,

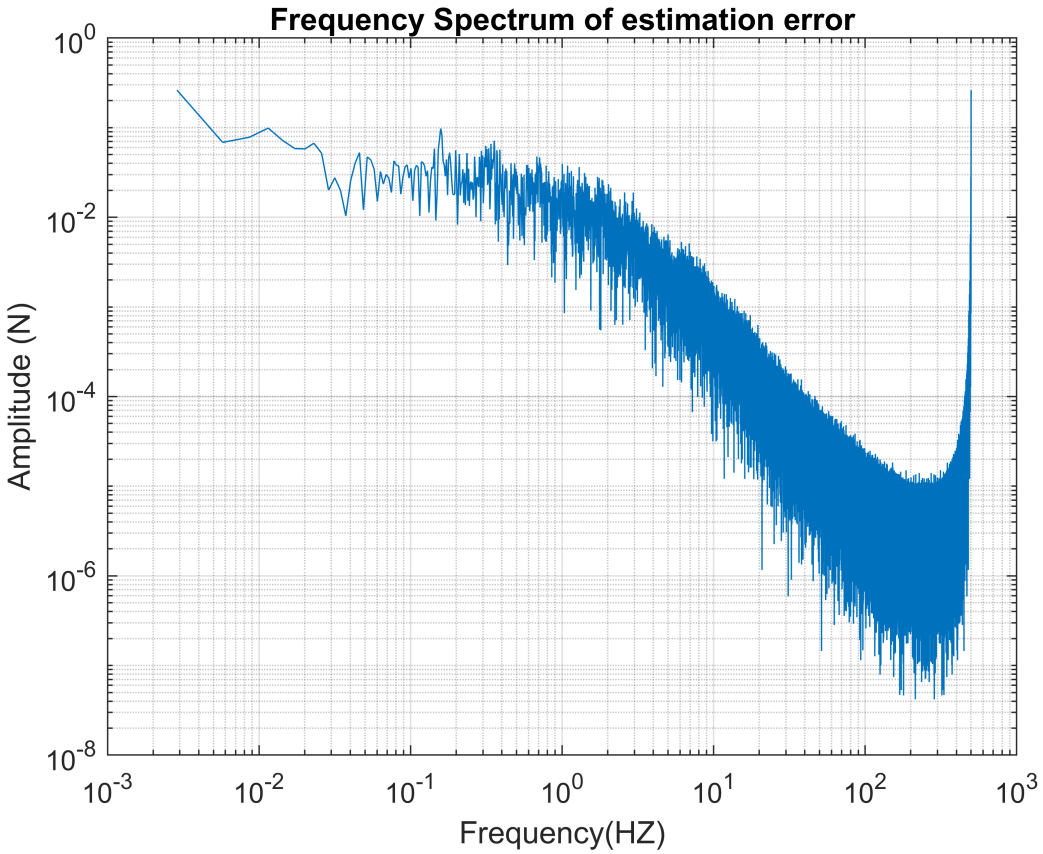


Figure E.5: Estimation Error Frequency spectrum for inertial estimator

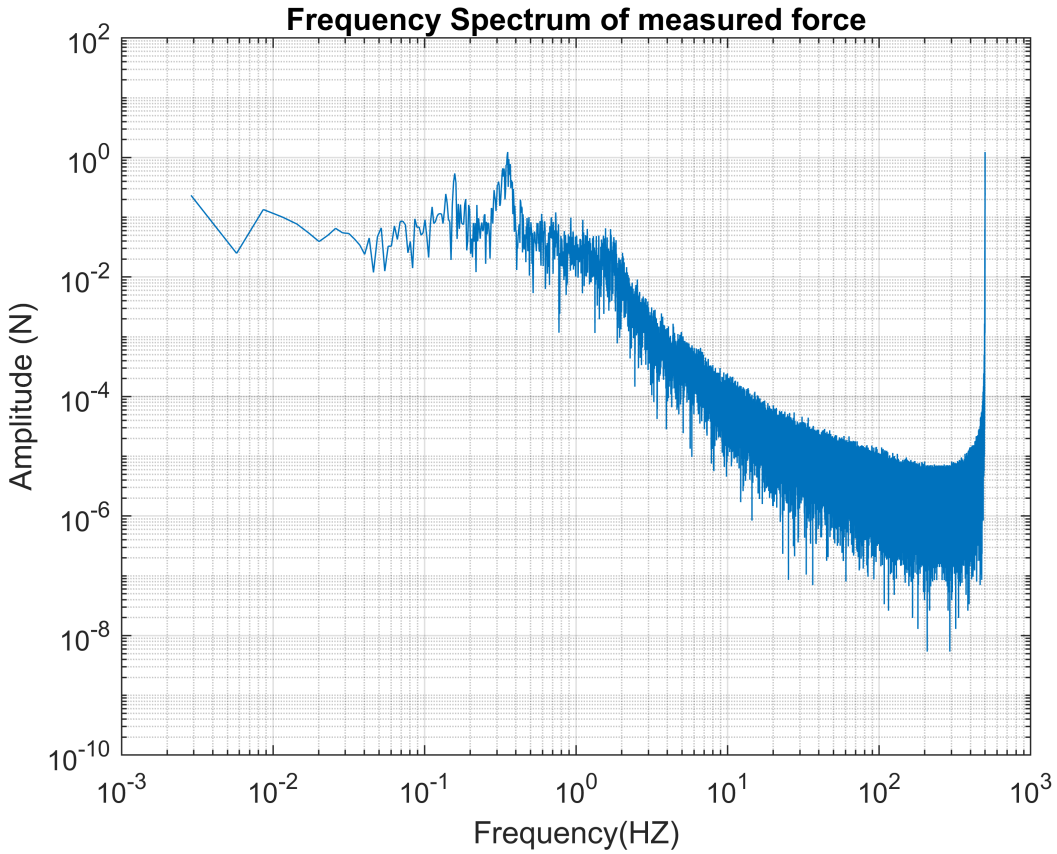


Figure E.6: Measured force Frequency spectrum for inertial estimator

The figures below show how the estimated parameter 'm' develops over time in seconds. It can be seen from figure E.7 that the estimator is converging and providing a stable estimate.

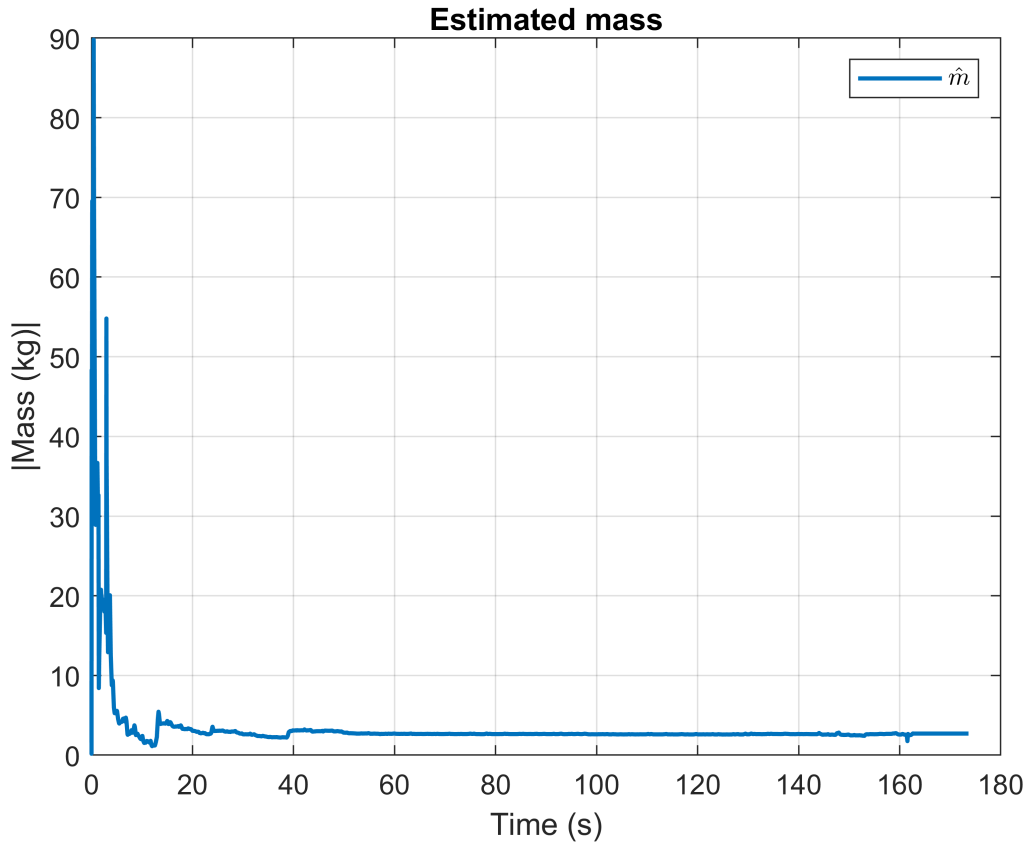
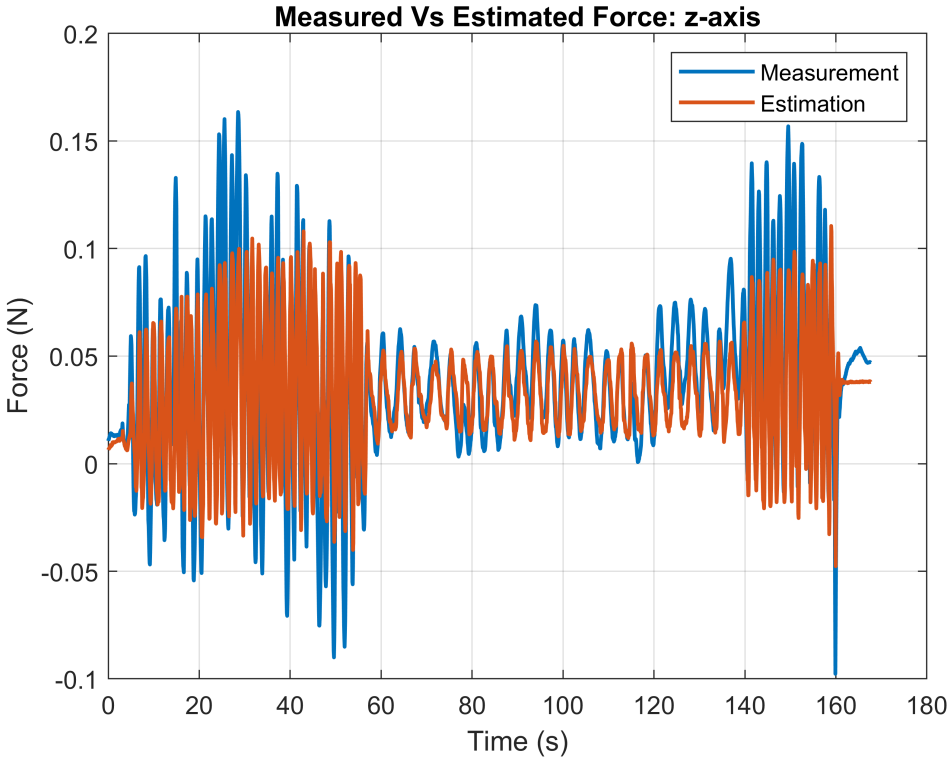


Figure E.7:Mass Estimation over time

The figure below shows the plot of estimated and measured force for the flexible model over time,



Figure

E.8: Measured Vs Estimated Force for flexible estimator

It can be seen that the tracking follows the measured force with noticeable inconsistencies, It also fails to match the magnitude of the measured force. Also the performance of the flexible model is poor than that of inertial model at high and low frequencies both. The frequency spectrum of the estimation error and measured force on a log-log scale is shown in figure F.8 and F.9. It can be seen that the error is roughly below 0.003 N between 1.6 Hz to 10.8 Hz, after which the estimation error shows barely minimum changes and is mostly noise. This matches the order of magnitude of the measured force at same frequencies.

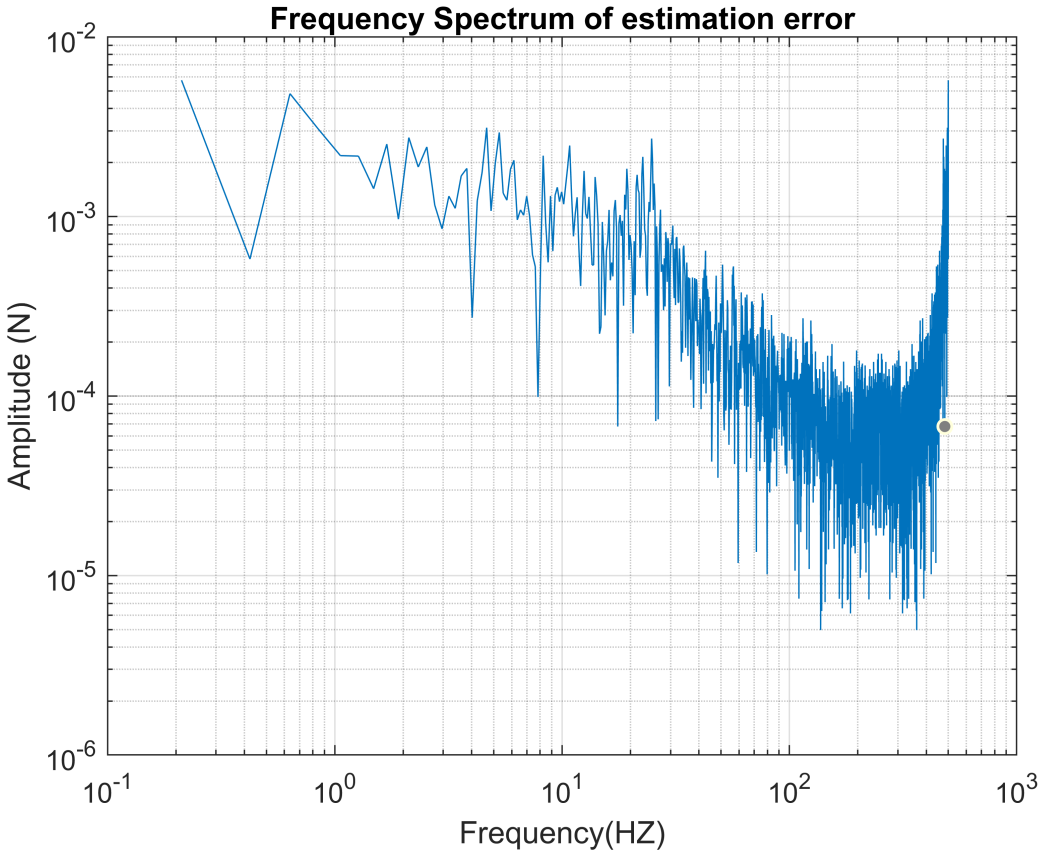


Figure E.9: Estimation Error Frequency spectrum for flexible estimator

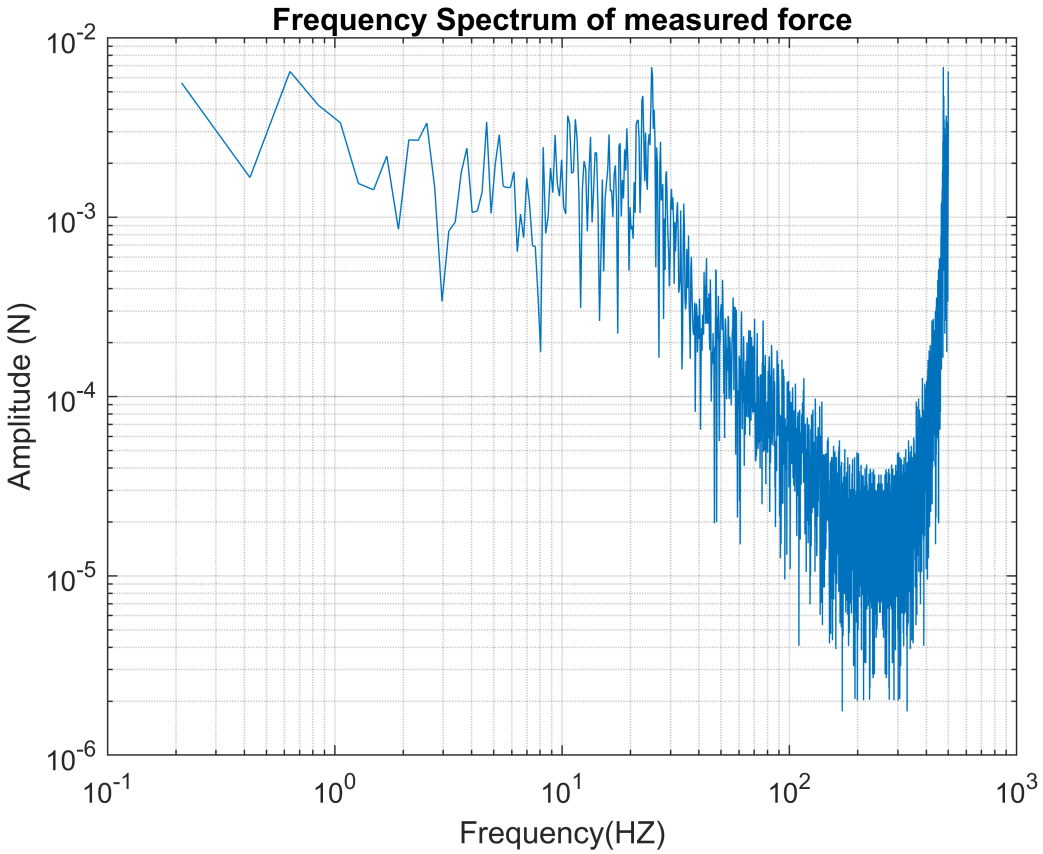


Figure E.10: Measured force Frequency spectrum for flexible estimator

The figures below show how the estimated parameters 'K', 'B' and 'n' develops over time in seconds. It can be seen from figures E.11,E.12 and E.13 that the estimator is not converging and providing a stable estimate.

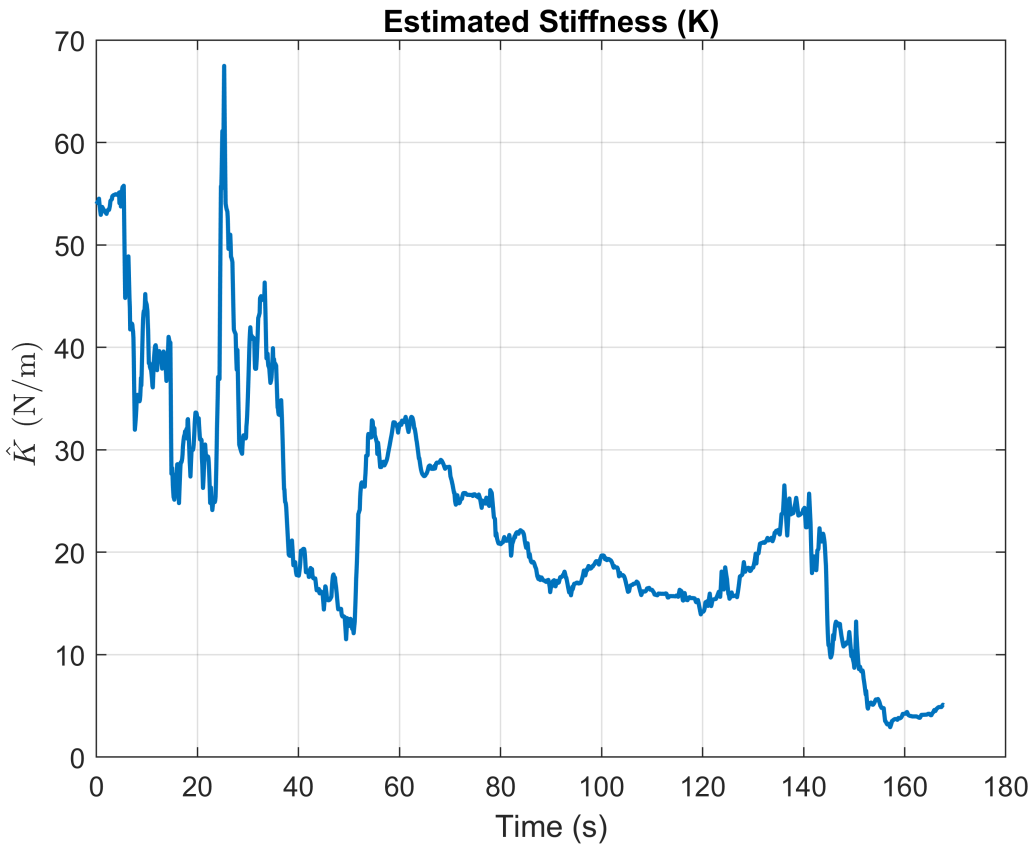


Figure E.11: K Estimation over time

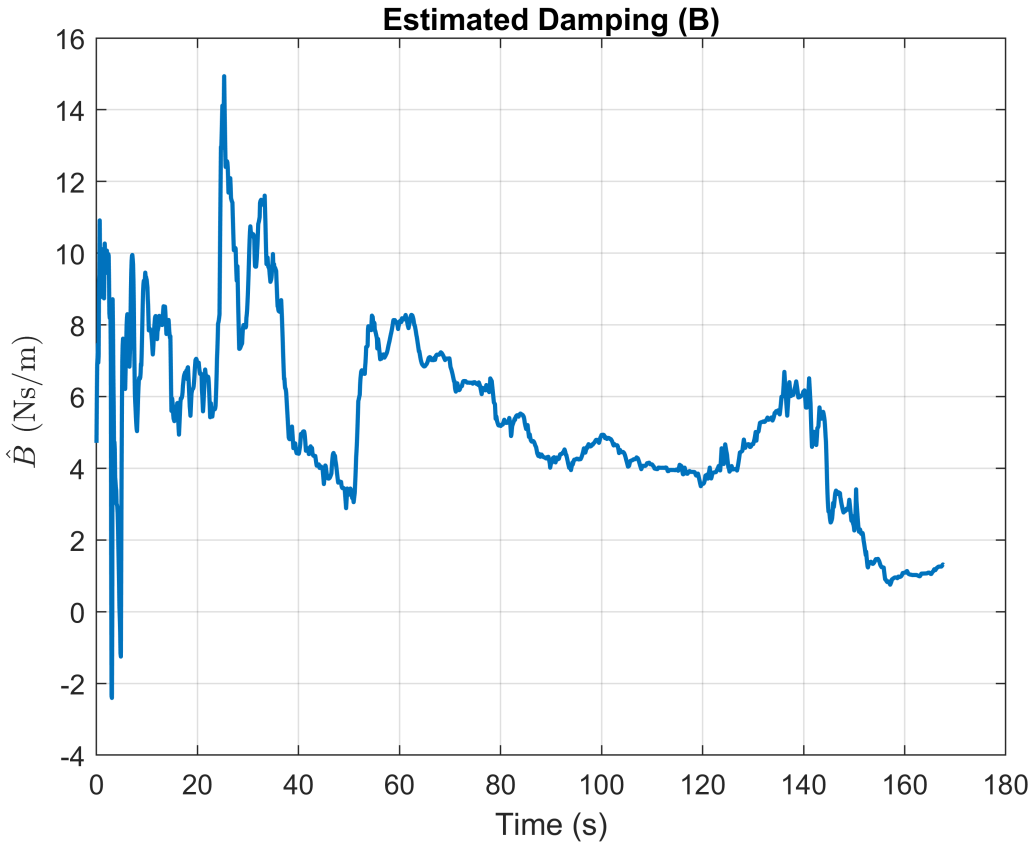


Figure E.12: B Estimation along over time

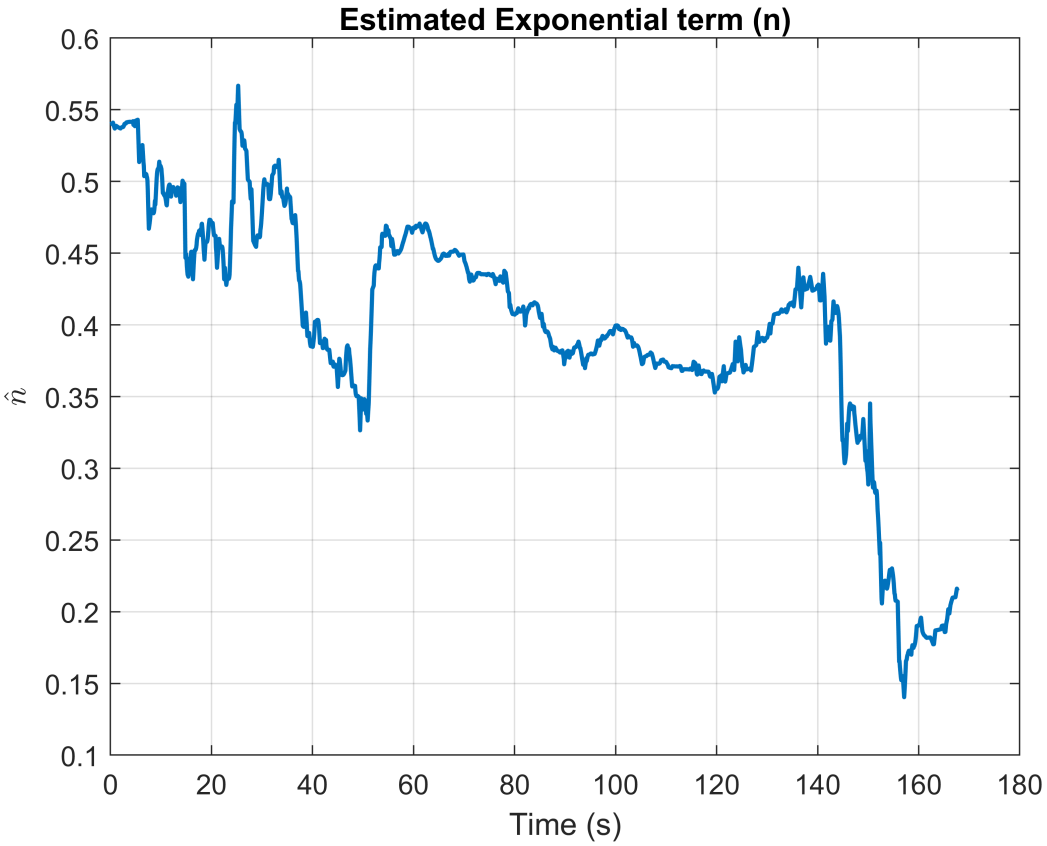


Figure E.13: n Estimation along over time

F Conclusion and future work

This section draws conclusions based on the experiment performed to give more information on our hypothesis. Based on our conclusions a brief overview is provided to improve this work in future.

F.1 Conclusions

To reiterate, our goal for this research work is to validate our hypothesis of breaking object dynamics to two sub-models and to approximate flexible and inertial dynamics. Also, to identify what parasitic effects affect the functioning of respective dynamic models.

The inertial estimator provide sufficiently good results at high frequencies. For flexible estimator the estimation is very poor throughout. For inertial estimator the estimation provides good results for frequencies between 0.04Hz to 1.4Hz. Above these frequency range the estimation error is of the order of the magnitude of the measured signal and can be considered as noise as the measured force is extremely small. Intuitively speaking, the objects mass and gravity which is absent in the flexible models equation of motion can cause parasitic effects. The estimators for the flexible model and the inertial model parameter estimation have been adequately realised, and our hypothesis holds true at high frequencies for inertial model.

F.1.1 Future Work

Our research work can act as a starting point for further improving the novel concept of breaking object dynamics into two sub-models. Since our method works well at high frequencies, research can be carried out to design a improved method which is reliable at low frequencies. Also, different on-linear models can be tested for estimating the flexible dynamics.

In this work major focus is given on the estimator design than force rendering, For a complete practical realisation a haptic device shloud be use in the form of a wearable glove which receives force feedback at two different locations, one at the wrist and other at the finger which would provide force feedback to the operator. Also, further research has to be carried out on manipulation of flexible objects in free space.

References

[1] X. Xu, B. Cizmeci, C. Schuwerk and E. Steinbach, "Model-Mediated Teleoperation: Toward Stable and Transparent Teleoperation Systems," in IEEE Access, vol. 4, pp. 425-449, 2016, doi: 10.1109/ACCESS.2016.2517926.

[2] G. Niemeyer and J.-J. E. Slotine, "Stable adaptive teleoperation," IEEE J. Ocean. Eng., vol. 16, no. 1, pp. 152 162, Jan. 1991.

[3] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," in IEEE Transactions on Automatic Control, vol. 34, no. 5, pp. 494-501, May 1989, doi: 10.1109/9.24201.

[4] D. Sun, F. Naghdy, and H. Du, 'Application of wave-variable control to bilateral teleoperation systems: A survey', Annual Reviews in Control, vol. 38, no. 1. Elsevier, pp. 12–31, 2014.

[5] Hannaford, B. and J. Ryu. "Time domain passivity control of haptic interfaces." Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164) 2 (2001): 1863-1869 vol.2.

[6] J.-H. Ryu, "Bilateral control with time domain passivity approach under time-varying communication delay," in Proc. 16th IEEE Int. Conf. Robot Human Interact. Commun., Jeju, Korea, Aug. 2007, pp. 986-991.

[7]J.-Q. Huang and F. L. Lewis, "Neural-network predictive control for nonlinear dynamic systems with time-delay," IEEE Trans. Neural Netwo.,vol. 14, no. 2, pp. 377-389, Feb. 2003.

[8] A. C. Smith and K. Hashtrudi-Zaad, "Adaptive teleoperation using neural network-based predictive control," in Proc. IEEE Int. Conf. Control Appl., Toronto, ON, USA, Aug. 2005, pp. 1269-1274.

[9] C. Tzafestas, S. Velanas, and G. Fakiridis, "Adaptive impedance control in haptic teleoperation to improve transparency under time-delay," in Proc. IEEE Int. Conf. Robot. Autom., Pasadena, CA, USA, May 2008,pp. 212-219.

[10] D. Verscheure, J. Swevers, H. Bruyninckx, and J. De Schutter, "Online identification of contact dynamics in the presence of geometricuncertainties," in Proc. IEEE Int. Conf. Robot. Autom., Pasadena, CA,USA, May 2008, pp. 851-856.

[11] X. Xu, J. Kammerl, R. Chaudhari, and E. Steinbach, "Hybrid signal based and geometry-based prediction for haptic data reduction," in Proc. IEEE Int. Workshop HAVE, Hebei, China, Oct. 2011, pp. 68-73.

[12] X. Xu, B. Cizmeci, A. Al-Nuaimi and E. Steinbach, "Point Cloud-Based Model-Mediated Teleoperation With Dynamic and Perception-Based Model Updating," in IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 11, pp. 2558-2569, Nov. 2014, doi: 10.1109/TIM.2014.2323139.

[13] S. Sirouspour and A. Shahdi, "Model Predictive Control for Transparent Teleopera-

tion Under Communication Time Delay," in IEEE Transactions on Robotics, vol. 22, no. 6, pp. 1131-1145, Dec. 2006, doi: 10.1109/TRO.2006.882939.

[14] Walt, Christophe van der, 2020, essay:85051, "Extending an Isotropic Virtual Environment Model with Geometrical Information in Model-Mediated Teleoperation"

[15] A. Haddadi and K. Hashtrudi-Zaad, "A New Method for Online Parameter Estimation of Hunt-Crossley Environment Dynamic Models," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 981-986, doi: 10.1109/IROS.2008.4650575.

[16] Force dimension - products - omega.7 - overview. https://www.forcedimension.com/products/omega-7/overview. Accessed: 2021-08-30.

[17] Franka Website. https://www.franka.de/. Accessed: 2021-08-30.

[18] Franka Control Interface (FCI) Documentation. https://frankaemika.github.io/docs/libfranka.html. Accessed: 2021-09-14.

[19] ROS Wiki. http://wiki.ros.org/. Accessed: 2021-08-21.

[20] Biagiotti L., Melchiorri C. (2007) Environment Estimation in Teleoperation Systems. In: Ferre M., Buss M., Aracil R., Melchiorri C., Balaguer C. (eds) Advances in Telerobotics. Springer Tracts in Advanced Robotics, vol 31. Springer, Berlin, Heidelberg.

[21] N. Diolaiti, C. Melchiorri and S. Stramigioli, "Contact impedance estimation for robotic systems," in IEEE Transactions on Robotics, vol. 21, no. 5, pp. 925-935, Oct. 2005, doi: 10.1109/TRO.2005.852261.

[22] A. Pappalardo, A. Albakri, C. Liu, L. Bascetta, E. De Momi, P. Poignet, Hunt–Crossley model based force control for minimally invasive robotic surgery, Biomedical Signal Processing and Control, Volume 29, 2016, Pages 31-43, ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2016.05.003. (https://www.sciencedirect.com/science/article/pii/S1746809416300453)