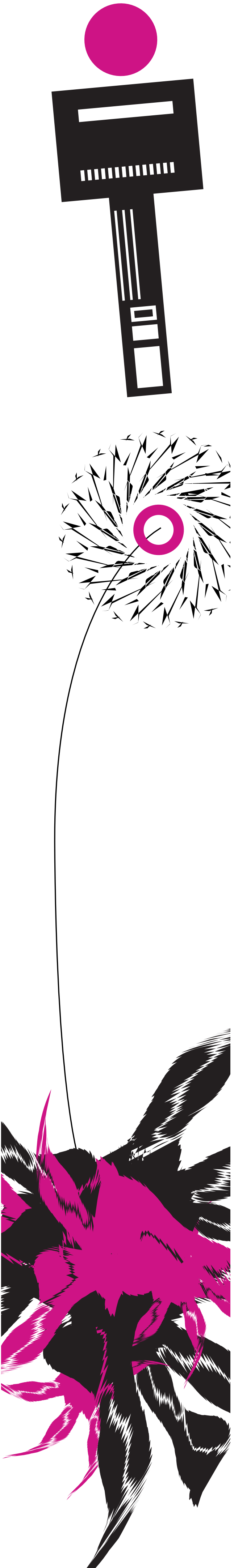


RAM. ROBOTICS AND MECHATRONICS

DESIGN OF A MMT SYSTEM FOR AN ENVIRONMENT WITH CHANGEABLE DYNAMICS

A. (Alejandro) Lopez Tellez

MSC ASSIGNMENT

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

January, 2023

003RaM2023
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

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

Abstract

Telemanipulation enables a user to operate in a remote environment by using a robotic system. However, challenges such as enabling a stable and transparent system under time delays have to be overcome to achieve that goal. Model-mediated Teleoperation (MMT) seeks to address this problem by avoiding traditional closed-loop control; instead, the MMT approach uses data from the teleoperated environment to construct and update a competent model of its environment. While the MMT literature provides solutions for an environment with a single type of dynamics, the dynamics in a teleoperation in complex environments are expected to switch depending on the type of interaction. This work explores how a MMT system can support an environment with changeable dynamics. In this study, two designs are created and experimentally tested to identify and switch model dynamics in an environment with changeable dynamics. To achieve that, an environment with changeable dynamics is modeled, where the type of information available on the environment is the main source to inform model switching for a MMT system. The effectiveness of the two designs, along with the modelling accuracy of the proposed environment, are assessed, showing the conclusion and future lines of research for the proposed MMT design.

Contents

1 Introduction **1**

1.1 State the problem 2

1.2 Research question 3

2 Related Work **4**

2.1 Modelling of the Environment 4

2.2 Model Switching Design 5

3 Model Switching Design **7**

3.1 Model switching design 9

4 Environment Modelling Design **12**

4.1 Introduction 12

4.2 General Assumptions 13

4.3 Planar Friction Submodel (3DOF) 14

4.4 Inertial Submodel (6DOF) 15

4.5 Cavity Friction Submodel (1DOF) 16

4.6 Submodels transitions 18

5 System Design **22**

5.1 Introduction 22

5.2 Reactive MMT System 22

5.3 Proactive System 26

6 Experiment Design **35**

6.1 Experimental Setup 35

6.2 Remote Environment Reconstruction 36

6.3 Experimental Protocol 36

6.4 Experiment Evaluations Tools 37

7 Experiments Results and Discussion **40**

7.1 Introduction 40

7.2 Model switching detection 40

7.3 Model switching time response C_t 41

7.4 Modelling accuracy NMRSE 43

8 Conclusion and Future Work **48**

8.1 Conclusion 48

8.2 Future Work 49

A Data Acquisition **50**

A.1 Acceleration estimation 50

A.2 Analysis of the initial remote forces 50

B RLS Estimation: Mathematical Derivation **52**

B.1 Inertial submodel estimation 52

B.2 Planar friction submodel estimation 52

B.3 Cavity friction submodel estimation 53

Bibliography **54**

1 Introduction

Telemanipulation enables a user to operate in a remote environment by using a robotic system. The teleoperated system projects the operator's actions in the remote environment, allowing the application of human dexterous skill in hostile or unreachable environments to the human user. An ideal telemanipulation system could produce a complete immersion in the remote environment for the users. The user could replicate their movements on the remote side and be able to perceive the remote environment as they were present. The main challenges for Teleoperation are the time delays between the master and slave communication systems which motivates the creation of control strategy that can accomplish the functionality described above. These goals are grouped in two terms, stability and transparency [5] that shapes the different proposals for Teleoperation. System's Transparency is described as an exact match between the position and force signals of the master and slave device [23], from the perspective of [26] transparency is the match between the impedance of the environment and the impedance that the teleoperation user perceives. Stability relates to the stability of system problem in control theory, to produce a bounded output for a given bounded input.

Both concepts are highly affected by the communication time delays. To overcome time delays some of the techniques used in teleoperation tend to improve one term to the cost of the other, e.g by increasing the impedance of the system to avoid out-bounded responses, stability is improved to the cost of decreasing the transparency of the system [7]. In other words, teleoperation systems prioritize one criteria of the system at the expense of the other.

To tackle both stability and transparency in the presence of communication delays, there are multiples approaches, passivity-based control schemes, such as the wave-variable transformation [10], time domain passivity control (TDPC) [6], however, for these cases the system remains too conservative, ensuring stability over transparency, therefore reducing its performance. Other methods are used in combination with new technologies to produce enhanced performance, these methods fall under the umbrella of Model-mediated teleoperation [8], such as the case adaptive impedance control [16].

In contrast with the other methods Model-Mediated Teleoperation (MMT) handles teleoperation by avoiding the traditional closed loop control.The MMT approach uses sensor data from the remote environment to construct and update a competent model of its environment. The user interacts locally with a virtual environment constructed from the extracted model of the remote environment.

A MMT system decouples the teleoperation into a local system, where the user interacts with a modelled environment, and a remote system, where the slave device replicates the users actions in the real environment after they interacted with the virtual environment in the local system. The application of a local model is the main advantage of MMT as the user interaction is not disturbed directly by the communication time delays which is the greatest problem for any teleoperation method [7]. Especially in the case of space application where the time delays are extremely large [27].

However, time delays remains an essential challenge for updating and extracting an environmental model. In the initial state of system, the local environment follows a model where its parameters are unknown. The remote side provides this information by the use of an estimator system whose data will converge after a time, stabilizing the local model.If the remote environment changes, the local model needs to be updated. If no update is given, then the local model is imprecise and the movements of the operator in the remote environment wont follow the expecting outcome and the computed forces from the estimation are incorrect.

The problem of stability and transparency in an MMT system relies in the accuracy of the environment‘s model and the capabilities of the estimator to keep pace of the changes in the environment. Then, if the local model is precise and stable the transparency of the system is ideal.

1.1 State the problem

In MMT, the main challenge is modelling the environment. A perfect modelling of the remote environment results in a stable teleoperation even under arbitrary communication delays [20]. To achieve it the system needs a competent model and a good estimation of the parameters that determine the physical behaviour of the model. The use of parameter estimation methods implies that during a period of time, the local and remote environment do not match. Based on this lack of synchronization between the local and remote enviroment [23], two states are defined in an MMT system: steady and transition states. In the transition state, the remote environment and the local virtual model do not match. During this state, the operator won't be able to operate in a precise manner in the remote environment. An MMT system should minimize the time of the transition state by minimizing the time required for parameter estimation. Opposite to the transition state, the steady state represent the case when local and environment models match and the local model has a good estimation of its parameters.

If we interact with a complex environment, we cannot expect that its dynamics remains the same in all type of interactions. For example, in a manipulation activity we can always expect contact with different elements of the environment. For every interaction, humans perceive different dynamics (friction, inertia, etc). This type of environment, where its dynamics switch, will be denominated an environment with changeable dynamics. The central goal of this study is to design an MMT system that supports changeable dynamics environment.

In a MMT system, figure 1.1, the user interacts with the local model that renders forces computed from the dynamic equation of the model. In the case of a changeable dynamic environment, we can group those multiple dynamics into a single model with a switching behaviour. If the model of the environment is the entire representation of the environment, a sub-model represents one type of those dynamics present in the environment, defined by a single differential equation (DE). As a consequence each sub-model included in the system needs an estimator.

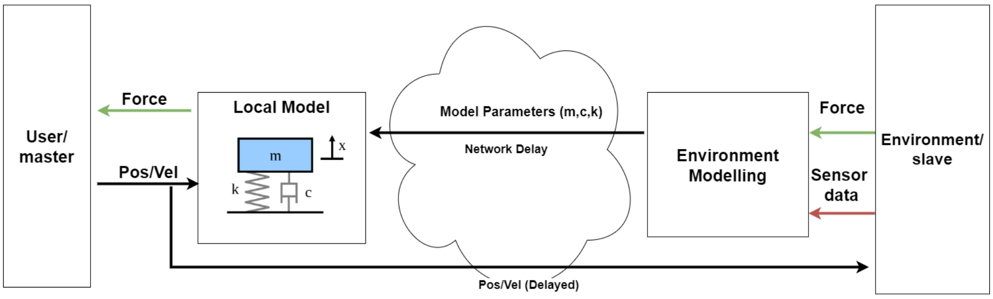


Figure 1.1: Structure of a MMT system

With multiple sub-models, the Local model can reconstruct different dynamics for the user. The decision of what sub-model is displayed requires an additional process in a MMT system called model switching.

1.1.1 Model Switching

Model switching is deciding what theoretical model can represent the environment dynamics between multiple possibilities. In previous MMT studies the type of dynamics to model and estimate is selected by the researcher. In the case of an environment with multiple dynamics this decision is not straightaway and it becomes a function of an MMT system to discern what are the appropriate type of dynamics, sub-model. The system needs information from the environment to support this decision before it can be made. The MMT environment applies multiple sources of information to model the environment; however, the topic of what type of information is valuable for model switching does not have a straightforward solution.

1.2 Research question

The design of model switching for a MMT system in an environment with changeable dynamics is the main focus of this research. Its main intention is to explore the application of MMT in an environment with multiple dynamics. To achieve that, the system needs to select a sub-model that matches the current dynamics of the remote system. This functionality is called model switching an its design depends on what type of information is used to make the selection. The research questions considered here can be formulated as follows:

”How can a MMT system be designed for an environment with changeable dynamics?”

To answer the research question in depth one sub-level question need to be addressed, which is as follows:

- What type of information available in the environment is effective to inform model switching?

2 Related Work

In this section, we described the related literature used in designing an MMT system for an environment with changeable dynamics. This chapter explores first how to model an environment with changeable dynamics and secondly a methodology to design model switching in this type of environment based on the information available in the environment.

2.1 Modelling of the Environment

Modeling the environment is the first step in developing the design of a MMT system for an environment with changeable dynamics.This section provides an overview of the models that may be used to support the suggested environment and discusses how those models are estimated.

Previous studies in MMT literature do not offer a clear example of an environment with multiple dynamics. Xu et. al. [23] provides an extensive illustration of the physical models applied in MMT literature until its publication in 2011. Most of the described approaches model the dynamics that occurs when the teleoperation contacts an element of the environment named contact models or so-called rigid models dynamics. Passenberg, C [12] splits MMT models into movable and static object models. Movable object models present an inertial component in their dynamics, while static object models are dominated by damping and stiffness parameters. The distinction between mass and stiffness dominated models hints what type of information of the environment will be present during the activation of those dynamics. In the case of mass dominated it expect to measure relevant accelerations and velocities in the remote environment, while in the static object these kinematics will be minimal. This provide a notion of what type of information will be useful if the selected models for the environment follow this classification.

Another categorization that combines the notion of movable and static object are friction model which maintained a rigid contact and move, resulting in modelling a friction component. For modelling the friction component of the environment, Olsson et. al.[11] provides a general overview of friction models. In the MMT literature Verscheure et. al.[17] demonstrates the estimation of a friction model which combines an static object models, stiffness dominated models, and a friction component that ignores the static friction and only estimates the dynamic friction, represented as the Coulomb friction. Xu et. al.[22] utilises a friction model where the static friction coefficient is assumed to be the same as the dynamic static coefficient. Xu et. al.[21] incorporates the concept of a movable object and combines the inertial component with a dynamic friction model, it also provides the first model for a rotational friction in a 2D plane. Ni et. al [9] integrates the Karnopp friction model in an MMT system where all the previous parameters are estimated: static friction, dynamic friction, mass, damping and stiffness of the environment. Overall the results in Ni et. al [9] shows exceptional results to estimate the dynamics of an object sliding in a 2D plane, however, the trade-off is a significant increment in the complexity of the estimation system, including that one the estimations steps needs a manual evaluation.

For the estimation methods in MMT Xu et. al.[23] provides a complete description of the estimation methods used in each MMT model, where Recursive Least Squares filter (RLS) is the most used. Passenberg, C.[12] provides an evaluation of which estimation method performs better given the nature of the model, static or movable object. Her results show the RLS estimator as the most efficient and, in particular the Recursive Least Squares filter with self-perturbation (RLS-SP) for both types of object. For both cases a higher forgetting factor (λ) improve significantly disturbance and noise rejection to the cost of lower tracking performance. Wang et. al [18] presents the Recursive Least Squares with Adaptive Forgetting Factor (RLS-A) that improves the tracking performance of the system by varying the forgetting factor parameter, however Passenberg, C. [12] evaluation of RLS-A results in the lowest performance compared with the other RLS estimation processes.

The literature review on modelling the environment does not provide a direct answer to model an environment with changeable dynamics. The reviewed models outlines the possible physical parameters required to describe the environment, but they don't provide a model on how different dynamics may be switched between them. The classification of movable objects in [12] and the different friction models explain that the dynamics of manipulating an object shift when it comes into touch with other objects in the environment, while maintaining certain features such as the object's mass. As a result, it suggests that a manipulation task with multiple interaction could be a possible scenario for an environment with multiple dynamics and that all of the various dynamics contained in the manipulation task should include a mass parameter. According to the analysis of estimation techniques, the RLS algorithm is the best tool due to its effectiveness and widespread use. In the event that the system requires improved tracking or in the present of noisy data, the RLS algorithm offers a number of alternatives.

2.2 Model Switching Design

For the design of a model switching functionality Xu et. al. [23] does not describe model switching strategies but it provides a description of what type of information is used in MMT systems, and therefore what type of information can be used to inform model switching. The most common type of information are forces, the motion of the slave device(position, velocity and acceleration), visual data in form of images or pointcloud data and processed information. Processed information is a byproduct of the previously mentioned information type: forces, motions of the slave and visual data. Processed information includes the model parameters obtained from the estimation process and as a special case in Smith et. al. [14] it used weights coming from a Neural net (NN) used to model the dynamics of environment using a non-parametric modelling. The weights are obtained by using the forces of the environment and kinematics of the slave device. The quality of these weight, as an important feature, is dependent on the available knowledge of the environment.According to this interpretation, MMT systems use information from the environment in two ways: directly extracted from the environment or after it has been processed in an initial process.

The MMT literature provides some methods for discerning the dynamics of the environmental. Ni et. al. [9] provide a friction model that is divides in three motion stages.These phases are the sliding phase, the critical phase, and the static phase. Understanding how forces affect how objects move is the base for reconstructing such dynamic model in the virtual environment. The first and only direct method to distinguish between two dynamics models: Hunt-Crossley(HC) model and the Kelvin-Voigt(KV) model was put forth by Achhammer et al. in [1]. The proposed model switching strategy in Achhammer et al. in [1] is determined by the estimation value of a parameter shared by both models. The system model switches to the appropriate model by including the following features in the system:

- A contact detector, to inform the system the presence of two possible dynamics.
- An estimator for each model, where both estimator runs in parallel
- A defined value of the shared parameter that it uses as threshold to model switch between the two possible models.

If the estimated stiffness is higher than the threshold, suggesting a stiff object, the system displays KV model, otherwise the HC model is selected. The understanding of Achhammer et al. in [1] points in the direction that model parameters can be utilised as a source of information to inform model switching, but it does not offer a conclusive solution because it only takes into account which model is more accurate in one type of interaction. For the geometrical detection in the virtual environment, Xu et. al. [25] use the position in the virtual environment to change the type of dynamics that are used in the rendered forces. In the remote environment, the contact forces are compared with a threshold force to initiate the parameter estimation process. In this example, the forces are utilised as a threshold to activate or deactivate the estimating process, which may imply that a model switching process could benefit from using a similar technique.

This chapter's conclusion is that there is no established method for designing a MMT system for an environment with changeable dynamics.

3 Model Switching Design

This chapter introduces an analysis of the available type of information in the environment, addressed in in the section 2.2, that can be applicable to inform a model switching functionality. The research led to the development of two designs for modelling switching. Given the nature of the information, each model switching approach necessitates an adjustment in the MMT architecture, resulting in a unique system design. At the end of this chapter, the requirements for each system design are suggested. In MMT systems the most common type of information present in the environment are the forces of the environment, the position of the manipulation, visual data from the environment and the results of the estimation process.

Forces
In order to estimate the model parameters, a MMT system's remote forces are traditionally employed. The remote forces can also be utilised as a contact detector to start the estimate process [20]. Yet, the application of remote forces can go further. In this scenario, the remote forces are helpful because they enable the system to distinguish when the teleoperation is in contact with an object or in free space. Additionally analysing how forces varies during time can reveal a transition in the environment's dynamics. For instance, given that there is a force sensor located in the frame of the teleoperated object, the gravitational forces cease to be observed if the teleoperation comes into contact with an object that holds its weight.

In this regard, the main benefit of employing forces is that they have a direct relationship with the dynamics of environment. Integrating an analysis of the forces variation with the knowledge of the possible physical models that outputs those forces profiles it will be possible to discern dynamics of the environment and inform a model switch strategy. The main disadvantage of the application of remote forces is that it will likely slow down the process when there are time delays if it is compared with a model switching functionality located and that applies information from the local side of the teleoperation.

Position of the manipulation and visual data
Using position of the manipulation and visual data can work as combination to reconstruct the remote environment as it being done in previous MMT literature [19][16][24][25]. These studies used this information to create a virtual environment, that is the reconstruction of the remote environment to improve the modelling accuracy. In a virtual environment the user can interact with objects of environment before it occurs in the remote environment.If the reconstruction is accurate the virtual position, that is the master device position in the virtual environment, matches the position of the teleoperation, the slave device position.

Instead of limiting the use of virtual environment for modelling it can be used for model switching. For that purpose, the objects in the remote environment are analysed to find its features such as its type of material, shape and dimensions. These features are used to evaluate what type of dynamics the teleoperation will sense when in contact with those objects. Under this premise a type of dynamics,therefore sub-model, is assigned to an object located in a physical space in the remote environment. By matching a type of dynamics with a physical position of the environment, the virtual environment is used to select what type of dynamics the teleoperation will sense. Then when the user navigates in the virtual environment and enter in contact with an object the system can switch to the type of dynamics that were assigned to that object.

As an example, see figure 3.1 that represents the reconstruction of a remote environment with two elements and so two type of dynamics. When the teloperation moves to position A the type of dynamics will be changed to the assigned type of dynamics. From A to B position will also changes the sensed forces from A to B dynamic type, creating a relationship between the multiple dynamics present in the environment.

The main advantage of this type of information is the virtualization of model switching functionality and therefore avoiding times delays. Its main disadvantages are the requirement of a more complex system that needs an accurate reconstruction of the environment, the ability to couple the motion and dynamics of the environment and that it needs a way to verify that those dynamics and reconstructed environment match the remote environment.

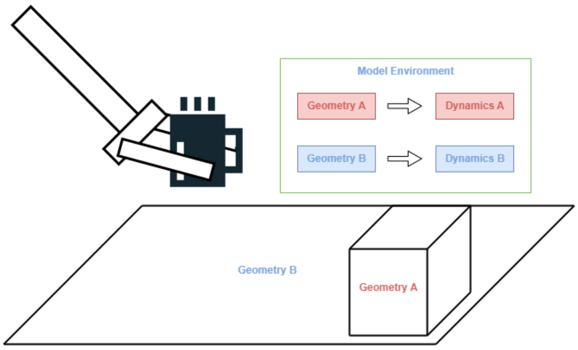


Figure 3.1: The reconstructed environment seen as a set of geometries, the dynamics that the user feels in contact with them are matched to their geometrical properties

Estimation error and model parameters

Another type of information relevant to identify a sub-model comes from the output of the parameter estimators. The parameter estimator applies the environmental information to evaluate if its model fits the environment. The parameter estimator provides an estimator error, that is the difference between the forces of the environment and estimated forces computed from the environmental information and the selected dynamic model whose parameters are being calculated.

The estimation error is a direct measurement of how the selected dynamic model represents the remote dynamics of the environment. If the estimation error is minimal it means that the estimated model fits the dynamics of the environment, therefore the system will switch to the type of dynamics that the estimator is computing. In an environment with multiple dynamics there will be a parameter estimator for each type of dynamics, an the value of estimator error of each estimator will be used to inform the model switching functionality. The main advantage of this approach is its simplicity as it does not need additional modules and that can be expanded to as many dynamics as estimator are included in the system. The main disadvantage is that to inform model switching the system needs to wait for the parameter estimator located in the remote side of the teleoperation,therefore suffering from time delays.

The second output from a parameter estimator is the estimated parameters, that can be also used to discern between dynamics models as it shown in [1]. The application of model parameters to inform model switching only works if all possible dynamics shares a common parameter and if the value of that parameter is different in each dynamics. For instance, its application would be helpful to identify a friction model between materials with low friction, such as ice, and high friction coefficient materials, such as rubber. The high level of specialisation in its application is the key benefit and downside of the model parameters in this scenario. In this case, this methods suffers from time-delays, its provides a simple structure but it cannot be used in all type of scenarios given its specific application.

The four types of information described are grouped in table 2.2, where their advantages and disadvantages are categorised into performance under time delays, the complexity of the system and its application to multiple type of environments. From the table none of the four variables has a clear advantage over the other. The estimation error provides a simple and general approach to the cost that it cannot prevent time delays to influence its performance.

On the other hand, position and visual data approach avoids time delays effect to the cost of a more complex system that needs the creation of an accurate reconstruction of the remote environment, that matches the remote environment. Based on this analysis we proposed two strategies, a model switching based on the estimation error and a model switching based on the combination of position, forces and visual data to overcome the flaws that both type of information have individually.

	Model switching features		
	Time delays independence	Complexity of modifications	General application
Forces	-	-+	+
Position and visual data	+	-	+
Estimation Error	-	+	+
Model parameters	-	+	-

Table 3.1: Summary of the information analysis to inform model switching in a MMT system

3.1 Model switching design

Two set of information are selected for model switching design: the information directly available in the environment (forces,visual data,position) and the estimation error output from the estimation process.

3.1.1 Identification of sub-models using the estimation error

Compared with the MMT structure of figure 1.1, the implementation of this type of model switching design does not require significant modifications. This design uses the output of the parameter estimators, therefore the estimation process of the MMT system and the model switching functionality work simultaneously. To include this design in an environment with multiple dynamics, the MMT system increases the number of estimators it uses, adding one for each additional type of dynamics, sub-model, that might be present in the environment.

The main idea is that all sub-model estimator runs in parallel and select the sub-model that fits better with the current dynamics of the remote environment. This functionality follows the same procedure that a MMT system does to update its parameters by minimizing the estimation error $e_k = y_k - \hat{y}_k$, where y_k are the remote forces at step k and $\hat{y}_k$ the estimated forces in the estimation process. The submodel results that minimize the most the error is selected and rendered to the user. The additional function to include in a MMT system is a module that gathers all errors for the incoming sub-model estimators. This function will decide what submodel is selected based on the estimation error and send the submodel and its model parameters to the local model to be rendered to the user. In case that another submodel becomes more fitting for the current dynamics this module will model switch to the appropriate submodel and its model parameters.

To manage the proposed model switching, an MMT system needs to adjust its structure as shown in figure 3.2. To provide a detection system based on this information the MMT system needs to include the following modules:

- Estimators for each submodel.
- A local renderer that can render multiple dynamics.
- A model switch module that assesses the fittest dynamics to be delivered to the local renderer based on the estimation error of each submodel estimation process.

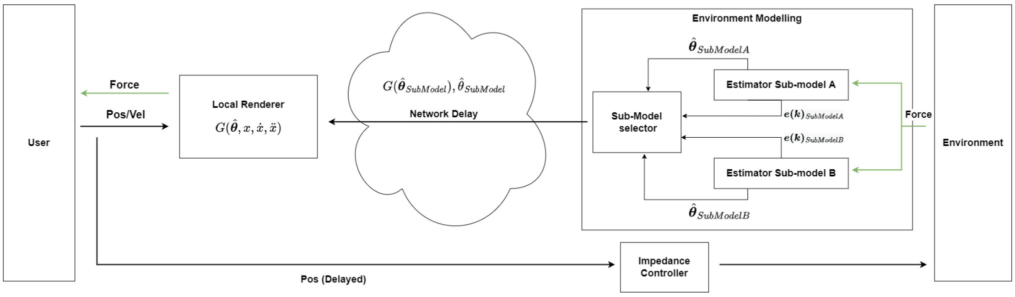

User
Force
Pos/Vel
Local Renderer
$G(\hat{\boldsymbol{\theta}}, x, \dot{x}, \ddot{x})$
$G(\hat{\boldsymbol{\theta}}_{SubModel}), \hat{\theta}_{SubModel}$
Network Delay
Environment Modelling
$\hat{\boldsymbol{\theta}}_{SubModelA}$
Estimator Sub-model A
Sub-Model selector
$\boldsymbol{e(k)}_{SubModelA}$
$\boldsymbol{e(k)}_{SubModelB}$
Estimator Sub-model B
$\hat{\boldsymbol{\theta}}_{SubModelB}$
Force
Environment
Impedance Controller
Pos (Delayed)

Figure 3.2: MMT system for an environment with two type of dynamics, sub-models, A and B, the Sub-model selector choose the sub-model that provides less estimation error, the chosen sub-model and parameters are sent to the local model to render forces to the user.

3.1.2 Identification of sub-models using direct information from the environment

In the application of direct information from the environment, virtual position, forces and visual data are used to reconstructed the environment and to select the appropriate sub-model. Using the reconstructed environment, the system matches each sub-model of the environment with a physical position of the object that provides those dynamics. Then, the detection of sub-models consist in analysing if the teleoperation is in contact with the geometry of the object that provides that type of dynamics. To achieve that, the system observe the position in the virtual environment, named virtual position, and the reaction to the contact with an object, which is the observed kinematics constraints. The kinematics constraints are detected by observing a constraint in the movement, checking the variation of the velocity, and the reaction force produces from the contact. As an example, the teleoperation activity interacts with a rigid table, the system will detect first that the position of the teleoperation is in the physical space of the table and, secondly a kinematic constraint is observed in the normal direction of the table, as the object cannot go through it.

In this strategy before the system starts to detect dynamics and to model switch, it needs to reconstruct the virtual environment and to create a database of the information extracted from the environment. This database contains the possible sub-models of the environment, its location in the environment and the estimated parameter for each sub-model. All this information, combined with the virtual position and motion constraints it computed in a model switch module to switch to the appropriate sub-model that matches the dynamics of the environment.

As the system selects the sub-model based on the position and detections of motion constraints, the estimation process can begin after the system switches to the appropriate sub-model. In this context the estimation process is controlled by the switching model process as it select the sub-model. The system model switches to the appropriate sub-model, that decision is sent to selected sub-model estimator to begin the estimation process. This hierarchy can be used to add an initial estimation to the parameters in the estimation process. This initial estimation can be provided by previous estimations of the other sub-models that shares common parameters, which theoretically, it will minimize the time the estimator needs to provide accurate results.

To manage the proposed model switching, an MMT system needs to adjust its structure as shown in figure 3.3. To provide a detection system based on this information the MMT system needs to include the following modules:

- Estimators for each submodel.
- A virtual environment of the reconstructed remote environment that provides the virtual position and kinematics constraints.

- A local renderer that can render multiple dynamics.
- A database to gather all known information of the environment.
- A model switch module that incorporates the described strategy and select the submodel, that it send to the local model and to be estimated in the remote environment.

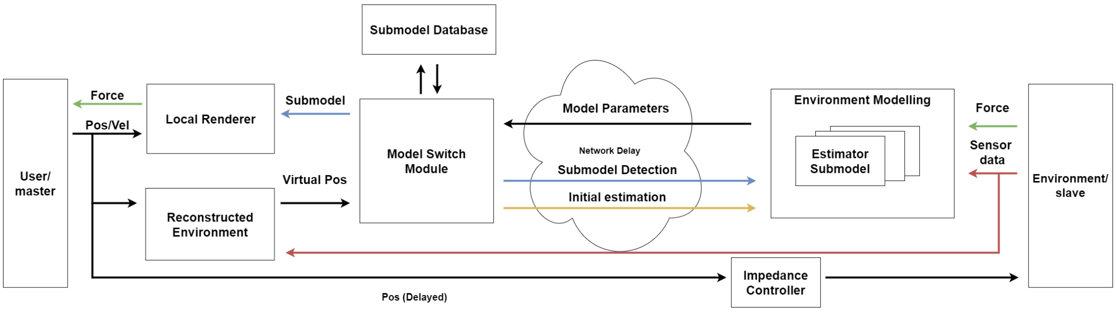


Figure 3.3: Proactive System schematic, detection based on position and kinematics constraints of the environment

4 Environment Modelling Design

4.1 Introduction

The proposed sub-models for manipulating an object in a dynamic environment are described in detail in the following section. This section covers the design decisions that are imposed to define the relationship between each sub-model and couple the kinematics and dynamics of the environment in order to implement such sub-models in the model switching strategies of chapter 3. This chapter begins by examining the examples of human activities where telemanipulation of objects is applicable, how these activities can be synthesised into a collection of simple tasks, and finally, how these can be classified as a specific type of dynamics and the information required to identify them. During manipulation, it is expected that the operator will grasp an object or tool that will interact with the environment. To illustrate these interactions Catoire et. al. [3] proposes a list of basic tasks that are found in each activities performed in telemanipulation of inspection, maintenance and repair. A basic task performs the same movement independently of the context of the activity, [3] defines eight basic tasks for manipulation:

a Screwing an object off another object

b Screwing an object on another object

c Placing a constrained object

d Removing a constrained object

e Placing a freely movable object

f Removing a freely movable object

g Following the external structure of an object

h Following a visual trajectory

While some of the basic tasks have a clear description of what type of dynamics are sensed in its execution, basic tasks c) to f) can be seen as transitions between two types of dynamics. In this way, the basic tasks are the reproduction of a type of dynamic or the transition between two types of dynamics. Figure **??** shows this relationship, it proposes four types of dynamics, sub-models, where tasks c) to f) are transitions, a) and b) are a screw constrainted sub-model and tasks g) and h) are placed in multiple models due to its vague definition.

An additional analysis of these basic tasks is to group them based on its motion and dynamics to fit them in the description of an environment with multiple dynamics. Table 4.1 displays this analysis, activity g) is simplified and it is supposed that the external structures have a flat, smooth and rigid geometry.

Activity	Dynamics	Motion	Constrainted to	Location
a & b	Friction dynamics [4]	screw joint translation	1 rotation & 1 translation	In contact with a screw hole
c & d	Linear Force: Inertial & friction dynamics	linear translation	1 translation in the direction of the cavity	Inside or in contact with a constrained object
e & f	Forces and torques: inertial dynamics	rotational and linear	no constraints	In free space
g & h (in contact with an object)	Forces and torques: inertial and friction dynamics	rotational & linear	2 translation and one rotation	In contact with a structure
h(in the free space)	Forces and torques: inertial dynamics	rotational & linear	no constraints	In free space

Table 4.1: Analysis of the basic tasks

This analysis presents the possibility that we can model the environment by describing each sub-model by kinematics constraints and the type of dynamics they experienced in their unconstrained motions. During a teleoperation, the user will experience freedom or limitation in its DOF and simultaneously a change in the shape of the dynamics in the unconstrained DOF. In this case, understanding these kinematics constraints of each sub-model, provides a link between each sub-model that can be apply as the source of information for the model switching design in section 3.1.2. The interconnection of the sub-model is used to predict the dynamics changes and allow the MMT system's model-switching functionality.

Table 4.1 and figure **??** grouped the basic tasks into four types of dynamics. Tasks a) and b) are grouped together in the named screw constrained sub-model which is the most complex model in terms of dynamics and type of motion, screwing. Given the scope of this study, it has been decided that task a) and b) won't be modelled as its incorporation do not provide an additional value over the other tasks to explore model switching in a MMT system in an environment with multiple dynamics.The remaining tasks include an inertial component, which represents the object's mass, and a friction component when they come into contact with other environmental objects. Additionally, the table categorised the activities based on where they would occur in the environment and the constraints that would apply to teleoperation. Depending on where the activity is, it may take place in free space, where there are no constraints, inside or in contact with a constrained object, where the motion is restricted to one degree of freedom, or when it is in contact with a structure, where motion is restricted to three degrees of freedom. Real environments are more complex than the description provided; for example, structures are not always flat and confined objects are not always restricted to a single axis of motion. In contrast, this simplified description of the environment for a manipulation task allows us to provide for both model switching design, first different dynamics models to process, and second the application of location and kinematics constraints to determine which dynamic model to select. Based on those reasons, three sub model are proposed to model task c) to task h):

- Planar Friction Submodel (3DOF)
- Inertial Submodel (6DOF)
- Cavity Friction Submodel(1DOF)

4.2 General Assumptions

We require the following assumptions if the kinematics constraints and position of the telemanipulation are used as sources to identify submodels.

- The object is assumed to be rigid and fixed to the gripper of the slave device, therefore the velocity and acceleration of the object are assumed to be equal to the slave device end-effector.
- The position of the manipulated object, its body's center of mass, coincides with the end-effector frame of the slave device w.r.t the fixed framed of the world located in the base of the slave device: $p_{object} = [x^w_{ee}, y^w_{ee}, z^w_{ee}]$.
- The environment's objects are rigid and its geometries have rectangles surfaces.

This assumptions guarantees that the kinematics of the manipulation remain constant during the whole manipulation and so they can be measured and used by the MMT system. For the same reason, it is assumed that during telemanipulation, neither the surroundings nor the objects will deform.

4.3 Planar Friction Submodel (3DOF)

This submodel describes the dynamics present when the object enters into contact with a surface. A surface is defined as a flat, limited geometry whose dimensions are significantly larger than the size of the telemanipulated object. The object's degrees of freedom (DOF) while in contact with a surface are restricted to a 2D linear movement that follows the plane and a rotation in the plane's normal direction. While the object maintains contact with the surface or is within the dimension of the surface, the telemanipulation experiences friction and inertial dynamics when it moves in the directions of the surface.

4.3.1 Submodel assumptions

The interaction between the manipulated object and an object that present a planar friction submodel has the following assumptions:

- The surface of the object can be found in any orientation.
- The model becomes invalid if the contact point is lifted or outside the surface's limits
- The object cannot penetrate through the surface that is considered rigid and insensitive to the applied forces.

These assumptions indicates that the potential type of elements in the environment that will fit this submodel are rigid big objects such as tables and walls.

4.3.2 Kinematic constraints and geometrical limitations

This model provides three degrees of freedom, it has two linear movements along the x and y axis of the plane that follows and one rotation in the normal of the plane, see figure 4.1.

The object cannot is constrained in the normal negative direction of the plane as described in the assumptions.However, it is allowed to move outward from the plane that will remove the contact point between object and plane, thus, invalidating the planar friction sub-model.

The sub-model is also invalidated if it reaches its geometrical limitations. The limits of the model are given by the dimensions of the finite plane that represent the larger object surface. If the object goes beyond these limits the model is also invalidated. To evaluate this situation the object is tracked inside the dimensions of the finite plane. For both the kinematics and geometrical limitations it is necessary to know:

- The dimensions of the finite plane, given by four 3D points(x,y,z) that defines the limits of the plane.
- The position of the manipulated object (x,y,z).

4.3.3 Dynamic equations

The dynamic equation for the given sub-model follows the friction modelling of Xu et. al [21], [22] and adapted to the orientation of the surface. The linear forces equation is described in eq. 4.1:

$$\vec{\mathbf{f}} = m \cdot \vec{\mathbf{a}} + c \cdot \text{sign}(\vec{\mathbf{v}}) + \vec{\text{d}}, \quad c = \mu \cdot m \cdot g \quad \vec{d} = m \cdot \vec{g} \tag{4.1}$$

where **a** and **v** are the object linear acceleration and velocity, respectively in the frame end-effector. f is the measured slave contact force, m is the object mass, c denotes the Coulomb friction between the object and the plane, μ is the dynamic friction coefficient,d is the gravity vector term, g is the gravitational acceleration constant and $\vec{g}$ is the gravity vector in the expressed frame. As in [21] is assumed that the maximum static friction is identical to the dynamic friction.

The rotational forces are described in eq. 4.2, where M is the applied torque on the object, I is the object rotation inertia, e is the torque produced by the ground friction and ω is the angular velocity in the normal direction of the surface measured in the end-effector frame.

$$M = I \cdot \dot{\omega} + e \cdot \text{sign}(\omega) \tag{4.2}$$

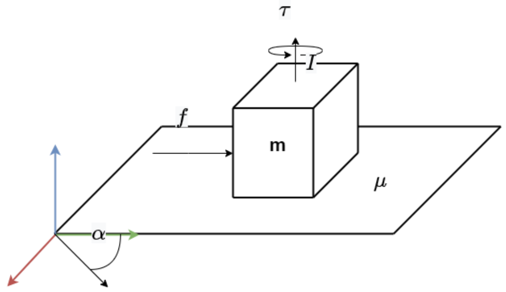


Figure 4.1: Planar friction model and its dynamic parameters

4.4 Inertial Submodel (6DOF)

An object in free space can move freely in many DOF as the slave manipulator posses, commonly 6 DOF. The environment does not add any restriction to its movement as in this state the object is not in contact with the environment and only with the manipulator which holds it.

4.4.1 Submodel assumptions

The operator in this sub-model is supposed to only be able to sense the object and its motion. Because of its unrestricted movement, the object's position is necessary in order to observe when it comes into touch with other objects in the environment and recognise a model switch. These presumptions are reasonable if the object can be held firmly with its grip and if only little movements occur during the procedure.

- The dynamics depends on the inertial parameters of the object, air drag is not modelled.
- The teleoperation activity is assumed to operate in low velocities and acceleration, therefore coriolis and centrifugal effects are not modelled.
- The end-effector frame of the slave device is considered to coincide with the body's center of mass as expressed in the inertial frame, so the distance to the center of mass is zero, simplifying the inertial dynamics.

4.4.2 Kinematic constraints and geometrical limitations

The free space model is characterized for the absent of kinematic constrains as it can moves and rotates in any direction.

With the same reasoning there are no geometrical limitations in this model except if it moves close to other geometries, entering in contact and then invalidating the model.

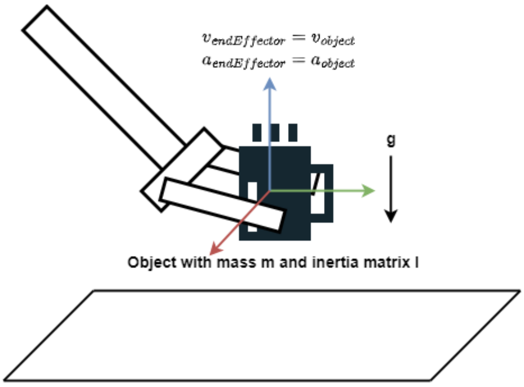

$v_{endEffector} = v_{object}$
$a_{endEffector} = a_{object}$
g
Object with mass m and inertia matrix I

Figure 4.2: Free space model and its assumptions

4.4.3 Dynamic equations

To calculate the dynamics of the object, the forces and torques measures are described by Euler-Newton equations $\boldsymbol{f}$ and $\boldsymbol{\tau}$:

$$\vec{f} = m(\vec{a} - \vec{g}) \tag{4.3}$$

$$\tau = I\alpha \tag{4.4}$$

where m is the object mass, **a** and α are the object linear and rotational acceleration in the end-effector frame, respectively, g is the acceleration of gravity and I is the moment of inertia about the center of mass.

4.5 Cavity Friction Submodel (1DOF)

This submodel represents the environment during the execution of basic tasks c) and d). A constrained object is defined as a geometry whose surfaces have cavities which are the space where the interaction with the object is performed. An example of this type of geometries are holes or objects with pockets and the well known peg-in-hole interaction. This sub-model describes the dynamics of a manipulated object when it is introduced in a cavity that match the dimension and shape of the object.

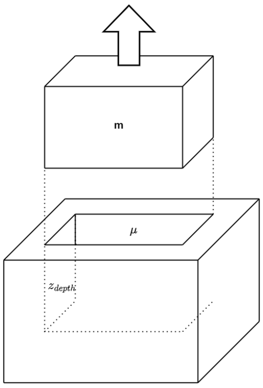


Figure 4.3: Case of hollow objects and its parameters

4.5.1 Submodel assumptions

The interaction between the manipulated object and a constrained object with a cavity has the following assumptions:

- The cavity has finite depth and the object can touch its end.
- Through holes are not considered in this scenario.
- The position of the cavity is fixed, and it is not perturbed by any force applied to it.
- The orientation of the cavity of the constrained object is limited to be in the Z axis of the fixed world frame to simplify the equations of motion.
- Given that we expect the object's material to be homogeneous, the friction dynamics inside the cavity are equal on each of its sides.

This model's design decisions are intended to present a realistic scenario in which the slave device can insert or remove the object from the cavity. To better distinguish its dynamics and kinematics constraints from the planar friction model and ensure that the gravity term is always present in this form of interaction, the cavity's position is fixed to the Z axis of the fixed world frame.

4.5.2 Kinematic constraints and geometrical limitations

When the object enters in the cavity its movement is constraint in every degree of freedom except in the one used to enter on it. This 1DOF is linear, e.g placing the object in a tight container. The movement of the object is restricted to the direction of the cavity. The linear movement is limited, it cannot go further than the depth of the cavity in the inward direction and it can moves outward the cavity until the object completely exit the cavity, invalidating the model. The geometrical limitation of this model is the depth of the cavity that represent the space of this model. For both the kinematics and geometrical limitations it is necessary to know:

- The external dimensions of the constrained object, given by four 3D points(x,y,z) that define the upper part of the object.
- The dimensions of the cavity, inner dimensions, given by the length, width and depth of the cavity and centered in the plane defined by the four 3D points of the constrained object.
- The position of the manipulated object (x,y,z).

4.5.3 Dynamic equations

The dynamics of the cavity submodel are the modification of the planar friction model, constrained to one degree of freedom and fixed to vertical cavities where the gravity term is always present. The described dynamics are shown in eq. 4.5, the first part is the inertial component, the second term is the Coulomb friction that is the Euclidean norm of the forces sensed perpendicular to the force in the direction of the cavity; $\mathbf{a_z}$ and $\mathbf{v_z}$ are the object linear acceleration and velocity expressed in the end-effector, m the object mass, μ the dynamic friction coefficient and g is the gravity term.

$$\mathbf{F_z} = m \cdot a_z + \mu \cdot \text{sign}(\mathbf{v_z}) \cdot \|Fx + Fy\|_2 - m \cdot g \tag{4.5}$$

4.6 Submodels transitions

In this section, it is determined how sub-models can transiently behave in an environment with changeable dynamics using the kinematic constraints and locations of each sub-model. Both elements set where and how the sub-models are detected. With this perspective, the environment is seen as a finite state machine(FSM), where the sub-models are the states. The position of the object and the kinematics constraints of the environment are the inputs that trigger the transition between states. We apply both inputs to reinforce the detection of transitions as we assume the geometrical reconstruction of the environment is not perfect.It is assumed that the measured velocities are almost zero when the user tries to move in a constraint direction. Additionally, the kinematics constraints can be also identified by seeing its effect in the forces. As the environment is assumed rigid, when the object is constrained in one direction, it is expected that there will be a reaction force in that direction. This description also introduces the use of forces to identify transitions.

In the selected environment, basic tasks c) to f) are an example of what transitions the environment perceives. Transitions are detected using visual and haptic sensors that evaluate and recognise a change in the kinematics to switch to an appropriate model given the circumstances. Figure 4.4 displays the interconnection of the selected sub-models. The inertial sub-model is considered to be the central model. The other sub-models represent interactions with other geometries, thus its kinematics constraints, and when the interaction stops, the object is free of constraints going to the inertial sub-model. The transitions between the friction sub-models planar and cavity is not considered.It is considered that in case there is a transition between the two friction sub-models, the user will move first to the inertial sub-model, transition always through the free movement. In the case of transitioning from a surface to a hole, we expected the user to lift the object before entering the cavity, avoiding a direct transition between both friction models. In these terms, the selected environment has four transitions. All transitions consider the forces, position and velocity conditions.

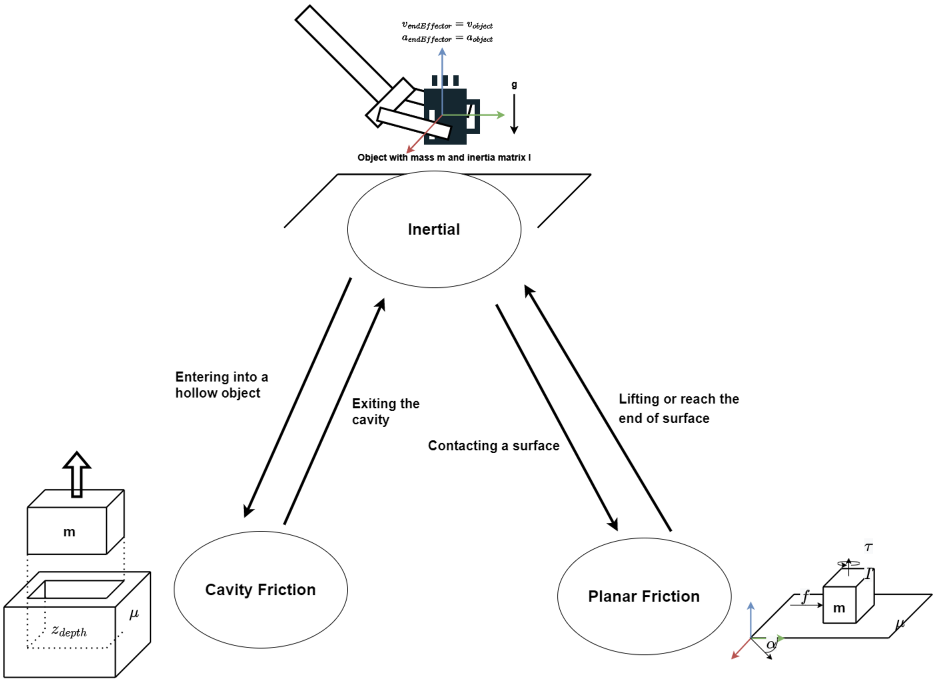

$v_{endEffector} = v_{object}$
$a_{endEffector} = a_{object}$
g
Object with mass m and inertia matrix I
Inertial
Entering into a hollow object
Exiting the cavity
Contacting a surface
Lifting or reach the end of surface
m
z_{depth}
μ
Cavity Friction
Planar Friction
τ
f
m
μ
α

Figure 4.4: Transition between models

4.6.1 Lifting an object from a surface or reaching the end of a surface (From 3DOF to 6DOF)

This transition describe the change of dynamics when the contact between the manipulated object and the surface breaks. There are two possible scenarios: that the object is lifted from the surface or travels out of surface dimensions and enters free space. In both cases the object frees all its DOF. Position and kinematics constraint changes are searched for to spot when the planar friction submodel switches to the inertial submodel. From the positional perspective, these conditions can be seen when the object position, p_{object}, is over the plane of the surface, $\pi_{surface}$, or out of the limits of $\pi_{surface}$ whose limits are displays in $\pi_{Constrainedsurface}$. Combining both cases, from the positional perspective the dynamics changes to the inertial model when the object does not belong to $\pi_{Constrainedsurface}$.

$$\pi_{Constrainedsurface} = \begin{cases} d(p_{object}, \pi_{surface}) < \Delta \\ a_u < x < a_l \\ b_u < y < b_l \end{cases}$$

where $d(p_{object}, \pi_{surface})$ is the normal distance between the object position, and the plane $\pi_{surface} = ax + by + cz + d$ that represent the surface of the object and it is computed from the points that belong to that surface. Δ is the threshold distance to evaluate if there is no contact point and depends on the dimension of the grasped object. a_u and a_l are the upper and lower limits of one dimension of the plane, e.g its length and b_u and b_l represent the limits of the other direction.

According to the kinematics restrictions, the velocity normal to the surface should be greater than zero if the object is lifted or falls from the limits. In the case of reaching the end of the surface it can be argue that the normal velocity wont be greater than zero as the manipulator maintains the position, however, it is assumed that the operator is not able to operate in a straight line during a teleoperation without additional support; therefore the normal velocity will be greater than zero.

From the viewpoint of the forces, the forces experience a change from pressing up against a surface, where the gravitational forces are compensated, to the removal of the contact with the surface and feeling again gravitational forces. To observe this transition from the force perspective the change in the sign of the force is observed. The main reason to not use a quantifiable threshold in the force measurement is that the user can push to the surface changing the magnitude of the forces making difficult to find an ideal threshold as the forces can change over time.The proposed conditions to switch from the planar friction to the inertial submodel are grouped as follows:

$$\text{Update model trigger } = \begin{cases} yes, & p_{object} \not\in \pi_{Constrainedsurface} \text{ and } |v_{n\pi_{surface}}(t)| > 0 \\ & \text{and } \text{sign}(f_{measured}(t)) \neq \text{sign}(f_{measured}(t-1)) \\ \text{no,} & \text{else} \end{cases}$$

where $p_{object} \not\in \pi_{Constrainedsurface}$ represented the mentioned positional condition, $v_{object}(n)$ is the velocity of the slave system in the normal vector of π and $f_{measured}$ is the remote force sensed in the normal direction of the plane of the surface $\pi_{surface}$.

4.6.2 Removing a constrained object (From 1DOF to 6DOF)

This transition represents basic task d). This transition frees the object DOF once it is out of the cavity and is not in contact with it. From the position perspective, the object is out of the cavity if its position is higher than the upper limits of the cavity. The object's velocities in the perpendicular direction play a significant role in detecting the kinematic constraints. After the object is released the movement in these directions is unconstrained and are expected to be higher than zero. As they come into touch with the cavity's sides in the cavity sub-model, the perpendicular forces $f_{measured_\perp}$ are primarily responsible for providing the friction component of the dynamics, eq 4.5. Since these forces are no longer in contact with the object once it has left the cavity, their measurements ought to be identical to the inertial dynamics of the handled object. This transition should have a significant change in the forces acting along those axes.

$$\text{Update model trigger } = \begin{cases} yes, & p_{n,object}(t) > c_u \text{ and } |v_{\perp,object}(t)| > 0 \text{ and} \\ & \text{and } f_{measured_\perp}(t) \ll f_{measured_\perp}(t-1) \\ & \\ \text{no,} & \text{else} \end{cases}$$

where c_u is the upper limit of the cavity, $p_{n,object}(t)$ the position of the object in the normal direction of the cavity, $v_{\perp,object}$ is the perpendicular velocities of the object and $f_{measured_\perp}$ is the remote force sensed forces perpendicular to the normal direction of the cavity.

4.6.3 Placing an object on a surface(From 6DOF to 3DOF)

The transition is noticed when the object makes contact with the surface, in opposed to its counterpart transition 4.6.1. In a similar manner, its kinematic constraints are likewise the opposite, its velocity is restrained, and it runs into contact forces, and so the gravitational terms is compensated by the constraint .Place an object into a surface can be done in various ways; it can be done delicately with little force exerted or aggressively with large force reaction. These circumstances make it impossible to generalise that a higher force in transitions is expected.

$$\text{Update model trigger } = \begin{cases} yes, & \text{if } d(p_{object}, \pi) < \Delta \text{ and } |v_{n\pi}(t)| \approx 0 \\ & \text{and } \text{sign}(f_{measured}(t)) \neq \text{sign}(f_{measured}(t-1)) \\ \text{no,} & \text{else} \end{cases}$$

where $d(p_{object}, \pi)$ is the normal distance between the object position, and the plane $\pi_{surface}$ that represent the surface of the object, Δ is the threshold distance to evaluate if there is no contact point and $v_{n\pi}$ is the velocity of the slave system in the normal vector of $\pi_{surface}$. $f_{measured}$ is the remote force sensed in the normal direction of $\pi_{surface}$.

4.6.4 Placing a constrained object (From 6DOF to 1DOF)

When the object is placed in a constrained object it enters in contact with the sides of cavity. It is expected that the friction forces of the sides of the cavity will increase the measured forces of the slave device. This increased in the forces is detected in the direction perpendicular to the direction of the cavity. From a geometrical perspective the object is below the upper limit of the cavity.

$$\text{Update model trigger } = \begin{cases} yes, & p_{n,object}(t) < c_u \text{ and } |v_{\perp,object}(t)| \approx 0 \\ & \text{and } f_{measured_\perp}(t) \gg f_{measured_\perp}(t-1) \\ \text{no,} & \text{else} \end{cases}$$

where c_u is the upper limit of the cavity, $p_{n,object}(t)$ the position of the object in the normal direction of the cavity and $f_{measured_\perp}$ is the remote force sensed forces perpendicular to the normal direction of the cavity.

Conditions for Submodel transitions			
Transition type	Position	Velocities changes	Forces changes
4.6.1	$\mathrm{p}_{object} \notin \pi_{Constrainedsurface}$	$\lvert v_{n\pi_{surface}}(t)\rvert > 0$	sign $(\mathrm{f}_{measured}(t)) \neq$ sign$(f_{measured}(t-1))$
4.6.2	$\mathrm{p}_{n,object}(t) > c_u$	$\lvert v_{\perp,object}(t)\rvert > 0$	$\mathrm{f}_{measured_{\perp}}(t) \ll f_{measured_{\perp}}(t-1)$
4.6.3	$\mathrm{p}_{object} \in \pi_{Constrainedsurface}$	$\lvert v_{n\pi_{surface}}(t)\rvert \approx 0$	sign $(\mathrm{f}_{measured}(t)) \neq$ sign$(f_{measured}(t-1))$
4.6.4	$\mathrm{p}_{n,object}(t) < c_u$	$\lvert v_{\perp,object}(t)\rvert \approx 0$	$\mathrm{f}_{measured_{\perp}}(t) \gg f_{measured_{\perp}}(t-1)$

Table 4.2: Overview of the submodel transitions conditions to switch

In summary, three sub-models are created to represent the manipulation task situation in the context of a multiple dynamic environment. Based on the manipulation's position, its velocity restrictions, and the forces that shift across sub-models, it provide a possible solution to identify the distinct dynamics of each sub-model. The inertial sub-model becomes the common sub-model for transitioning to the others. There is a constant relationship between the sub-models that offered four transitions. Observations of changes in position, velocity, and forces—which are summarised in table 4.2 can be used to identify this interconnection. This knowledge is utilised to construct a model switch process described in section 3.1.2, where the identification of the change of dynamics is designed based on the observation of the transitions, table 4.2 , and not by measuring the states that are proposed in figure 4.4.

5 System Design

5.1 Introduction

This chapter describes how the MMT framework incorporates the model-switching functionality. Two model switching techniques were presented in chapter 3; this chapter elaborates on exactly how they are implemented into an MMT system, the modules that make up the system, and the functions of each module to produce an MMT system that operates in an environment with changing dynamics. The first model switching strategy select a sub-model based on the outcomes of the model estimation process, the estimation error. Because the selection of a sub-model occurs during the estimation process, the implementation of this model switching method in a MMT system is called Reactive system. The second approach incorporates forces, kinematics and visual data of the environment to select a sub-model. In this approach, the selection occurs before the estimation process, the implementation of this model switching method in a MMT system is called Proactive system.

5.2 Reactive MMT System

5.2.1 System overview

The reactive system is formed by three modules that work together to integrate the model switching strategy based on the estimation error. It also has an impedance controller to ensure that the position of the slave device x_s follows the position of the master device x_m during the manipulation. The impedance control is selected to dynamically control the force output when it interacts with the environment, and to guarantee that the force position relationship is considered. The impedance controller is modelled as a virtual spring-damper in the 6DOF of the slave device. The general overview of this system is illustrated in figure 5.1.

The reactive system extracts the forces of the environment and motion of the slave device(position, velocity and acceleration) to initiate the estimation process in the model estimators module.The model estimators module outputs three sets of model parameters and estimation errors, one per each sub-model that describes the environment.The estimation errors are then evaluated in the model switch module, which selects the sub-model based on that type of information. The sub-model and its parameters are then handed to the local model module, which applies the sub-model dynamics to render the forces in combination with the position, velocities and accelerations of the master device. If the model switch module determines that another sub-model's estimation error is lower than the current sub-model's, the sub-model used in the local model switches.

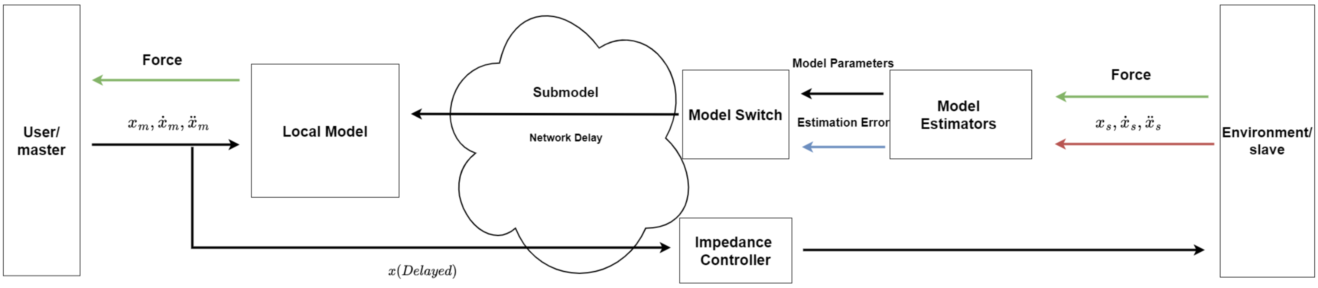


Figure 5.1: Overview of the Reactive MMT system

5.2.2 Local Model

The local model is the module in charge of the direct interaction with the user. Its main function is to render the forces and torques of the selected sub-model using the position, velocity and acceleration of the master device, its schematic is shown in figure 5.2. The local model contains the three dynamics equations that describe the environment proposed in chapter 4. The selected sub-model is represented by the right input of the module in the figure 5.2, which supplies the parameters of that sub-model as well as an identification for the module to know what dynamic equation to compute.

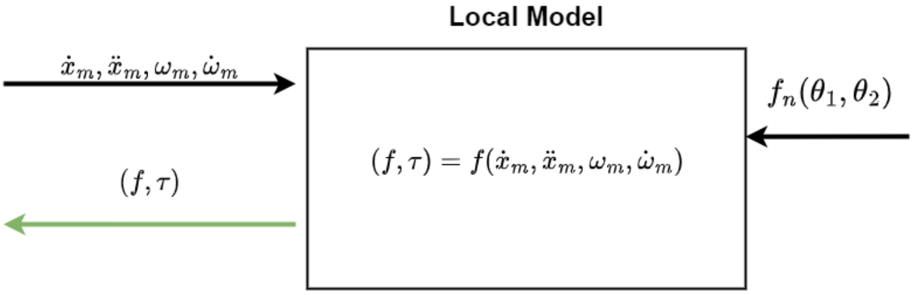

Local Model
$\dot{x}_m, \ddot{x}_m, \omega_m, \dot{\omega}_m$
$f_n(\theta_1, \theta_2)$
$(f, \tau) = f(\dot{x}_m, \ddot{x}_m, \omega_m, \dot{\omega}_m)$
(f, τ)

Figure 5.2: Local model in the Reactive system

5.2.3 Model Estimators Module

The model estimators module contains three RLS estimators that provides the parameters and the estimation error of each sub-models based on the information of the environment. Once the system is initiated all sub-model estimators start to run until the end of the teleoperation is ended. Each estimator uses the forces of the environment and the velocity and acceleration of the slave device to output the parameters of such model, figure 5.3 displays its schematic. From the related work section 2.1, the Regressive Linear Estimation Algorithm (RLS) is the chosen as the algorithm to estimate parameters in this design. The model has to be given in the linear-in-parameter form:

$$y_k = \phi_k^T \boldsymbol{\theta}_k$$

with $y_k \in \mathbb{R}$ the system output at a time step $k, \phi_k \in \mathbb{R}^n$ the input regressor, and $\theta_k \in \mathbb{R}^n$ the parameter vector at a time step k. By minimizing the cost function V_k

$$V_k = \frac{1}{2} \sum_{i=0}^{k} e_i^2$$

depending on the estimation error $e_k = y_k - \hat{y}_k$ between measured and estimated system output, the estimate of the parameter vector $\hat{\theta}$ minimizing the cost function is found. The solution is a recursive set of equations:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_k &= \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{\kappa}_k \left(y_k - \phi_k^T \hat{\boldsymbol{\theta}}_{k-1}\right) \\
\kappa_k &= \boldsymbol{P}_{k-1} \phi_k \left(\lambda \boldsymbol{E} + \phi_k^T \boldsymbol{P}_{k-1} \phi_k\right)^{-1} \\
\boldsymbol{P}_k &= \lambda^{-1} \left(\boldsymbol{E} - \boldsymbol{\kappa}_k \boldsymbol{\phi}_k^T\right) \boldsymbol{P}_{k-1}
\end{aligned}$$

where $\kappa_k \in \mathbb{R}^n$ is referred to as adaptation gain vector and $\boldsymbol{P}_k \in \mathbb{R}^{n \times n}$ as covariance matrix. The parameter $\lambda \in [0;1]$ is known as forgetting factor. It can be used to weight recent measurements differently to older measurements.

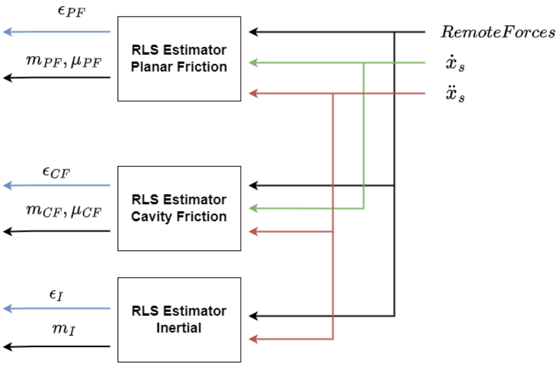


Figure 5.3: Model Estimators in the Reactive system

5.2.4 Model Switch Module

The model switch module selects the sub-model that is sent to the local model and its parameters. The selection of a sub-model acts as a control entry for sending model parameters to the local model. The methodology used to select the sub-model has two requirements: the estimation error should be minimal to provide accurate parameter estimation and fast to send model parameters and provide force feedback to the user. Knowing these conditions, the estimation error evaluation should be designed to be lower than a given threshold and remain under it, over a period of time. This method ensures a consistent error and provides a quick response. What threshold to select is a complicated task since, due to the differing dynamics, the estimators may not produce equivalent estimates.To avoid a set threshold that would have to be calculated through trial and error, it is chosen to use the relative estimation error based on the evaluation tools of[12].

Relative estimation error standard (REE)

REE evaluates if the estimation error recorded during N samples and divided by the highest estimation error encountered is lower than 1%. This metric works by assuring that the error remains steady and low in a window of time, and by including a force threshold that is the scaling of the highest estimation error. The design metric is shown in eq. 5.1 ,if the relative error is 100 times lower than the maximum error that is measured in each sub-model estimator, then the sub-model is selected.

$$REE_e = |\frac{\frac{1}{N}\sum \epsilon_k}{\epsilon_{max}}| \cdot 100\% \tag{5.1}$$

where ϵ_k is the estimation error at k, ϵ_{max} the maximum estimation error and N the number of samples of ϵ_k taken for evaluation.

The REE is applied individually and parallel to the three sub-model estimation errors. Once one sub-model is selected, its model parameters collected during the evaluation time N samples are averaged and send to the local model. A simpler alternative to the proposed technique is to select the sub-model with the lowest error when compared to the other submodels. This approach will work for the model switch functionality but given that it is coupled with the estimation process of the system it wont satisfy the last function. This approach does not guarantee that the error will be small enough to provide a good estimation of its submodel parameters, as its main requirement is to provide a better results than the others submodel estimation errors.

A module diagram is shown in figure 5.4. On the left, the selected sub-model and its parameters are sent to the local model, which is the REE standard's results. Its main inputs, displayed on the right part of the figure, are the estimation error of the three sub-models $\epsilon_{PF}, \epsilon_{CF}, \epsilon_I$ and the estimated parameters the mass and dynamic friction from each sub-model respectively at k time.

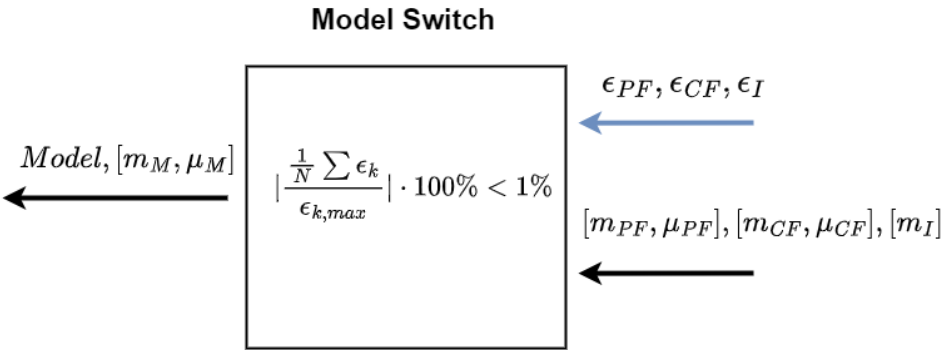

Model Switch
$\epsilon_{PF}, \epsilon_{CF}, \epsilon_I$
$Model, [m_M, \mu_M]$
$|\frac{\frac{1}{N}\sum \epsilon_k}{\epsilon_{k,max}}| \cdot 100\% < 1\%$
$[m_{PF}, \mu_{PF}], [m_{CF}, \mu_{CF}], [m_I]$

Figure 5.4: Model switch module in the Reactive system

5.3 Proactive System

5.3.1 System overview

The proactive system uses visual data, the motion of the slave device and forces to detect the appropriate sub-model for the current dynamics of the environment. How this information is used to inform model switching is described in section 4.6. The proactive system uses this type of information to detect transitions between the three sub-models.The knowledge of those conditions is the leading feature of the proposed design. Figure 5.5 displays the general schematic system, compared with the proposal in figure 3.3, the diagram is adapted to include the modules of the system that are allocated in the remote and local part of the MMT system.

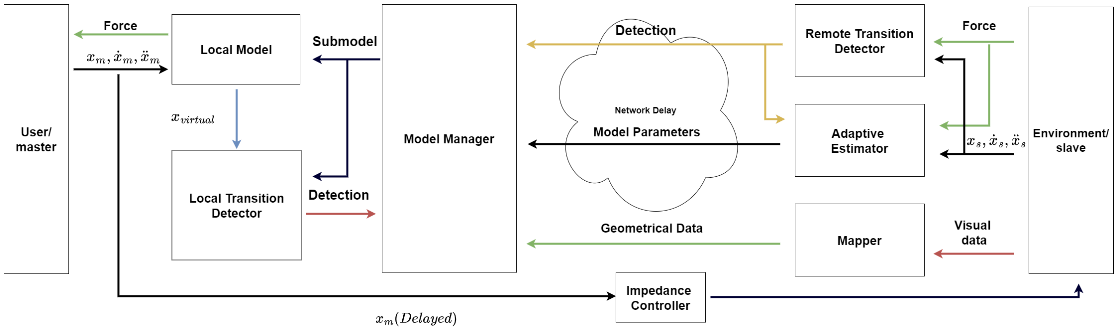


Figure 5.5: Overview of the proactive MMT system

To satisfy the design conditions described in section 3.1.2 the system is composed of six modules and an impedance controller as in the reactive system. In the remote side of the system the forces, position, velocities, accelerations of the slave device and visual data extracted from the environment are used in three modules: the mapper, the adaptive estimator and remote transition detector.The mapper module uses the visual data of the environment to create a virtual reconstruction of the environment that is send to the model manager module. The adaptive estimator uses the velocities, accelerations and forces to estimate the model parameters of each sub model. The last module, the remote transition detector, is part of the model switching functionality of this design. It works as a contact detector with the environment by using the forces and velocities of the environment. It also estimates the kinematic constraints of the environment to identify which sub-model causes them. The detection of the remote transition detector is sent to the model manager and to the adaptive estimation where is used as a control signal.

The output of the three described modules is sent to the model manager module which is the central module in this design. The model manager has two main functions storing the gathered information of the environment and perform the final decision in the model switch functionality. The model switch strategy in this system design combines the model manager and the local and remote transition detectors. The transitions detectors diagnose the type of submodel based on environmental data and the position of the user in the reconstructed remote environment, section 4.6; and the model manager collects that information and the incoming parameters estimation to formulate the selection. The local transition detector evaluates if the virtual position of the system enters in contact with any of the objects present in the reconstructed environment. The virtual position is the position of the master device in the reconstructed environment imitating the position of the remote environment. The last module. the local model, renders the forces to the user from the the selected sub-model and displays the virtual environment, that is the reconstruction of the remote environment.

5.3.2 Local Model

The local model is the module in charge of the direct interaction with the user. Its main function is to recreate the dynamics of the remote environment to be displayed the user and to compute the virtual position of the system. The virtual position, $x_{virtual}$ integrates the master device position x_m in the reconstructed environment to provide the position data necessary for the local transition detector module. To achieve its function the local model has three inputs and two outputs. Figure 5.6 shows the interchange with the module.

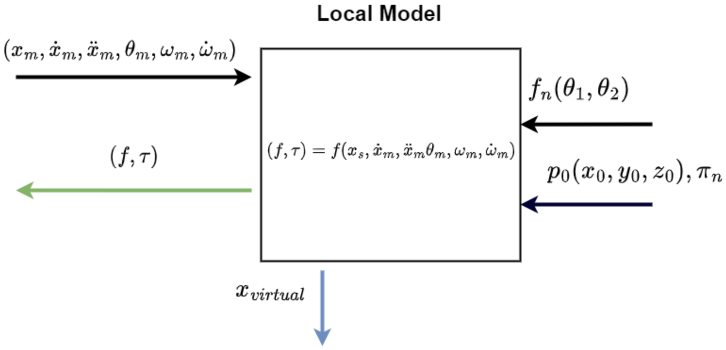

Local Model
$(x_m, \dot{x}_m, \ddot{x}_m, \theta_m, \omega_m, \dot{\omega}_m)$
$f_n(\theta_1, \theta_2)$
$(f, \tau) = f(x_s, \dot{x}_m, \ddot{x}_m \theta_m, \omega_m, \dot{\omega}_m)$
(f, τ)
$p_0(x_0, y_0, z_0), \pi_n$
$x_{virtual}$

Figure 5.6: Schematic of the local model

The first input is the motion of the master device that is used to render the forces of the selected sub-model. Given the three possible type of dynamics, the local model needs the linear and rotational position, velocity and acceleration of the master device.

The second input is the sub-model that represents the remote environment at the current time. The sub-model input provides the dynamics of the sub-model. The dynamics function depends on parameters obtained from the remote environment and the input motions that generate the forces and torques in combination.

The third input is the sub-model's geometrical properties that come with the second input. It provides the position where the sub-model is detected in the remote environment and the geometrical definition of the object, it can be a defined plane for the 3DOF or cavity dimensions 1DOF sub-models, that is assigned with the sub-model. The geometries of the sub-models are used in the local model to render the constrained forces if the user moves in a direction where the physical dimension of the environment restricts movement.

In a multi-model environment, physical space can be used to locate the boundaries of a model. In the case of a free mass model (6DOF), the model is valid within the free space in the environment. Once it enters contact with other geometries present in the environment, the free mass model is not valid as it becomes constrained. In the planar friction model (3DOF) the model is valid while the object maintained contact with the surface. If the operator decides to move up or out of the surface the object removes its constraint and the model is not valid.

For these cases, the environment is mapped to evaluate the physical limitations of the sub-model, therefore to render the constrained forces that will limit the motion in the master device, replicating the motion of the remote environment. To update the virtual position, $x_{virtual}$, the master position, x_m, is combined with the position where the submodel is detected, p_0, that is the robot's end-effector position.

The combination of these three inputs results in the two outputs of the local model:

- Generate force feedback based on the current model.

- The position of the robot end-effector in the virtual environment to evaluate the physical limitations of the model.

Compared with the local model in the reactive system, this module renders the constrained forces using the geometrical information of the selected sub-model.The calculation of constrained forces provides the user with the ability to sense the kinematics constraints in the virtual environment. As an example lets assume the planar friction model from section 4.3. The requirements to render the forces and torques of this model are:

- Input one: user motions $(x, \dot{x}, \ddot{x}, \theta, \omega, \dot{\omega})$
- Input two: type of sub-model dynamics, planar friction and its parameters $[m, \mu, I, e]$
- Input three: initial position in the remote environment $p_0[x_0, y_0, z_0]$ and the defined plane π that represent the space of the sub-model.

The plane π is bounded by the physical limitations of the surface, in this example the plane is an horizontal plane:

$$\pi = \begin{cases} z = 0 \\ a_u < x < a_l \\ b_u < y < b_l \end{cases}$$

Given the model parameters $[m, \mu, I, e]$ the three unconstrained forces are calculated, taking into account the physical boundaries of the plane.

$$f_x = \begin{cases} m \cdot \ddot{x} + \mu \cdot m \cdot g \cdot \operatorname{sign}(\dot{x}) & \text{if} \quad a_u < x < a_l \\ m \cdot \ddot{x} & \text{if} \quad \text{else} \end{cases} \tag{5.2}$$

$$f_y = \begin{cases} m \cdot \partial \ddot{y} + \mu \cdot m \cdot g \cdot \operatorname{sign}(\dot{y}) & \text{if} \quad b_u < y < b_l \\ m \cdot \ddot{y} & \text{if} \quad \text{else} \end{cases} \tag{5.3}$$

$$\tau_z = I \cdot \dot{\omega_z} + e \cdot \operatorname{sign}(\omega_z) \tag{5.4}$$

The constrained forces are computed to limit the degree of freedom of the given sub-model. Its dynamics are modelled by a contact model, spring-damper system that matches the impedance controller parameters, $K_{impController}$ and $B_{impController}$. In the given example, the plane is located in $z = 0$, the weight of the object is held by the physical plane. If the user pushes against the surface, the local model will render the constrained forces.

$$f_z = \begin{cases} K_{impController}(z_{plane} - z_{input}) - B_{impController}\, v_z & \text{if } z_{input} \le z_{plane} \\ -mg & \text{if } z_{input} > z_{plane} \end{cases} \tag{5.5}$$

$$\tau_y = \begin{cases} 0 & \text{if } \omega_y = 0 \\ K_{impController}\Delta\theta_y - B_{impController}\omega_y & \text{if } \omega_y \neq 0 \end{cases} \tag{5.6}$$

$$\tau_x = \begin{cases} 0 & \text{if } \omega_x = 0 \\ K_{impController}\Delta\theta_x - B_{impController}\omega_x & \text{if } \omega_x \neq 0 \end{cases} \tag{5.7}$$

5.3.3 Local transition detector

The local transition detector is the module that checks the positional conditions of the model switching functionality based on the transitions described in section 4.6. The module follows the virtual position of the manipulation in the reconstructed remote environment. It evaluates if the current sub-model triggers any of the transitions updated models' conditions based on its position. Figure 5.7 shows its design architecture. The local transition detector has three inputs and one output. As inputs, it has the virtual position given by the local model, the current sub-model information, and the set of detected objects in the environment with an assigned sub-model from the mapper module. The output of the module is the identification of the sub-model based on the position of the virtual environment.

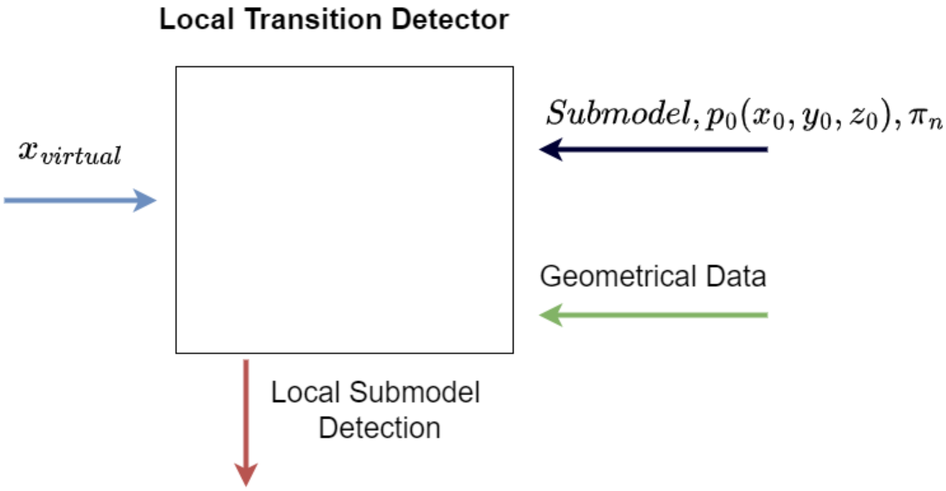

Local Transition Detector
$x_{virtual}$
$Submodel, p_0(x_0, y_0, z_0), \pi_n$
Geometrical Data
Local Submodel Detection

Figure 5.7: Schematic of the Local Transition detector

To illustrate its functionality, figure 5.8 represents the geometrical data obtained from the remote environment that comes from the mapper module. The current sub-model is inertial, the object can move freely. The virtual position is represented in the figure as a 3D point (black star). The reconstructed environment also includes a surface, colored green, and a red box that exemplifies a cavity, surrounded by its external dimensions colored in yellow. Suppose the virtual position enters in the red zone of the cavity object. In that case, the local detector model will trigger the transition conditions, to the cavity friction, and indicate its detection,based on the geometrical properties, to the model manager module.

5.3.4 Remote Transition Detector

The remote transition detector is the complementary module of the local detector. It complements the model-swiching function by interpreting the kinematics constraints in the remote environment based on the transitions described in section 4.6. The remote transition detector can be seen as contact detector, that uses velocities and the forces to diagnose what sub-model matches the remote environment. Its output is the identification of the sub-model based on these inputs and the knowledge of table 4.2. The system adds the end-effector position where the contact is detected and the sub-model identified. This position is used to update the initial position p_0 in the local model and to update the geometries in case there is a mismatch between reconstructed environment and the remote environment.

The sub-model identification is sent to two modules, to the model manager to complete the model switch functionality and to the adaptive estimator module to activate or deactivate the sub-model estimators. The last functionality is one of the main feature of the model switching

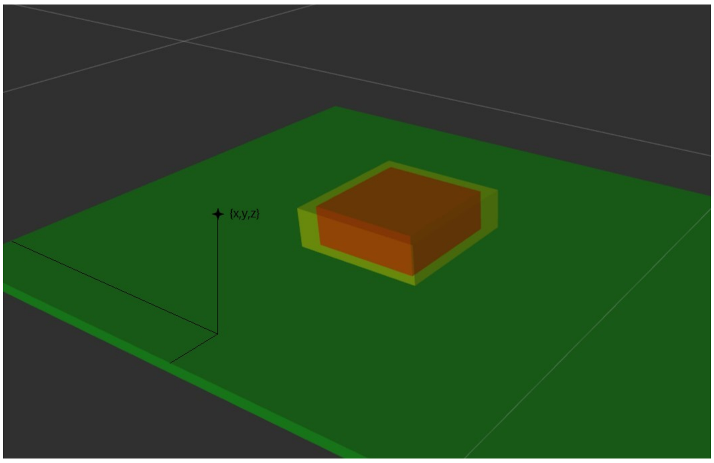


Figure 5.8: Reconstructed geometries of the remote environment

strategy described in chapter 3, where the model switching strategy controls the estimation process.

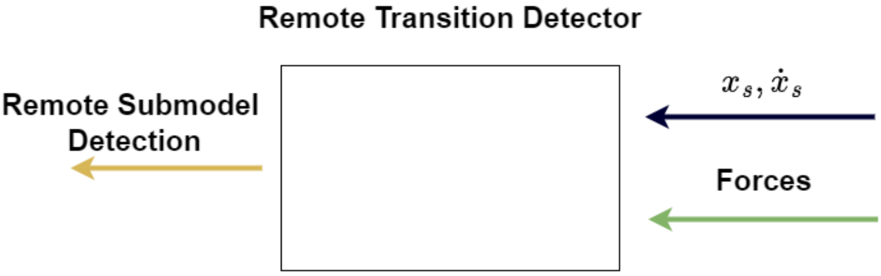


Figure 5.9: Schematic of the Remote Transition detector

5.3.5 Model Manager

The model manager module gathers the information of the environment to select the sub-model that matches the dynamics of the remote environment. The first steps of the model manager is to collect all the information necessary for the selection. This information consist of the geometrical data, that is the reconstructed remote environment, and the information incoming from both transitions detectors modules. The reconstructed environment is a set of objects that are detected by the mapper module. Each object has assigned one of the type of sub-models described in section 4, a physical position in the remote environment and its reconstructed geometry. The model manager uses the geometrical data to create a map of the environment, this map is used to locate the sub-model assigned to the objects of the environment to initiate the model switch functionality. As the map is based on the assumption that visual data can assign objects with a certain type of dynamics, the system is designed to verify this assumption using the detections from both transition modules. The local detector indicates that the object is touching an element of the virtual environment while the remote detector verifies the contact in the remote environment and the type of dynamics based on the kinematics constraints that it observes. Once the map is constructed, the model manager begins the model switch functionality, this implies that the system assumed that the remote environment will remain the same during the model switching process. In this design the model switch sends the sub-model dynamics, its assigned geometry and an initial position of the sub-model location to the local model module, see figure 5.10

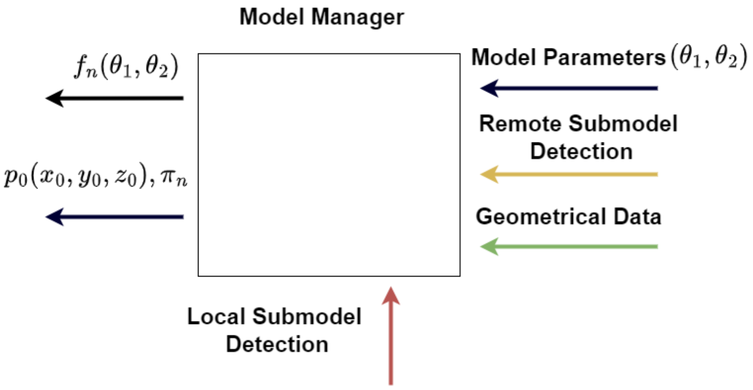

Model Manager
$f_n(\theta_1, \theta_2)$
$p_0(x_0, y_0, z_0), \pi_n$
Model Parameters (θ_1, θ_2)
Remote Submodel Detection
Geometrical Data
Local Submodel Detection

Figure 5.10: Schematic of the Model Manager

To verify the assumption that the reconstructed environment, based on the visual data can assign objects with a certain type of dynamics, the model manager confirms that both detections identify the same sub-model. The application of both detectors for this function varies on the available information from the remote environment. The next subsection explain in detail how the the model switch strategy.

Model switch strategy

The model switch strategy integrated into this design selects the sub-model before the estimation process. The model manager first detects the correct sub-model based on the transition detectors, and then it begins to read the incoming parameters from the adaptive estimator module. To follow these criteria, the model manager does a model switch in three modes depending on the type of information available it uses and that it sends to the local module: detection, update and recall.

Detection mode

The detection mode uses the detections of two the transition detectors modules. The model manager selects a sub-model because the detections of the transition modules correlate with the map of the environment. As the map of the environment has assigned the type of sub-model based on an assumption, the model manager waits for the remote and local transitions to confirm that assumption. In this mode, the model manager send to the local model, the dynamic model and its assigned geometrical properties. This type of model switch happens before the estimation process is finished, so the parameters sent to the local model are an initial estimation, e.g for friction models, it assumed that there is friction in the environment, so the dynamic friction coefficient of the sub-models is initialized to $\mu = 0.4$.

Update mode

The update mode occurs after the estimation process outputs accurate parameters of the sub-model. This model switch is the update of the detection mode and its main function is to provide an accurate update of the sub-model parameters that is currently being rendered in the local model. In this case, the model manager sends to the local model an update of the parameters of the selected sub-model. The updated parameters are saved in the environment map, so if the user enters in contact again with the same object, the sub-model is already updated and no estimation process is necessary.

Recall mode

When the teleoperation contacts again a known sub-model whose parameters have previously been updated, the recall mode is applied. This mode is based on the local transition detection; if the virtual position comes into contact with a known object, the model manager switches to its sub-model using data from the database. In this case, no interaction with the remote environment is necessary as the verification of the virtual environment has already been done and the initial sub-model assumption from the visual data has been already confirmed in the Detection mode. As in the case of the detection mode, the model manager sends the dynamic model and assigned geometrical properties of the selected submodel.

This procedure is illustrated in figure 5.11. To represent the different processes before each model switch type, there are two previous possible functions before them the identification and update task. These tasks describe the type of information that the model manager is utilising to make a choice.The identification task listens to the detections. It uses both detections when the sub-model has not being confirmed, detection mode, or only the local detection in case of the recall mode. The update task waits for the incoming parameters of the sub-model. In the case of the detection model switch mode the identification and update task runs in parallel in case the user decides to move during the estimation process.

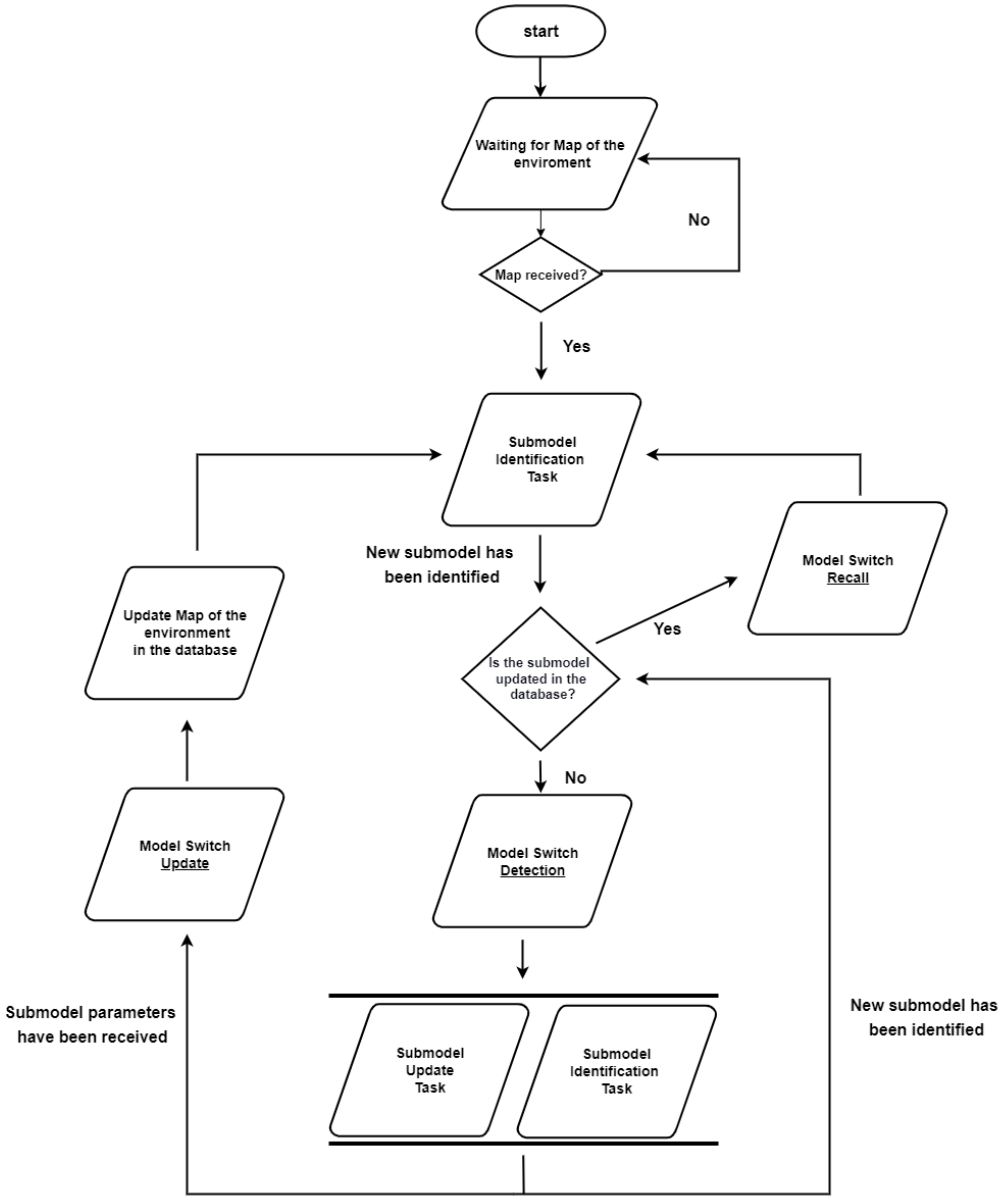

start
Waiting for Map of the enviroment
No
Map received?
Yes
Submodel Identification Task
New submodel has been identified
Model Switch Recall
Yes
Update Map of the environment in the database
Is the submodel updated in the database?
No
Model Switch Update
Model Switch Detection
Submodel parameters have been received
Submodel Update Task
Submodel Identification Task
New submodel has been identified

Figure 5.11: Workflow of the Model Manager

5.3.6 Adaptive estimator

The adaptive estimator module computes the parameters used in the dynamics of the sub-models. The adaptive estimator uses the forces, velocities and accelerations to estimate the model parameters of the three selected sub-models of the environment. The output of this module is sent to the module manager where the sub-model update is parameters.

The estimation method used is the same as in section 5.2.3. The main difference is that the estimators do not run in parallel. Instead of computing all RLS estimations, the adaptive estimator activates only the RLS estimator that matches the sub-model detected in the remote transition detector module. The estimators in this framework passes from one to the other. Therefore, the previous mass estimation, which is shared by all sub-models, is used as the initial measurement for the next estimation. Figure 5.12 depicts a graphic representation of this function. On the left part of the figure the last function is represented, the measurement evaluation. The method applied is the same used in the model switching and estimation process of the reactive system, described in section 5.2.4. This function evaluates if the model parameters obtained from the RLS estimators are accurate enough to be sent to the model manager. Compared with the reactive system, the adaptive estimator sends the parameters of each sub-model only once. The purpose for this is to avoid parameter changes while the estimating process is incomplete, implying that the model parameters stay constant during the teleoperation. Once the model parameters are considered accurate they are sent to the model manager and the mass parameter is sent to the estimators as an initial estimation of the mass.

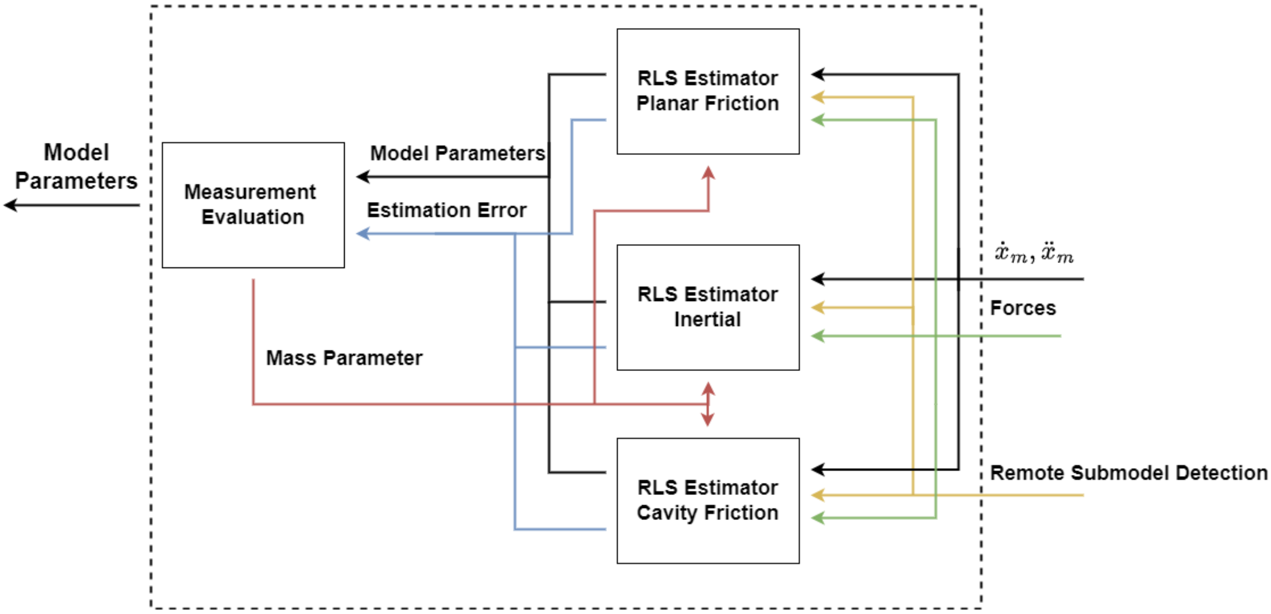


Figure 5.12: Schematic of the Adaptive estimator

5.3.7 Mapper

The mapper is the module that extracts the geometrical properties of the remote environment to reconstruct the remote environment in the local side of the system. The remote environment is registered in a set of objects that can be reproduced in the local side. Each object is labelled with a type of sub-model and is location in the remote environment. The mapper encounters objects with cavities and surfaces in the proposed remote environment. The mapper looks for features to classify into planar or cavity friction sub-model.For the case of free space, it is considered an inertial sub-model and is not processed. When the objects are labelled with their model, the physical dimensions of the geometries are also extracted to known the geometrical limitation that the system will use to render the forces. To compute the physical dimensions of the geometries the module extracts the four edges of the planar surface and the four edges of the constrained object and the depth, length and width of the cavity. Because of time constraints and the fact that the information extracted in the mapper, a set of points from each object, can be acquired separately from visual data, it was chosen to simulate the mapper data. Section 6 describes how this information was simulated.

6 Experiment Design

This chapter creates an experimental setup to put the model switching capabilities of the systems created in Chapter 5 to the test in a manipulation activity where time delays are not considered. The chosen manipulation task consists of three actions designed to feel multiple dynamics, namely the dynamics presented in Chapter 4. The three activities are:

- Moving an object against surface.
- Lifting and moving the object in the air.
- Placing and removing the object from the cavity of an object of the environment.

The dynamics of these three activities are represented in the system by modelling as a 1DOF sec.4.5, 3DOF sec.4.3 and 6DOF sec.4.4 respectively. Each activity is performed one after the other in the same experiment to create a variation in the dynamics of the environment, section 6.3 provides a protocol of the experiment in detail. The rest of the chapter offers two evaluation tools to define the accuracy and response time of the system in the specified environment in order to evaluate the capabilities of each system. Additionally, it discloses the parameters utilised in the estimation process for both systems, as well as a hypothesis about the expected outcomes from the evaluation tools.

6.1 Experimental Setup

To interact with the system from the user perspective the haptic device Omega6 is used as the master device of the system that only provides linear force feedback with a maximum force of 12N. The haptic devices allows 6DOF movement but as is not equipped with torque feedback; the experiment does not estimate the torque dynamics.

In the remote environment, the chosen slave device is the robot arm Franka Emika controlled with an impedance controller that provides a cartesian movement in the end-effector. Rotations in the slave device are restricted as the master device cannot provide torque feedback. To measure the remote forces, the ATI Mini40 Force sensor is used with a resolution of 0.08 N in the Z axis and 0.04N in the X and Y axis.The hardware communication and the software implementation is done in the Robotics Operating System (ROS) [13]. The ATI Mini40 is placed between the franka gripper and the robot arm as it is shown in figure 6.1.

The workspace of the haptic device Omega6 is 160 x 110 mm while the remote environment provided for telemanipulation is 500 x 1000mm. Due to the limitations of the physical workspace in the haptic device, the motion of the haptic device is only send when a button in the omega6 is activated. In this way when the omega6 workspace limits are reached, the user can disable the communication between the haptic device and the robot to reset the position of the haptic device. When the button is not pressed no position is sent to the remote environment and the local model stops the rendering of the virtual forces.

To allow the user to interact with an object in the experiment's three activities, an object that is capable of all of these interactions and that can be gripped by the robot's gripper must be designed. The selected object is a 3D-printed cube with a mass of 0.350kg and dimensions of 80x80x80mm. Given the size and materials available in 3D printing, the object is designed to be as heavy as achievable. The constrained object is a 3D-printed cube with an external dimension of 100x100x100mm and internal cavity with dimensions of 82x82x30mm. The constrained object is designed to allow some play between the manipulated object and the constrained environment and ease the insertion of the object.

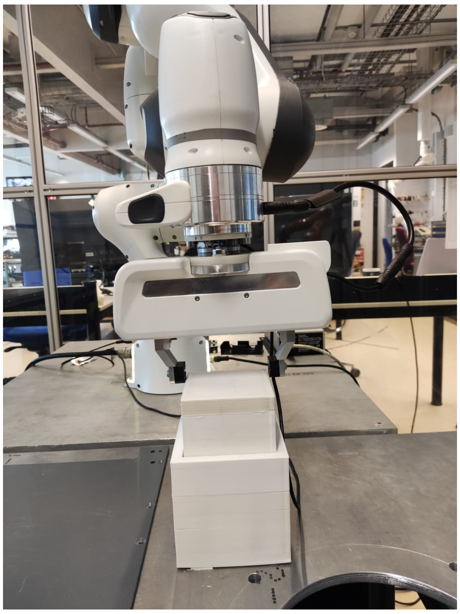


Figure 6.1: Setup of the Robot Franka Emika with the F/T sensor ATI Mini40. The force sensor is placed between the robot and the panda gripper

6.2 Remote Environment Reconstruction

The reconstruction of the remote environment was not created by the mapper module but instead it was simulated. The franka robot was used to evaluate the position and geometry of the objects in this experiment. The position of the object in the remote environment was determined by probing with the robot; the geometries were obtained from the 3D printing objects' CAD files.

6.3 Experimental Protocol

The goal of this experiment is to assess the model switching techniques implemented by the reactive and proactive systems in an environment with changeable dynamics. For both systems, it is proposed a manipulation task with no time delays, ilustrated in figure A.1. The protocol for this experiment is as follows:

1. The experiment begins when the 3D-printed cube is grasped and the robot is positioned in the red triangle show in figure A.1 which is the initial position of the robot in the x and y position and 400mm above the table.

2. The user moves in the free space until the system receive an estimation from the estimator to render the virtual forces of the inertial sub-model.

3. Next the object will be placed in the red diamond position of figure A.1, entering in contact with the table.

4. The user carries the estimation process as in 2) to obtain an estimation of the planar-friction sub-model.

5. The user moves the object in contact with the surface 200mm in the X positive direction of the workspace until reaching the green diamond position showed in figure A.1.

6. The user moves the object left and right 100mm to render the forces in the Y direction.

7. The user lift the object and place it above the constrained object represented by the green arrow in figure A.1.

8. The user introduce the object inside the constrained object and carries the estimation process for the cavity-friction sub-model. This process is done by pressing the object against a corner of the constrained object and moving up and down.

9. The user introduce the object inside to feel the end of the cavity and then the cavity of the constrained object.

10. The user moves the object to the black triangle show in figure A.1, 100mm higher in z direction over the constrained object.

11. The experiments ends.

By the end of experiment the system interacts with the different sub-models of the environment, following the order of: inertial->planar friction->inertial->cavity friction->inertial.

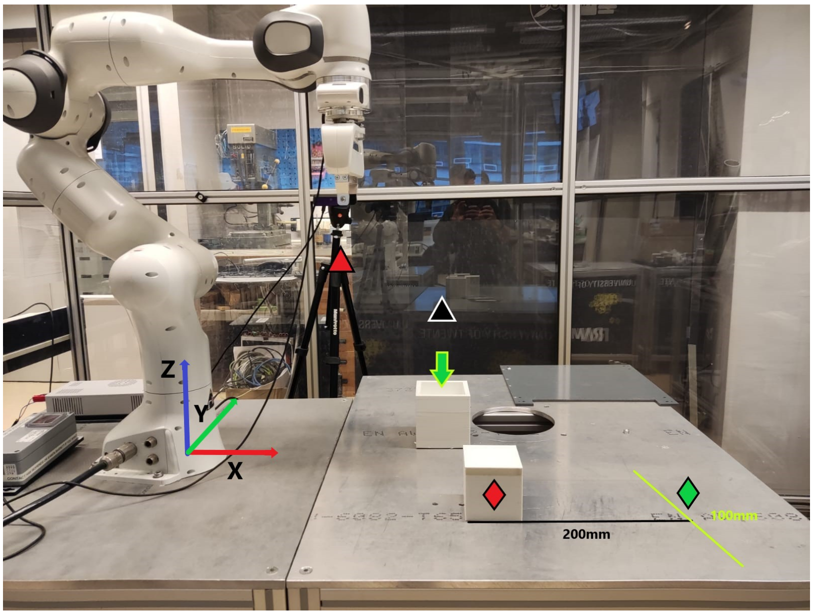


Figure 6.2: Physical setup for the manipulation test

6.4 Experiment Evaluations Tools

The steps previously described for this experiment can be divided into two types: an estimation process(steps 2,4 and 8) and an interactive process where the user experiences the dynamics of each sub-model(steps 3,5,6 and 9). Steps 7 and 10 are seen different for each system. The proactive system see them as a transition between known sub-models whose parameters are already estimated, e.g step 7 planar-friction -> inertial sub-model. The proactive system categorises these steps as an interactive process. The reactive system is not aware of a transition and needs to estimate the sub-model again, so steps 7 and 10 are a combination of both types.

6.4.1 Evaluations tools: Estimation process

In the estimation process we measure the convergence time Ct[s] for the different models and estimation methods. The convergence time gives a quantitative result of the accuracy of the sub-model. Carolina Passenberg [12] proposed that the converge time is the instance of time when the parameter error $e = p - \hat{p}$ remain within a 5% bound, where p is the true parameter and $\hat{p}$ are the estimated parameters. However, this metric is only available when the true parameter p is known which in our test is not available. As an alternative we observe the time when the REE metric applied in the estimation process of both system goes below the 1% for each sub-model. In this proposition Ct is time that has passed between the first time that the objects enters in a sub-model zone and the time that the estimation process needs to satisfy the REE standard, $t_{(REE<1\%)}$. The time when the object enters in a sub-model zone is equal to the time that the remote transition detector captures the type of dynamics $t_{detection}$. This method works for both reactive and proactive systems on the assumption that the reactive system estimation process, particularly within friction models, cannot provide accurate estimations before coming into touch with the environment.

$$C_t = t_{(REE<1\%)} - t_{detection} \tag{6.1}$$

For this experiment, the parameters of the REE and the estimation processes, N and λ are selected to based on the analysis of Ahmad et. al. [2] that displays the tradeoff between the stability and the tracking ability based on the number of samples and the forgetting factor λ. The selected parameter are a window equal to $N = 100$ samples of the RLS estimator, executed to 1KHz; and forgetting factor equal to $\lambda = 0.999$ to guarantee accurate results in the estimations.

6.4.2 Evaluations tools: Interactive process

In this part of the experiments we evaluate the modelling accuracy of the three different sub-models. The main measurements is to calculate the error between remote and virtual force in each sub-model. To analyse modelling accuracy we use two measurements, the Normalized root mean square error (NRMSEf [%]), e.q 6.2 based on Model Evaluation Tools in [12].

NRMSE is applied to understand the error between the remote and virtual forces taking into account the scale of magnitude of the remote forces. This measurement is mainly used to check the competence of the submodel.The NRMSE results are compared with the Just Noticeable Difference(JND) of the forces [15]. The evaluation is positive if the results are below the JND for linear force in the arm 10% [15][12]. To analyse the hypothetical advantage of the proactive over the reactive MMT system in an environment with changeable dynamics, we compare the metric NRMSE in each step that corresponds with an interactive process.

$$NRMSE_F = \frac{1}{F^s_{max} - F^s_{min}} \frac{\|F^s - F^m\|}{\sqrt{N}} \cdot 100\% \tag{6.2}$$

where F^s_{max}, F^s_{main} is the max and min force in the measured slave forces, F^s measured slave forces, F^m the virtual forces render in the master device and N the number of samples.

6.4.3 Hypothesis of the experimental results

Based on the two evaluation tools and the differences between the reactive and proactive system model switching techniques, we proposed two hypotheses regarding the results. The reactive system should have an advantage in the initial set of encountered sub-models based on the convergence time measure, Ct. The main reason for this is that if the reactive system design identify correctly the sub-model of the environment it does not need the previous step that the proactive system does. As all its sub-models estimators are running constantly, and the overall system is simpler than the proactive system it should provide a faster estimation in an experiment without time delays. The proactive system's major benefit may be demonstrated in teleoperation with multiple switches between known sub-models, since it will use its memory system to reduce the total Ct of the experiment. From the NMRSE metric, both system should in theory provide similar results as the quality of the estimation is the same. The main difference might appear if the any of the system misidentify submodels which will decrease the perfomance of the NMRSE metric.

In relation to this study's research question, both designs should be capable of supporting an MMT system in an environment with changeable dynamics. However, in the case of an environment where interactions are similar and repetitive, the application of forces, kinematics and visual data to inform the model switch functionality will be the most effective between the two proposed designs.

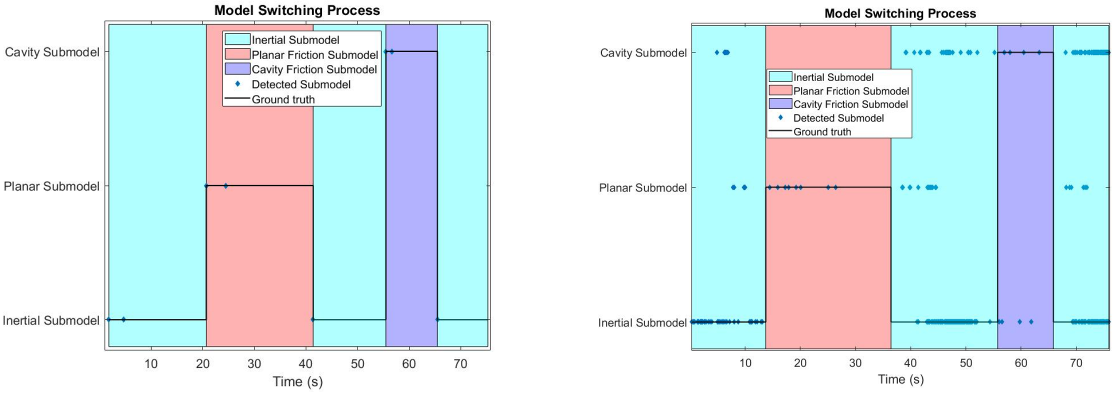
7 Experiments Results and Discussion

7.1 Introduction

The results of the experimental setup described in section 6 are presented in this chapter; which are presented into three sections. The first section demonstrates how both systems identify the multiple dynamics of the environment. Section 6.3 suggests a procedure for the experiments, leading to three types of dynamics being perceived during the experiment, one of which is recorded three times. As a result, the system must identify five zones where the dynamics of the environment change .The second part explores the response time of the system to provide an accurate model switch for each dynamic zone. The last part looks at each system's modelling accuracy of each submodel.

7.2 Model switching detection

The identification of the multiple dynamics and its changes for both system during the experiments are shown in figure 7.1. The Y-axis displays the results of the model switch, there are three possible results: Inertial, Planar or Cavity submodel. The X-axis divides the duration of the experiment into the five dynamics zones described in 6.3. The proactive system identification, figure 7.1a, follows correctly the ground truth of the experiment. The proactive system only gives one or two model switch per dynamic zone. It model switches twice when is the first time it detects a type of dynamic, one is the detection mode and the second is the update zone, section 5.3.5. When the system recognize a known dynamic the proactive system only model switches once, equivalent to the recall model switch mode. The reactive system identification, figure 7.1b, is not able to follow continuously the dynamics changes, specially in the inertial zones, when there is a constant mismatch between the inertial and cavity friction submodels.



(a) Proactive System

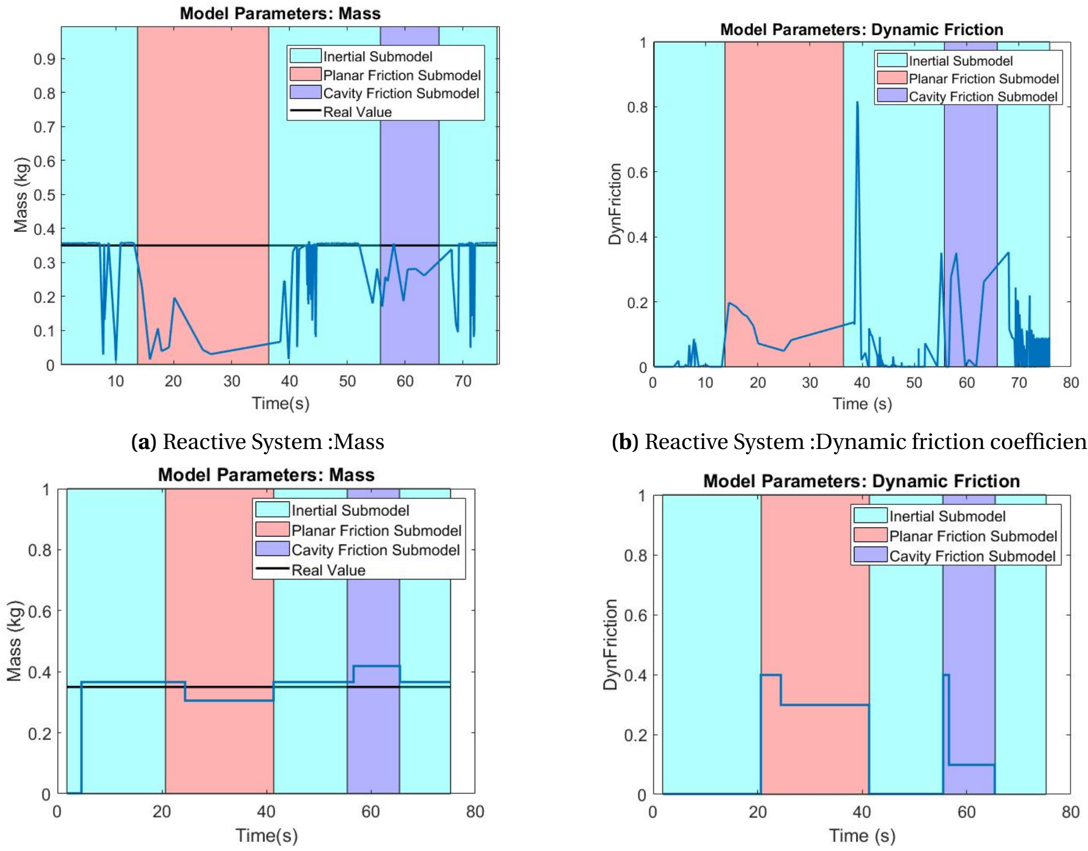
(b) Reactive System

Figure 7.1: Model switching function in the overall experiment

To explore in detail the consequences of the mismatches in the dynamics zones, figure 7.2 displays the physical parameters that are estimated and send to the local model during the experiments in both systems. Figures 7.2a and 7.2b show the effect of the misidentifications in the parameters of the reactive system. The cavity friction submodel is often selected in the inertial zones rather than than inertial submodel. Its effect can be seen in the the dynamic friction coefficient, μ, that instead of being zero is estimated with a low value, it seems that the estimation process in the cavity friction submodel understand the inertial zones, the free movement space, as a cavity whose dynamic friction coefficient is close to zero. Given the dynamic equations of the inertial and cavity friction, eq. 4.3 and eq. 4.5 and μ close to zero it is plausible this misdentification.

In the second inertial zone, the reactive system selects the cavity friction submodel with a high dynamic friction coefficient and a low mass value over the inertial submodel. However, this situation does not have a feasible explanation and, it indicates a clear failure in the identification explained further in section 7.3.

Figures 7.2d and 7.2c shows the parameters in the proactive system. Given the design of the proactive system, the system switches once or twice in each dynamic zone. When the system encounters for the first time a submodel the system parameters are changed twice. The first parameter value is the initial estimation of the submodel that it sends when the system switches to the detection switch mode. Figure 7.2d shows this initial estimation at the beginning of the planar and cavity friction zones that eventually is updated to the estimated value incoming from the estimation process. This update of the value that it shows as a step in the graph is the action of the model switch update mode that provides the estimated parameters from the estimation process. When the experiment reaches a known submodel, such as the second and third inertial submodel zones, the system parameters shift to the saved values from the prior inertial submodel estimation. Figure 7.2c shows that the parameter mass does not have a good estimation in the planar and cavity friction zones, however, the estimation remains more stable than in the Reactive system.



(a) Reactive System :Mass

(b) Reactive System :Dynamic friction coefficient

(c) Proactive System: Mass

(d) Proactive System :Dynamic friction coefficient

Figure 7.2: Parameter estimation in the experiment, the ground truth in the dynamic friction coefficient is unknown in the friction zones of the experiment

7.3 Model switching time response C_t

This section displays the results from the evaluation tool of section 6.4.1. Table 7.1 shows the response time of the system to provide an accurate estimation in each dynamic zone of the experiment.

Ct(s)	Inertial	Planar Friction	Inertial	Cavity Friction	Inertial
Reactive System	**0.43**	**0.75**	**4.72**	**1.26**	**3.42**
Proactive System	**1.06**	**3.72**	**0.0**	**1.15**	**0.0**

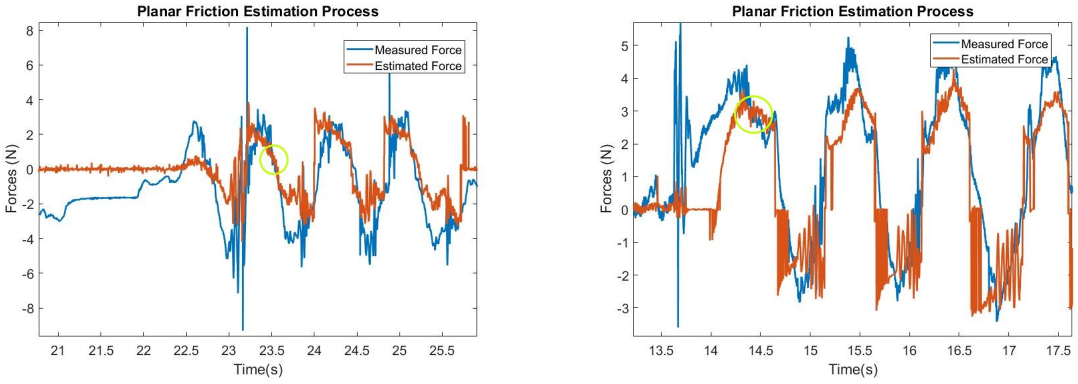
Table 7.1: Results of the Ct in the five dynamics zones of the experiment

As expected, under no time delays, the reactive system is faster than the proactive when they encounter for the first time the inertial and planar friction submodels. There is a large difference in the the first inertial and the planar friction zones. Once the systems re-enter the inertial zones, the proactive system recognizes the dynamics, and the response time is immediate. Contrary to the proactive system, the reactive system greatly increases its convergence time in the second and third inertial zones. In the first inertial zone, the times differences are given the increased complexity of the proactive system. While the reactive system starts estimating at the beginning of the experiment the proactive system activates the inertial submodel estimator only when the system detects the type of submodel.

To examine the disparities in the convergence time Ct in the planar friction zone between the proactive and reactive systems in further detail, we look into the data utilised to evaluate the REE estimate measure at the time, as mentioned in section 5.2.4. Table 7.2 displays the information used to satisfy the REE evaluation in the planar friction zone. The results of both systems satisfy the equation, however, the reactive system has a relative error three times bigger than the proactive due to a higher maximum error that was found outside of its ideal dynamic zone. As a consequence the estimation given by the reactive system is faster but less accurate, figure 7.3 shows the difference of tracking when the convergence time is computed.

Planar Friction Estimations	Ct(s)	Relative Error (N)	Max Error(N)	Max Error submodel location
Reactive System	0.75	0.22	-25.7	Inertial
Proactive System	3.72	0.075	7.52	Planar Friction

Table 7.2: Information used to estimate the planar friction submodel through the REE metric



(a) Proactive System

(b) Reactive System

Figure 7.3: Estimated forces vs measured forces of the planar friction estimator in the planar friction submodel zone. The green circle indicates when the REE metric is satisfied

To analyse the results in the second and third inertial zones of the experiment, figure 7.4 shows the mass estimation that the inertial estimator produces in the reactive system. As the inertial submodel dynamics only consider the mass parameter, its estimation is heavily degraded after being exposed to the measured forces in the friction zones of the experiment. The estimator varies the mass parameter in an attempt to match the friction dominant forces whose results is shown in figure 7.4. As a result, the inertial estimator requires additional time to generate acceptable estimates, as it did in the first inertial submodel zone.

At the end of section 7.2 it was observed that the system selected the cavity friction submodel during the second inertial zone with high values in the dynamic friction coefficient μ over the inertial submodel. The cause of that selection is the combination of the observations in the planar friction C_t and in the second and third inertial C_t. First that the REE metric does not provide a good estimation; and that the inertial submodel estimation is degraded in the friction models zones.

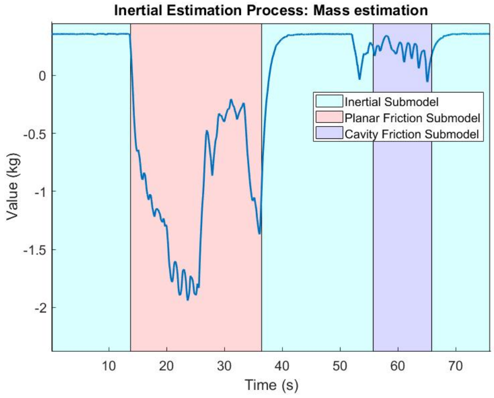


Figure 7.4: Mass parameter estimation of the inertial submodel in the reactive system

7.4 Modelling accuracy NMRSE

In this section, the accuracy of each submodel is evaluated using the NMRSE metric. To analise each submodel the three first occurrences zones of the experiments, figure 7.1 are selected, given that the first inertial submodel has the least mismatches in the reactive system. Table 7.3 groups the NMRSE results for each submodel in both systems. It includes an additional result, inertial unbiased, that shows the inertial results after the experiment where a DC component is eliminated, see section 7.4.1. The NMSRE results show that the only submodel that satisfy the JND threshold, under 10% , is the inertial submodel in the proactive system and only after being corrected. Each submodel is analysed to determine the importance of the NRMSE results in both systems.

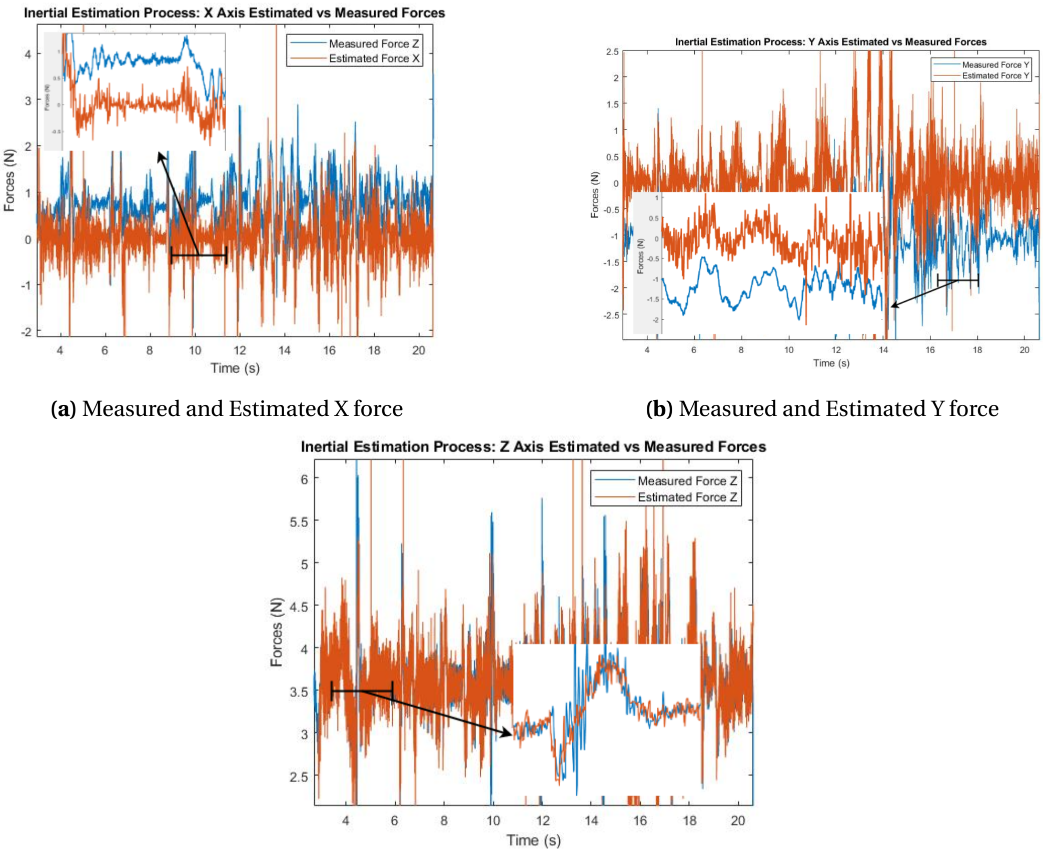
	Proactive System				**Reactive System**			
NRMSE(%)	**Inertial**	**Inertial unbiased**	**Planar Friction**	**Cavity Friction**	**Inertial**	**Inertial unbiased**	**Planar Friction**	**Cavity Friction**
Axis X	**16.46**	**10.11**	**16.34**	**-**	**13.45**	**13.45**	**10.27**	**-**
Axis Y	**20.01**	**9.37**	**11.62**	**-**	**15.53**	**11.80**	**25.84**	**-**
Axis Z	**4.23**	**4.23**	**-**	**11.08**	**10.06**	**10.06**	**-**	**13.31**

Table 7.3: NRMSE results for the submodels in each system

7.4.1 Inertial Submodel

To analyse the inertial submodel accuracy, the proactive system results are shown in figure 7.5 as they seems to be the better estimation. The Z direction tracks the measured forces, its error in centered in zero and with variance of ±0.3544N, figure 7.5c. The X and Y shows similar behaviour, however there is a DC component of the force that the inertial estimator does not seems to process. The cause of this DC component is a small variation in the orientation of the end-effector. At the beginning of the experiment, the ATI Mini40 sensor applied a bias function to eliminate the initial measurements before the object is grabbed, see appendix A.2. If the end-effector orientation changes, the initial measurements, such as the weight of the gripper, emerge again during the experiments.

In this case, the DC component is eliminated after the experiment to understand the effect of not including this effect in the estimation. As shown in table 7.3, removing the bias improves the performance of the inertial submodel. The results of the inertial system in the reactive system show worse performance than the proactive system. Its main cause is that the cavity friction is selected multiple times over the inertial submodel, as shown in figure 7.1b.



(a) Measured and Estimated X force

(b) Measured and Estimated Y force

(c) Measured and Estimated Z force

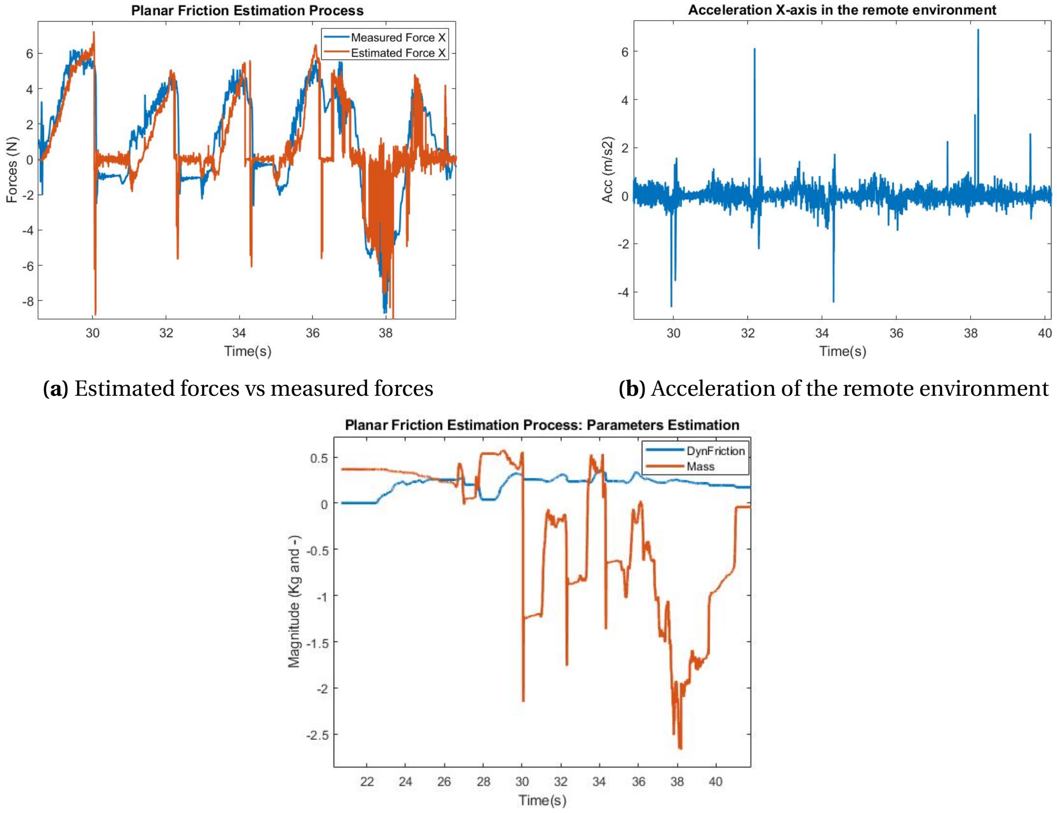
Figure 7.5: Estimated forces in the inertial submodel

7.4.2 Planar Friction Submodel

To evaluate the planar friction submodel, the X-axis of the proactive system is selected, whose NMRSE result is 16.34%. Figure 7.6 shows the estimation forces, its computed parameters and the measured acceleration during this time of the experiment. Figure 7.6a shows that the estimated forces cannot follow the force increase in the remote forces, specially when the static friction is dominant at the beginning of the forces.

During this phase of the experiment, the object is moved four times in the positive X direction and one backward at the end, which is represented in the five increments of forces. Every time the object is moved, it must stop as the workspace in the master device reaches its limits. Once the object stops, the object decelerates, generating large accelerations, figure 7.6b. When these accelerations are observed, the remote forces shift from their dynamic friction phase to the static friction. As the planar friction submodel does not estimated the static friction coefficient, the RLS algorithm tries to match the static friction force using the mass parameter. Figure 7.6c shows this circumstance. The combination of both situations is the leading cause of the unstable estimation of the mass parameter. As a result, the mass is incorrectly calculated, affecting the force estimates in the final movement of this phase of the experiment. The dynamic friction coefficient parameter is stable at around 0.23 unit-less.

In the dynamic friction phase, the estimator follows the forces, but it is not able to match the real values, which can be caused due to the estimation of the mass parameter. When the object is not moving, observed in the transition between the force increments, the DC component observed in the inertial submodel is also measured in the remote forces.



(a) Estimated forces vs measured forces

(b) Acceleration of the remote environment

(c) Estimated parameters

Figure 7.6: Planar friction estimation in the X-axis of the proactive system

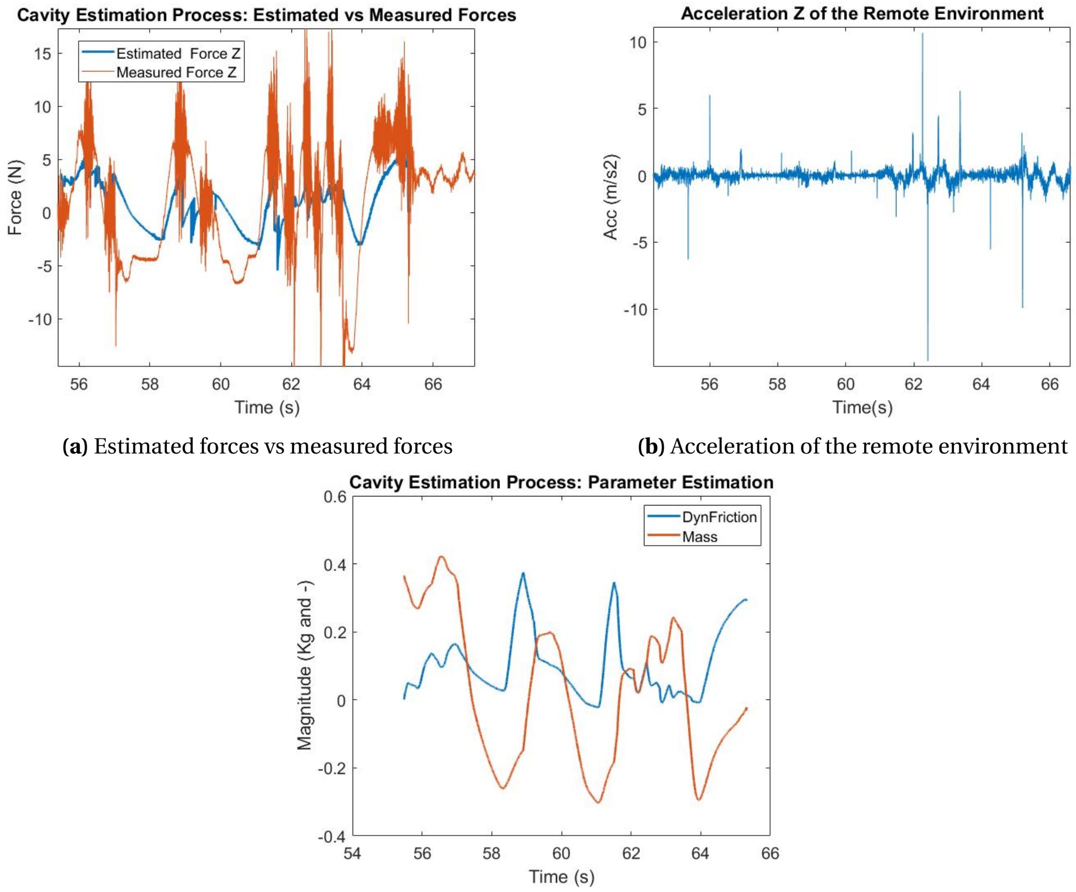
7.4.3 Cavity Friction Submodel

The NMRSE results of this submodel are over the JND threshold in both systems. Figure 7.7 shows the estimation process of the cavity friction submodel in the proactive system. In this phase of the experiment the object is introduced in the cavity, while is in contact with its sides to observe the friction between both elements. The dynamic friction phase of the model is observed in the oscilatory part of the graph 7.7a. The object also touches the bottom of the cavity three times, that is observed in figure 7.7a when the forces are relative constant. When the object touches the end of the cavity, the force value fluctuates depending on the force used to push the object by the Franka Emika robot, such as the case at the time 58s,61s and 64s.

Figure 7.7c shows that the estimator is unable to offer consistent parameters for both the mass and the dynamic friction coefficient; its main cause, as in the planar friction submodel, is the different phases of a friction model. Firstly, when the object is pushed against the bottom, the mass parameter becomes negative. During this phase of the experiment, the motion in the acceleration are velocities are minimal, figure 7.7b, therefore the RLS estimator cannot estimate the dynamic friction coefficient, whose estimation goes to zero. In the case of the mass parameter, the RLS estimator keeps estimating as the gravity term is not negligible, which results in the negative estimation of the mass in an attempt by the RLS estimator to match the forces of the contact.

On the other hand, during the phase of the experiment where the object is moved against the wall and the friction component is dominant, the dynamic friction coefficient is estimated while the mass parameter does not seem to be affected as much as the dynamic friction coefficient. This might indicates that the mass parameter may not have a significant impact during the friction phase, however during this time of the experiment the mass estimation have been already degraded by the previous contact with the bottom of the cavity and the estimated forces cannot match the measured forces, figure 7.7a.

It is also observable that the estimator is unable to track the system's static friction component, which impedes its ability to match the measured forces. However, compared with the planar friction submodel, the mass parameter does not seem to be affected by high values in the accelerations, as seen in figure 7.7a, this might be due to the presence of the gravity term in the estimation process. The peaks in the remote forces measure appear to be a vibration caused by the telemanipulation when it is pushed against the cavity X and Y axis while moving in the Z direction.



(a) Estimated forces vs measured forces

(b) Acceleration of the remote environment

(c) Estimated parameters

Figure 7.7: Cavity friction estimation of the proactive system

8 Conclusion and Future Work

8.1 Conclusion

The primary purpose formulated in this research is to explore the design of an MMT in an environment with changeable dynamics. For that pursuit, an MMT system needs to develop the ability to model switch the local rendered forces and adapt to the changeable dynamics of the environment. Based on the available information from the environment, two model switching strategies are formulated and implemented in the MMT architecture. The first method the estimation error given by the estimators of each sub-model, named reactive system. The second method, named the proactive system, utilises forces, visual data, and kinematics of the environment to detect the multiple dynamics of the environment.

The experiment findings demonstrated the ability of model switching strategies used by reactive and proactive systems in an environment with multiple dynamics. Section 7.4 shows the differences between both system in detecting three types of dynamics that switch five times during the experiment. The proactive system strategy limits the amount of model switch per zone, whose results matches the dynamics of the environment. In the case of the reactive system its model switching strategy provides multiple switches, based on the estimation error, that do not always match the dynamics of the environment, specially when the inertial dynamics are dominant. The cause of the mismatches is investigated in section 7.3.

The reactive system uses the REE metric to model switch and to estimate the parameters of the environment, the results shows that the ERR standard does not provide a robust solution for neither of both functions, as indicated in table 7.2. The main pitfall of REE is in the $e_{k,max}$ can be recorded outside of the agreeable dynamics of that specific estimator, therefore the REE does not assure a acceptable quality for the estimations. An alternative is to use an flat threshold to measure the relative error, however, given the accuracy shown in section 7.4 a threshold with that only allows minimal estimation error wont be able to detect the friction model. This situation can be fixed by improving the friction models.

Section 7.3 also demonstrates the disadvantage of a parallel estimation process that the reactive system used as the estimation error based strategy needs all estimators to work at the same time to discern the dynamics. Even in the case of the inertial estimator, whose estimation is the best of the three dynamics, the estimation results are degraded when subjected to dynamics that do not suit its model. Figure 7.4 and table 7.1 shows that once the dynamics of the experiment switch to the favorable dynamics of the inertial estimator, its response is slow down several times compared with the first inertial zone.

The main reason why the reactive system shows poor performance at model switching is the inability of an RLS estimator to discern between adequate or inadequate data without additional information from the environment. If the reactive system had been aware of the type of dynamics, the RLS estimation wont have a decrement in its response time. An alternative to improve the tracking response is to lower the forgetting factor λ, or to use an Adaptive Forgetting Factor in the RLS estimator, RLS-A [18]. For both cases the trade-off is a decrement in estimation accuracy that given the results in table 7.3 wont be beneficial for an identification based on the estimation error. The results indicate that using the estimation error to inform model switching is ineffective. The system needs more information to discern the multiple dynamics of the environment because of the effect experienced in the RLS estimator.

The strategy included in the proactive system seems to satisfy the requirement to discern multiple dynamics, even though the proposed submodels do not seem to model the experiment's environment properly. Figure 7.1a shows that the system correctly detects the dynamics of the environment, and table 7.1 shows that the system can take advantage of its memory system to improve its model switching response time. The proactive system results provide an acceptable model switch function because its possible transition are fixed to the conditions designed in section 4.6, however, if the system enters in contact with any element that is not considered in these conditions the proactive system wont be able to detect it. The proactive system and its model switching strategy seems to point in the right direction, and the application of forces, visual data, and kinematics of the environment are an adequate tool to inform model switching in the selected experiment.

In conclusion, the results from the experimental setup seems to indicate that the proactive system, using forces, visual data, and kinematics of the environment to reconstruct and understand the environment is a possible solution to inform model switching and therefore provide an MMT system design for an environment with multiple dynamics.

Finally the answer for the main research question is:

How can a MMT system be designed for an environment with changeable dynamics?

The results show that only the proactive MMT system is able to work in an environment with multiple dynamics. Its model switching functionality delivers stable detections of the sub-models that form the changeable dynamics of the environment, however, the estimations are not accurate as the modelling decision of the sub-model cannot fit all the features of the dynamics experienced in the experiment. The proactive mmt system works as it detects the multiple dynamics of the system using forces, visual data, and kinematics of the environment.Its model-switching feature is independent of the estimation process, and it incorporates a memory system that facilitates in detecting when the manipulation navigates to known dynamics that the system has retained in memory..

To complete the research question in depth we derive one sub-level question to address the type of information used in the model switching function of the MMT system.

What type of information available in the environment is effective to inform model switching?

The results shows that the combination of forces, visual data, and kinematics of the environment are an effective method to inform model switching. However, this comes with the cost of a complex system and the requirement to analyse how each type of dynamic can be detected based on this type of information and how the multiple dynamics are interconnected between them.

8.2 Future Work

The most important feature to develop in future work is reconstructing the remote environment using visual data. In this study, the final implementation of the reconstruction of the remote environment did not utilize this feature.As a result, measuring the measurement error associated with reconstructing the environment using visual data was unachievable. To validate the value of this design, an examination of the certainty of the visual measurement is required. It can also be used to remove the remote force detector in order to verify the match between virtual and remote environments.In this case, the model switching functionality could be completely virtual, making the model switching functionality immune to time delays.

Secondly, some improvements to the modelling of the environment should be made. Including the stiction friction in the friction models would improve the tracking of the remote environment. This means increasing the complexity of the system as it was implement in Ni et. all. [9].

A Data Acquisition

This appendix describes the techniques applied in this study to obtain reliable data during the experiment.

A.1 Acceleration estimation

The Franka Emika robotic arm and the haptic device omega 6 do not provide a measurement of their acceleration. To estimate its acceleration for both side of the teleoperation it is applied a Low pass filter to reduce the signal noise of the velocity, the cut off frequency used was 1Hz and delta time $\Delta t = 0.03$ to cycle the LPF. The estimation process for the acceleration is a time derivative whose Δt is equal to the frequency of the ROS nodes, 1000Hz.

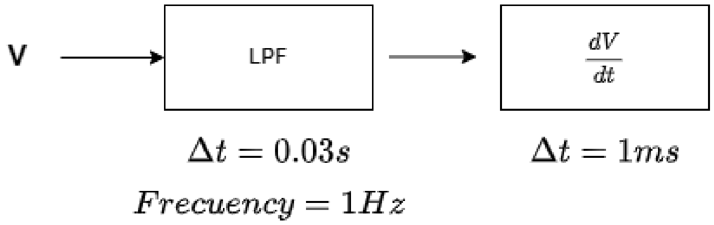


Figure A.1: Estimation process for the acceleration of both Franka Emika robot and Omega 6

A.2 Analysis of the initial remote forces

The ATI FT Sensor Mini 40 is used to measure the forces of the environment. Because it is an external sensor from the Franka Emika robot, the sensor measures some initial forces which are the forces applied by the robot to hold the sensor and the franka gripper in position. The mentioned initial forces are displayed in table A.1. The ATI FT sensor provides a bias function to remove these initial measurements and provides an unbiased measurement for the beginning of the experiment.The ATI FT bias function removes static forces, although some of these initial forces, such as the weight of the gripper located beneath the FT sensor, will emerge during free space movement. Additionally, if the orientation of the end-effector is not fixed, then some of the gravitational forces will be shifted from their original Z direction.

Sensed Forces Pre-bias (N)		
X	Y	Z
-0.72	-0.34	32.55

Table A.1: Initial forces sensed by the F-T sensor

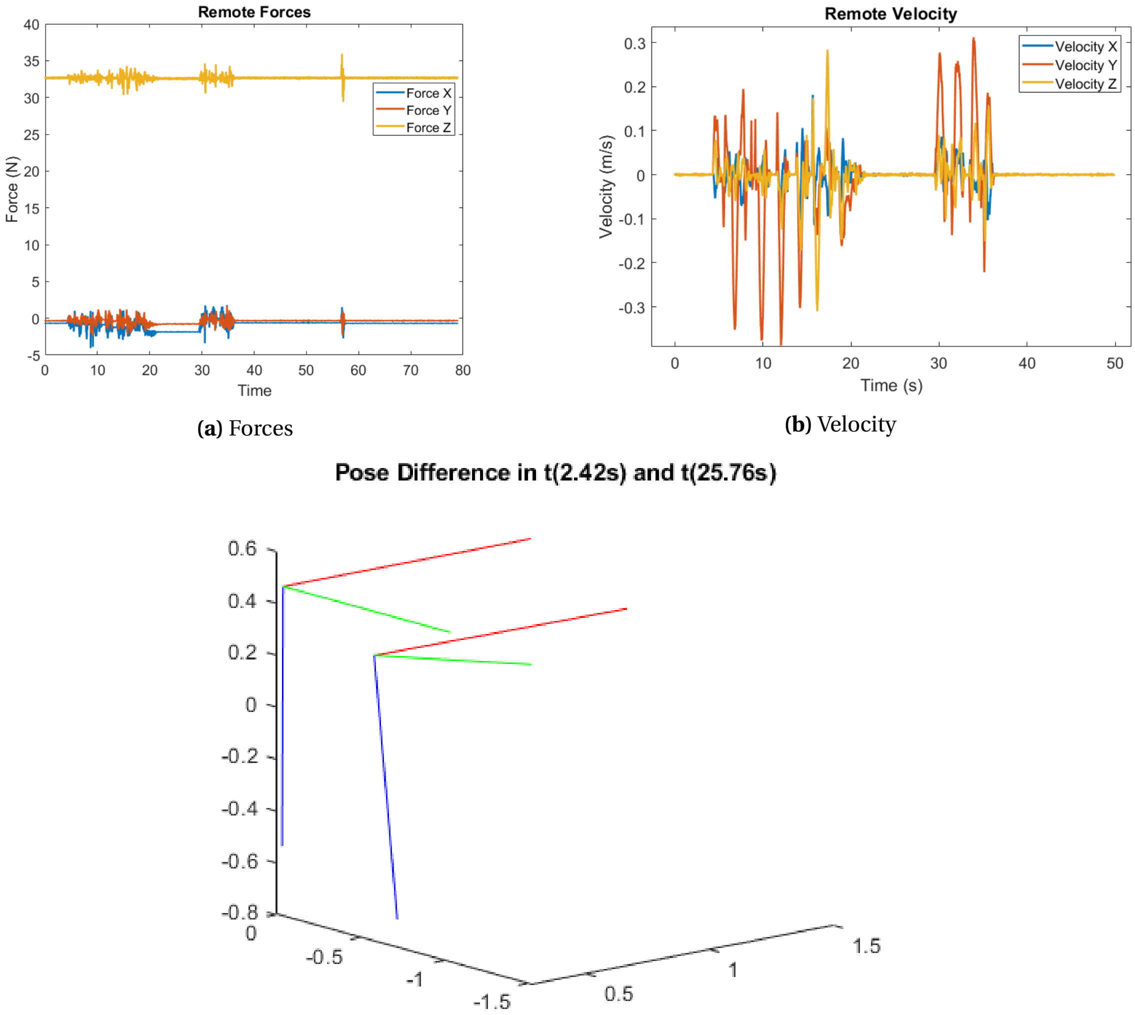
Figure A.2a illustrates a manipulation test without holding anything, it can be seen that the forces in the x and y axis changes over time. To discern if those changes are related to a dynamic move, figure A.2b displays the velocities of this experiment. The velocities figure explains the oscillatory phase of the forces but not the fixed shift in forces at 25.76s. This effect is caused by a rotation in the orientation of the end-effector, as seen in figure A.2c. To demonstrate if the change in forces is determined by rotation, the initial force F(0) are represented in the orientation given in t=25.76:

$$F^{*}(25.76) = R_0^{ee}(t = 25.76) \cdot (R_0^{ee}(t = 0))^{-1} \cdot F(0)$$

where $R_0^{ee}(t=0)$ and $R_0^{ee}(t=25.76)$ are the rotation matrix of the end-effector w.r.t the world frame at time t=0 and t=25.76 respectively. The results of comparing the transformation and the measured forces at 25.76 are shown in table A.2.The minimal difference of force can be assumed to be part of the noise present in the force measurements. This results indicates that even though the bias eliminates the initial forces they can reappear if the orientation of the end-effector changes during the manipulation. To include this influence in the real measurements of forces, it is required to correct this rotation at each time; unfortunately, the measurements from the rotation are not available in every instant, and their measurements include noise which makes them not accurate enough.

	Forces Pre-bias at t = 25.76s (N)		
	X	Y	Z
F	-1.82	-0.76	32.59
F^*	-1.82	-0.76	32.50
$F-F^*$	0.0052	0.0021	-0.0924

Table A.2: Results of transformation in the forces



(a) Forces

(b) Velocity

(c) Pose of the remote environment during time 0 and time 25.76

Figure A.2: Plots of the remote environment before applying the sensor bias function

B RLS Estimation: Mathematical Derivation

This appendix describes the parameter estimation algorithm applied in this study to obtain the parameters of three selected submodels for the linear forces: inertial, planar friction and cavity friction.

B.1 Inertial submodel estimation

Given the equation of the inertial submodel

$$\vec{f} = m\left(\vec{a} - \vec{g}\right) \tag{B.1}$$

In the experimental setup, the orientation of the gripper is fixed and coincides with the axis of the fixed world frame, the RLS algorithm is applied for each axis:

$$y = \phi^T \boldsymbol{\theta}$$

where:

$$y = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}, \quad \phi_{Axis} = \begin{bmatrix} a_{Axis} - g_{Axis} \end{bmatrix}, \quad \theta_{Axis} = \begin{bmatrix} m \end{bmatrix}$$

Before the estimation process begins the initial forces sensed by the F-T sensor are removed using its bias function. However, these initial forces, such as the weight of the gripper located beneath the FT sensor, will emerge during free space movement. To compensate this effect, the estimation process used the mass of the gripper $m_{gripper} = 0.7kg$ in the RLS process to ensure that the estimation error is correct and to divide the estimation of the $m_{gripper}$ and the mass of the object, m.

$$\epsilon = y_{measured} - y_{estimated}$$

$$\epsilon_{axis} = F_{axis} - (y_{estimated_{axis}} + m_{gripper} \cdot a_{axis})$$

B.2 Planar friction submodel estimation

Given the equation of the planar friction submodel:

$$\vec{\mathbf{f}} = m \cdot \vec{\mathbf{a}} + c \cdot \text{sign}(\vec{\mathbf{v}}) + \vec{\text{d}}, \quad c = \mu \cdot m \cdot g \quad \vec{d} = m \cdot \vec{g} \tag{B.2}$$

In the experimental setup, the orientation of the gripper is fixed and coincides with the axis of the fixed world frame, the RLS algorithm is applied for the X and Y axis, so the component d is not applied in this scenario. To calculate the dynamic friction coefficient, μ the normal force, $m \cdot g$ is substitute by the measured force F_z as the normal force applied to the surface:

$$y = \begin{bmatrix} F_x & F_y \end{bmatrix}, \quad \phi_x = \begin{bmatrix} a_x & sign(\mathbf{v_x})F_z \end{bmatrix}, \quad \theta_x = \begin{bmatrix} m \\ \mu \end{bmatrix}$$

$$\phi_y = \begin{bmatrix} a_y & sign(\mathbf{v_y})F_z \end{bmatrix}, \quad \theta_x = \begin{bmatrix} m \\ \mu \end{bmatrix}$$

B.3 Cavity friction submodel estimation

For the cavity friction submodel, the estimation is applied in the Z axis of the fixed frame world that coincides with the Z axis of the end-effector of the Franka Emika robot. From its dynamic equation we derive the regression matrix ϕ and the set of parameters θ:

$$\mathbf{F_z} = m \cdot a_z + \mu \cdot \text{sign}(\mathbf{v_z}) \cdot \|Fx + Fy\|_2 - m \cdot g \tag{B.3}$$

$$y = \begin{bmatrix} F_z \end{bmatrix}, \quad \phi_z = \begin{bmatrix} a_z - g & sign(\mathbf{v_z})\sqrt{F_x^2 + F_y^2} \end{bmatrix}, \quad \theta_x = \begin{bmatrix} m \\ \mu \end{bmatrix}$$

Bibliography

[1] Achhammer, A., C. Weber, A. Peer and M. Buss (2010), Improvement of model-mediated teleoperation using a new hybrid environment estimation technique, in *2010 IEEE International Conference on Robotics and Automation*, IEEE, pp. 5358–5363.

[2] Ahmad, M. S., O. Kukrer and A. Hocanin (2011), The effect of the forgetting factor on the RI adaptive algorithm in system identification, in *ISSCS 2011-International Symposium on Signals, Circuits and Systems*, IEEE, pp. 1–4.

[3] Catoire, M., B. N. Krom and J. B. van Erp (2018), Towards a test battery to benchmark dexterous performance in teleoperated systems, in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, Springer, pp. 440–451.

[4] Dinger, G. (2015), Dynamic modeling and simulation of the screwing behavior of thread forming screws, *Journal of Manufacturing Processes*, **vol. 20**, pp. 374–379, ISSN 1526-6125, doi:https://doi.org/10.1016/j.jmapro.2015.06.012.
https://www.sciencedirect.com/science/article/pii/S1526612515000572

[5] Franken, M., S. Stramigioli, S. Misra, C. Secchi and A. Macchelli (2011), Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparency, **vol. 27**, no.4, pp. 741–756, doi:10.1109/TRO.2011.2142430.

[6] Hannaford, B. and J.-H. Ryu (2002), Time-domain passivity control of haptic interfaces, **vol. 18**, no.1, pp. 1–10, doi:10.1109/70.988969.

[7] Lawrence, D. (1993), Stability and transparency in bilateral teleoperation, **vol. 9**, no.5, pp. 624–637, doi:10.1109/70.258054.

[8] Mitra, P. and G. Niemeyer (2008), Model-mediated Telemanipulation, **vol. 27**, no.2, pp. 253–262, doi:10.1177/0278364907084590.

[9] Ni, D., A. Song, S. Wang, H. Li and C. Zhu (2018), Translational objects dynamic modeling and correction for point cloud augmented virtual reality–based teleoperation, **vol. 10**, no.1, p. 1687814017753870.

[10] Niemeyer, G. and J.-J. Slotine (1991), Stable adaptive teleoperation, **vol. 16**, no.1, pp. 152–162, doi:10.1109/48.64895.

[11] Olsson, H., K. J. Åström, C. C. De Wit, M. Gäfvert and P. Lischinsky (1998), Friction models and friction compensation, **vol. 4**, no.3, pp. 176–195.

[12] Passenberg, C. (2013), *Transparency-and Performance-Oriented Control of Haptic Teleoperation Systems*, Ph.D. thesis, Technische Universität München.

[13] Robotics, O. (2016), Robotic Operating System.
https://www.ros.org

[14] Smith, A. C. and K. Hashtrudi-Zaad (2005), Adaptive teleoperation using neural network-based predictive control, in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, IEEE, pp. 1269–1274.

[15] Stanney, K. M., K. S. Kingdon, D. Graeber and R. S. Kennedy (2002), Human performance in immersive virtual environments: Effects of exposure duration, user control, and scene complexity, **vol. 15**, no.4, pp. 339–366.

[16] Tzafestas, C., S. Velanas and G. Fakiridis (2008), Adaptive impedance control in haptic teleoperation to improve transparency under time-delay, in *2008 IEEE International Conference on Robotics and Automation*, pp. 212–219, doi:10.1109/ROBOT.2008.4543211.

[17] Verscheure, D., J. Swevers, H. Bruyninckx and J. De Schutter (2008), On-line identification of contact dynamics in the presence of geometric uncertainties, in *2008 IEEE*

International Conference on Robotics and Automation, IEEE, pp. 851–856.

[18] Wang, Z., A. Peer and M. Buss (2009), Fast online impedance estimation for robot control, in *2009 IEEE International Conference on Mechatronics*, pp. 1–6, doi:10.1109/ICMECH.2009.4957217.

[19] Willaert, B., J. Bohg, H. Van Brussel and G. Niemeyer (2012), Towards multi-DOF model mediated teleoperation: Using vision to augment feedback, in *2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings*, IEEE, pp. 25–31.

[20] Willaert, B., H. V. Brussel and G. Niemeyer (2012), Stability of model-mediated teleoperation: Discussion and experiments, in *International conference on human haptic sensing and touch enabled computer applications*, Springer, pp. 625–636.

[21] Xu, X., S. Chen and E. Steinbach (2015), Model-mediated teleoperation for movable objects: Dynamics modeling and packet rate reduction, in *2015 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, IEEE, pp. 1–6.

[22] Xu, X., B. Cizmeci, A. Al-Nuaimi and E. Steinbach (2014), Point Cloud-Based Model-Mediated Teleoperation With Dynamic and Perception-Based Model Updating, **vol. 63**, no.11, pp. 2558–2569, doi:10.1109/TIM.2014.2323139.

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

[24] Xu, X., J. Kammerl, R. Chaudhari and E. Steinbach (2011), Hybrid signal-based and geometry-based prediction for haptic data reduction, in *2011 IEEE International Workshop on Haptic Audio Visual Environments and Games*, IEEE, pp. 68–73.

[25] Xu, X., A. Song, D. Ni, H. Li, P. Xiong and C. Zhu (2016), Visual-haptic aid teleoperation based on 3-D environment modeling and updating, **vol. 63**, no.10, pp. 6419–6428.

[26] Yokokohji, Y. and T. Yoshikawa (1994), Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment, **vol. 10**, no.5, pp. 605–620.

[27] Yoon, W.-K., T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda and T. Doi (2004), Model-based space robot teleoperation of ETS-VII manipulator, **vol. 20**, no.3, pp. 602–612, doi:10.1109/TRA.2004.824700.