

Intelligent Parametric Visual Thinking System (IPVTS) as a Paradigm For Control Strategies with Time-Delay in Robotics

D. W. Repperger¹, N. I. Tikhonov²

¹ Air Force Research Laboratory
Building 33, AFRL/HECP
Wright-Patterson AFB, Ohio 45433

² National Research Council/
Air Force Research Laboratory
Building 33, AFRL/HECP
Wright-Patterson AFB, Ohio 45433

REPORT DOCUMENTATION PAGE (Standard Form 298)

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE

August, 2003

3 REPORT TYPE AND DATES COVERED

Interim Report, August 2002 - August 2003

4. TITLE AND SUBTITLE

Intelligent Parametric Visual Thinking System (IPVTS) as a Paradigm
for Control Strategies with Time-Delay in Robotics

5. FUNDING NUMBERS

6. AUTHOR(S)

D. W. Repperger, N. I. Tikhonov*,

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory, Human Effectiveness Directorate

Crew System Interface Division

Air Force Material Command

Wright-Patterson AFB, OH 45433-7022

8. PERFORMING ORGANIZATION

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

Air Force Research Laboratory, Human Effectiveness Directorate

Crew System Interface Division

Air Force Material Command

Wright-Patterson AFB, OH 45433-7022

10. SPONSORING/MONITORING

AL/CL-TR-2003

11. SUPPLEMENTARY NOTES

* National Research Council

2001 Wisconsin Avenue, NW

Washington, DC 20007

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT

14. SUBJECT TERMS

Force-Reflecting Controller, Time Delay, Virtual Reality Systems, Robotics Stability, Bilateral Teleoperation, Wave Variable

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF ABSTRACT

UNLIMITED

PREFACE

The theoretical, practical methods and the computer programs dealing with coordinating and adapting the appropriate virtual reality visual display with the corresponding sensory feedback (force, tactile) in process simulation for PHANToM robotics with a time delay using image understanding technologies based on new technology - Intelligent Parametric Visual Thinking System (IPVTS) were developed.

This visual environment system is based on a computer program using MS C/Visual C++ with MFC, OpenGL high-level language for a graphic environment, and driver programs for PHANToM integrated with the IPVTS. This virtual environment allows us to solve the tasks related with animation of virtual objects, interactions of objects with reflected forces, control tasks for virtual objects in Real-Time, and simulation of time delay.

The system dynamics experiments were conducted to study the simulation of the time delay into the control teleoperation system and the force feedback response for the slave robot. This experiment was conducted using a PHANToM robotic simulator and virtual modeling based on the IPVTS with different time delay of 0 s, 0.5 s, 1.0s and 1.5s and more. The goal of these experiments is to compare the stability of the control system with the time delay and without the time delay as well as the forces being on or off.

These experiments were also conducted with force reflection and force feedback. Here the force feedback response was defined as having a feedback function (as bias distance into virtual model) where feedback has been represented by “virtual force” into the visual model. Use has been used of the alternative controllers such as PHANToM Haptic Robot interface, computer mouse, 3D visual control system, and computer control software. The goal of these experiments is to compare the stability of the control system

with the time delay and without the time delay using the different types of sensory information and when the cues work together with the intelligent m-dimensional parametric control. It allows the user to obtain the advantage of the intelligent m-dimensional parametric control based on IPVTS.

We experimented a great deal on combinations of parameters using the wave controllers based on computer modeling in Real-Time system for the PHANTOM master and slave in a virtual 3D environment with time delay and without delay for non-linear control laws. We have shown the successful means of stabilizing remotely controlled teleoperation systems when time delays may be present of up to 1 sec.

A methodology is proposed to design a human machine interface to enhance the sensitivity of the operator to an environment, which he is remotely located from. A procedure termed “stochastic resonance” provides the basis for this design approach. The interface parameters to be manipulated include the level of force feedback from a joystick controller as well as a proprioceptive signals derived from a motion chair device. Originating in physics, stochastic resonance, in principal, means that a small, but non-zero, amount of noise is inserted into the human-display system through various sensory modalities. It is demonstrated that the operator has enhanced sensitivity to the overall interaction through this exposure since performance improvement and situational awareness are both influenced by this design. An experiment was conducted and data obtained from nine subjects in a simple disturbance rejection tracking task. A theoretical investigation of why such a system works is presented both from a physics perspective as well as from examples that provide a mathematical basis. These analogies are introduced to add credence to the method presented here as well as to understand the underlying reasons why this method (which, at first, seems counterintuitive) has advantages in

certain applications.

TABLE OF CONTENTS

STATEMENT OF PROBLEM.....	9
BACKGROUND AND RELEVANCE TO PREVIOUS WORK	10
GENERAL METHODOLOGY AND PROCEDURES	12
EXPLANATION OF NEW AND UNUSUAL TECHNIQUES	17
EXPECTED RESULTS AND THEIR SIGNIFICANCE AND APPLICATION.....	17
IPVTS IS UNIVERSAL CONSTUCTOR FOR HIGH-LEVEL DESIGN OF COMPLEX SYSTEM	19
A Cell is an Entity That Has Position, Orientation and an Arbitrary Length State Vector.....	20
A Bias Distance (feedback) - $BD_i(V)$ as Tool for the Global Control of Heuristics	21
Hypergraph Composition Model IPPMS	22
... Physical Setup and System Equations for the PHANToM Mechanism.....	24
ALGORITHM OF BASIC TELEOPERATION.....	27
The Virtual Force Reflecting Algorithm is Based on a Physics Model and on the Torque, Force, Position, and Velocity Limits of PHANToM.....	29
Analysis of Input Control Laws for Nonlinear Real-Time Feedback Strategies	35
USING IPVTS FOR TELEOPERATION BASED ON TIME DELAY COMMUNICATIONS.....	50
Experimental Results for Analysis of Stability Using the Velocity and Position Characteristics with Time Delay.....	60
Experimental results for analysis of stability using different input control	

strategies with time delay.....	64
WAVE BASED TELEOPERATION.....	71
Experimental Validation of Increasing Stability Using Wave Variables	74
Experimental Results	76
Future work	78
MODELING OF DYNAMIC AND UNCERTAINTIES BASED ON TRANSFER FUNCTION.....	80
IMPEDANCE MATCHED TELEOPERATOR.....	82
SLAVE AND MASTER CONTACT WITH ENVIRONMENT.....	82
REFERENCES.....	83

STATEMENT OF THE PROBLEM

The objective of this research is to develop better theoretical and practical methods for using virtual reality, image understanding technologies, and conceptual parametric object-oriented design for control strategies in Robotics, algorithms and software that is based on coordinating and adapting the appropriate virtual reality visual display with the appropriate sensory feedback (force, tactile) for modeling high-level descriptions of the sensing of surfaces, shapes, and textures.

The Human Sensory Feedback (HSF) project at the AFRL headed by Dr. Repperger [Repperger et al., 1995, 1997, 2000], captures attention of computer scientists and engineers as a new methodology for building human-robot technology and for the support for many aspects of requirements analysis that simulates an expert in the performance of control tasks.

However the problems of control in Robotics based on human-like control behavior, with the appropriate sensory feedback for a integrated real-time system, are complex and many-sided. During this transformation process the HSF provides automated support for many aspects of virtual reality interfaces using Visual and Haptic rendering based on maintaining consistency between graphical views and algebraic notation of a design [Repperger et al., 1995, 1997, 2000].

Thus engineering methods and procedures that encompass both geometric and non-geometric objects require a level of human-like intelligence for understanding this paradigm, which current methods do not support. Most of the existing systems can execute direct human instructions yet cannot help him in abstract thinking. Considering this analysis, the research objectives of this proposal are:

- To develop theoretical and practical methods of coordinating and adapting the appropriate virtual reality visual display with the appropriate sensory feedback (force, tactile) in a simulation for robotics using image understanding technologies based on new technology - Intelligent Parametric Visual Thinking System (IPVTS).
- To transform a visual spatial model into a control process algorithm to achieve the required functional properties with optimal characteristics.
- It is necessary to create the IPVTS that has an intelligent human-like engineering methodology where design, geometry, analysis, intelligent control and optimization are combined into a single computational system to aid in engineering design using high-level visual prototyping.

BACKGROUND AND RELEVANCE TO PREVIOUS WORK

The need for investigations of alternative sensory feedback, such as with a haptic interface, is indicated by being extremely important for adaptive remote control of complex integrated systems. This is because the human hand is a powerful tool through which the human brain interacts with the world [Repperger 1997], [Flanagan 2002]. According to publications [Repperger et al., 1995, 1997, 2000] and other scientists [Flanagan 2002], it is necessary to note that fundamental research in development methods of haptic interfaces and “virtual force” methodology have significant importance in improving the operator’s ability to observe and control his environment.

The paper [Repperger 1995] addresses adapting the kinematical control for robots and concludes with a review and description of prior work with problems involving dynamics, control, design, the trajectory and speed planning utilizing control, and data flow based on the mathematical model using Denevit-Hartenberg transformation [Repperger 1995]. This paper shows how the seventeen coordinate systems based on

Denevit - Hartenberg transformation are used for the formal model of the components orientation. To improve these developments we define the focus for this proposal on intelligent human-robot interface models for generating the global formal graphical universal model based on sensory perception and visual thinking.

In the work [Repperger 1997], it describes how a haptic interface was correlated to the visual scene displayed to the pilots.

Early research in this area has demonstrated certain benefits of using Virtual Reality Systems. In the reference [Floyd 1999], this author has developed the system in which the PHANToM can interact with the 3D bitmapped virtual environments using the Cellular Automata CAM-8 Modeling. Methods for control in Robotics based on the time delay and the wave variables are described, also, to design the system with high effectiveness. This bitmapped simulation allows the extracted position data and return force data directly with the PHANToM robotic simulation. It allows the user to create a very universal method of modeling. This is main advantage of this system, but this system has low speed for complex systems and requires a very large computational resource. It should be noted that this system has a lower number of control parameters for the objects involving optimization.

However, it is more important to integrate the bitmapped environment and the discrete dynamical system with a high-level computational system based on a graphic implementation and a combination of heuristics [Tikhonov 2002].

Extending on these works we believe that there is a need for a fundamental work to automatically generate suitable control strategies for robots and position tracking involving complex shape surfaces and textures, such a model has a feature-based visual model in the query using formal graphic specifications and visual thinking.

To improve these developments we define the focus on this proposal based on intelligent human-robot interface models for generating the global formal graphical universal model based on sensory perception and the visual thinking.

Creating of integration mechanisms for IPVTS which will provide the integration including multi-sensory feedback tools, using mathematical modeling and programming languages (C++, Java, Visual Basic.), and other integration formats and databases.

GENERAL METHODOLOGY AND PROCEDURES

Extending the robotics model [Repperger 1995], it is required to develop a new discrete dynamical system (Fig. 1) as high-level computational systems with graphic implementation and a combination of heuristics:

$$\mathbf{G} = [\mathbf{V}, \mathbf{E}, \mathbf{T}(\mathbf{V}), \mathbf{C}(\mathbf{V}), \mathbf{X}_i(\mathbf{V}), \mathbf{Y}_i(\mathbf{V}), \mathbf{Z}_i(\mathbf{V}), \mathbf{A}_i(\mathbf{V}), \mathbf{HF}_i(\mathbf{V}), \mathbf{P}_i(\mathbf{V}), \mathbf{PF}_i(\mathbf{V}), \mathbf{BD}_i(\mathbf{V}), \mathbf{TO}_i(\mathbf{V}), \mathbf{W}_i(\mathbf{V}), \mathbf{PB}_i(\mathbf{V}), \mathbf{G}_i(\mathbf{V}), \mathbf{SB}_i(\mathbf{V})] \quad (1)$$

where a composite node's implementation is the graph \mathbf{G} , \mathbf{V} is the set of vertices of regular lattice associated with edges \mathbf{E} , minimum execution time $\mathbf{T}(\mathbf{V})$ and control constraints $\mathbf{C}(\mathbf{V})$ associated with \mathbf{V} , $\mathbf{X}_i(\mathbf{V}), \mathbf{Y}_i(\mathbf{V}), \mathbf{Z}_i(\mathbf{V})$ are coordinates of vertices of vectors associated with \mathbf{V} ; homogeneous transformation matrix $\mathbf{A}_i(\mathbf{V})$, heuristic functions and rules $\mathbf{HF}_i(\mathbf{V})$, parametric constraints $\mathbf{P}_i(\mathbf{V})$, parametric function $\mathbf{PF}_i(\mathbf{V})$, feedback function (as bias distance) $\mathbf{BD}_i(\mathbf{V})$, text $\mathbf{TO}_i(\mathbf{V})$; connection weighs, $\mathbf{W}_i(\mathbf{V})$, local feedback function $\mathbf{PB}_i(\mathbf{V})$, genetic code $\mathbf{G}_i(\mathbf{V})$, and optimization criterion $\mathbf{SB}_i(\mathbf{V})$.

For example, the relative position and orientation of frame $(\mathbf{XYZ})_i$ with respect to frame $(\mathbf{XYZ})_{i-1}$ (Fig. 1), can now be expressed by a homogeneous transformation matrix $\mathbf{A}_i(\mathbf{V})$ and complex composition matrix $\mathbf{A}_{0i}(\mathbf{V})$ (for representation of the vector $\vec{01}$) which describes the composition of the translation matrix \mathbf{T}_{0i} , the translation

inverse matrix T_{01-1} , the rotation matrices, $R_{X01}(\alpha)$, $R_{Y01}(\beta)$, $R_{Y01}(\varphi)$, $R_{Z01}(\theta)$, $R_{Z01}(\gamma)$ on angles respect to coordinate axes X_i, Y_i, Z_i [Foley , 1996], the inverse rotation matrices $R_{Z01-1}(\theta)$, $R_{Y01-1}(\varphi)$ and scaling transformation matrix S_{01} for arbitrary vector $\vec{O1}$ as follows:

$$A_i(\mathbf{V}) = A_{01}(\mathbf{V}) = T_{01-1} * R_{Z01-1}(\theta) * R_{Y01-1}(\varphi) * S_{01} * R_{X01}(\alpha) * R_{Y01}(\beta) * R_{Z01}(\gamma) * R_{Y01}(\varphi) * R_{Z01}(\theta) * T_{01} \quad (2)$$

The structure of IPVTS is a tree consisting of atomic and composite nodes (Fig. 1, Fig 2). The IPVTS automatically allows the transform of each formal parameter from graphic specification CAD systems into a primitive vector-prototype of fractal homogeneous neural network in according with the homogeneous composition the vector transformation matrix $A_m(\mathbf{V})$ based on generated hypergraph model matrix

$$A_m(\mathbf{V}) = A_{01} A_{12} A_{23} \dots \dots \dots A_{(m-1)m} \quad (3)$$

The general concept and procedure of IPVTS (Fig. 1, Fig. 2) consists of the followings steps:

- *create* the intelligent objects with visual information using mechanisms of homogeneous fractal interpreting visual images (Fig. 1, Fig. 2) . This method is based on self-similarity of atomic and composite prototypes and consists in modeling of composite prototypes by several smaller fragments of themselves. The image is sliced in segments of irregular or regular form using the minimum execution time $T(\mathbf{V})$ of prototypes (Fig. 1). During this concept, our approach allows structuring the homogeneous composition transformation matrix (2) $A_i(\mathbf{V})$ and hypergraph model matrix $A_m(\mathbf{V})$ (3) into components which can be executed in parallel as separate standard parametric prototypes (For example, the matrices A_{01} , A_{12} , etc.).

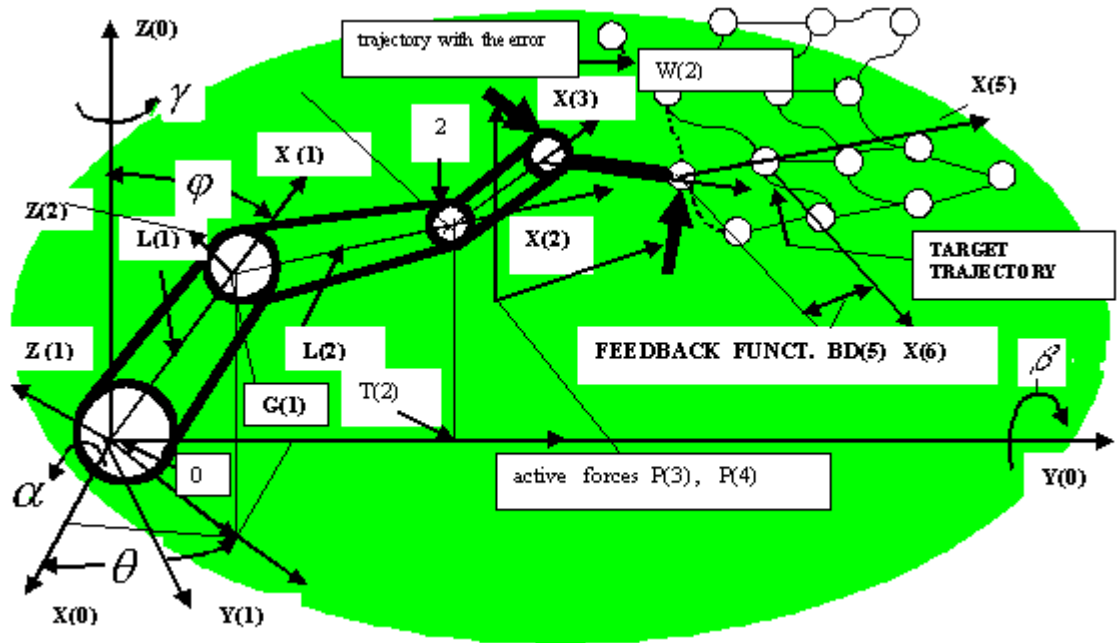


Fig. 1 Functional schema of control strategies for robotics using IPVTS

- *generate* a homogeneous network of nodes, each of which can be of a finite number of V possible states updates in discrete time steps according to the IPVTS model (1) with local identical interaction rule (see Fig. 1, 2);
- *build* initial 3D or m -vector parametric model based on Geometric Modeling System (Fig. 2) and model (1) with parametric function $PF_i(V)$ and parametric constraints $P_i(V)$ (for example the force) using both the IPVTS graphic specification and the full automatic transformation from geometric model or CAD systems;
- *create* the intelligent objects with visual information using mechanisms of homogeneous fractal interpreting visual images (Fig. 1, Fig. 2). This method is based on self-similarity of atomic and composite prototypes and consists in modeling of

composite prototype by several smaller fragments of themselves. The image is sliced into segments of irregular or regular form using the minimum execution time $T(\mathbf{V})$ of prototypes (Fig. 1). During this concept, our approach allows structuring the homogeneous composition transformation matrix (2) $\mathbf{A}_i(\mathbf{V})$ and hypergraph model matrix $\mathbf{A}_m(\mathbf{V})$ (3) into components which can be executed parallel as separate standard parametric prototypes (For example, the matrices \mathbf{A}_{01} , \mathbf{A}_{12} , etc.).

- *generate* a homogeneous network of nodes, each of which can be of a finite number of V possible states updates in discrete time steps according to IPVTS model (1) with local identical interaction rule (see Fig. 1, 2);
- *build* initial 3D or m -vector parametric model based on Geometric Modeling System (Fig. 2) and model (1) with parametric function $\mathbf{PF}_i(\mathbf{V})$ and parametric constraints $\mathbf{P}_i(\mathbf{V})$ (for example the force) using both the IPVTS graphic specification and the fully automatic transformation from the geometric model or CAD systems;
- *update* a site of nodes of a homogeneous lattice according to a definite heuristic rule and heuristic functions $\mathbf{HF}_i(\mathbf{V})$ that involve a neighborhood of sites around each one. Global control heuristic for all nodes of a model is a feedback function (bias distance) $\mathbf{BD}_i(\mathbf{V})$ for transform of the parametric constraints $\mathbf{P}_i(\mathbf{V})$ such as dimensions and normal/shear forces (Fig.1).

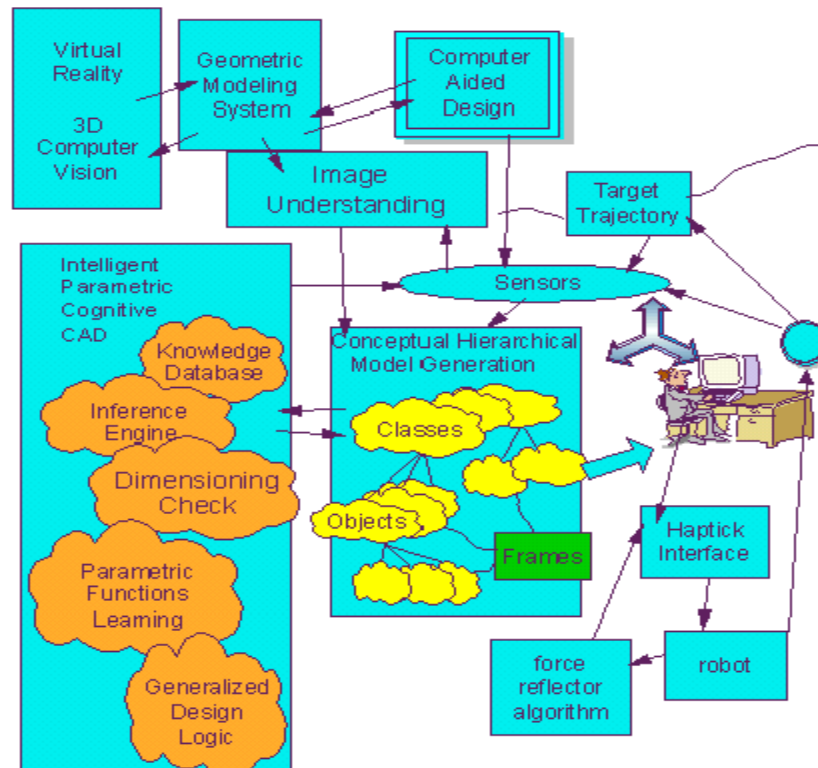


Fig. 2 The human-machine haptic interface based on IPVTS algorithm

- *change* the sites of nodes of a homogeneous lattice according to control constraints $C(\mathbf{V})$, the parametric constraints $P_i(\mathbf{V})$, a definite heuristic rules and heuristic functions $PF_i(\mathbf{V})$ involving the values of its nearest neighbors, where the position of sites around each of the vertices are determined relative to a local system of coordinates. It is implemented as a set of situation-action rules associated with Generalized Design Logic, and Knowledge Database (Fig.2) that implements the local solving method based on triangular systems of neighborhood structures;
- *transform* of the parametric constraints $P_i(\mathbf{V})$ or $C(\mathbf{V})$. IPVTS uses the global control heuristic for all nodes of the model such as a bias distance (feedback) $BD_i(\mathbf{V})$ associated with local system of coordinates from node to node along the edges (Fig.1);
- *determine* and evaluate the several possible parameters of lattices of the neighborhood structures for this step with respect to global structures for this step and global system

of coordinates using both the homogeneous transformation matrix $\mathbf{A}_i(\mathbf{V})$ and heuristic functions $\mathbf{HF}_i(\mathbf{V})$ with learning parametric function of geometric objects (Fig.2);

- *convert* atomic prototypes into many-level concept's system developing the "intelligent" parameterized prototype-classes and prototype-objects using on object-oriented analysis and composition of system from hierarchical modules based on SQL database and knowledge database (Fig. 2.);
- *make* the automatic constraints check with dimensional chains and relationships to the composite prototype;
- *repeat* all consecutive or concurrent steps for all vertices of a network with computing connection weighs $\mathbf{W}_i(\mathbf{V})$, cost weighs $\mathbf{SB}_i(\mathbf{V})$, and genetic code (chromosome) $\mathbf{G}_i(\mathbf{V})$ for search of optimal solutions.

EXPLANATION OF NEW AND UNUSUAL TECHNIQUES

A major difference between the IPVTS methodology and other systems is that it automatically specifies the method requiring parametric automatically coding for a genetic structure of the model, of the sensors and working components that have two styles for rapid prototyping:

- analytically using algebraic notation and formal specification;
- visual with formal parametric geometric representations.

IPVTS allows us to develop a wide variety of learning genetic algorithms created from graphical specifications. This approach has the following advantages over current practice:

- IPVTS is a simple system for very complex behaviors because they have used general and simple rules with heuristic functions $\mathbf{HF}_i(\mathbf{V})$. It allows easily and

quickly derived simulations to visualize a m-dimensional space with an n-vector attached to each point in the space in according to physical phenomena.

- It allows the development of a graphical formal specification that is fully associated with the task of learning the parameters, weights and structure of each of these representations for genetic optimization.
- We can sense both normal and shear forces as well as render texture using the PHANTOM robotic systems. We can also generate texture on a joystick and use a number of force reflecting robotic devices to test this theory and practical application of the methodology.

EXPECTED RESULTS AND THEIR SIGNIFICANCE AND APPLICATION.

IPVTS is a simple system that exhibits very complicated behavior using global control strategies, generic rule and heuristics. IPVTS will allow the implementation of the following:

- Image understanding and converting into high-level robotics/ human-machine interface models, which allow us to remove the operators from potential hazardous environments;
- Automatic generating target trajectory, surface shapes and texture and also other control programs from graphic specification using IPVTS;
- Simulating, visualizing and checking constraints of integrated systems such as forces and pressure using the knowledge database more accurately and usually faster than conventional algorithms;
- Dramatically improving both the quality and speed of control tasks for a system of linear, polynomial or transcendental equations instead of the Newton-Raphson method.

- Run Haptic Research based on the reflected force for PHANToM devices with time delay and employing wave variables to improve situational awareness and operator performance.

The dynamic intelligent parameterization is used, in this system, to provide a great opportunity to transform a visual spatial model into a control process algorithm to achieving the required functional properties with optimal characteristics.

IPVTS IS UNIVERSAL CONSTUCTOR FOR HIGH-LEVEL DESIGN OF COMPLEX SYSTEMS

Based on discrete dynamic systems (Fig. 1) as a high-level computational system with graphic implementation and a combination of heuristics (1) we describe the IPVTS model here for a more detailed discussion. In this report we focus on detailed equations of the model for transformation between local and global coordinates being specified by orientation graphic objects associated with the cell's state array. The homogeneous transformation matrix $A_i(\mathbf{V})$ (3) is key complex component of equation (1) and it captures a set of the following definitions and functions:

A Cell is an Entity That Has Position, Orientation and an Arbitrary Length State Vector

A cell is an entity which has position, orientation and an arbitrary length state vector for physical parameters. The state of a cell is a vector containing values representing position, orientation and size (Fig. 3) where:

- $X_i(\mathbf{V}), Y_i(\mathbf{V}), Z_i(\mathbf{V})$ are coordinates of vertices of vectors associated with \mathbf{V} ;
- $\alpha_{\dots}, \beta_{\dots}, \lambda_{\dots}$ are the rotation angles respect to global $\mathbf{X}(\mathbf{V}), \mathbf{Y}(\mathbf{V}), \mathbf{Z}(\mathbf{V})$ axes of coordinates (world coordinates);

- θ, φ are the Euler's rotation angles with respect to $\mathbf{X}(\mathbf{V})$ and $\mathbf{Z}(\mathbf{V})$ axes;
- α, β, γ are the rotation angles with respect to $\mathbf{X}_i(\mathbf{V}), \mathbf{Y}_i(\mathbf{V}), \mathbf{Z}_i(\mathbf{V})$ axes of coordinates;
- $\alpha_{-}, \alpha_{+}, \beta_{-}, \beta_{+}, \gamma_{-}, \gamma_{+}$ are the rotation angles of nearest neighbors of the local axes with respect to $\mathbf{X}_i(\mathbf{V}), \mathbf{Y}_i(\mathbf{V}), \mathbf{Z}_i(\mathbf{V})$ axes of coordinates;
- $\alpha_{-}, \alpha_{+}, \beta_{-}, \beta_{+}, \gamma_{-}, \gamma_{+}$ are the rotation angles of the next-nearest neighbors of the local axes with respect to $\mathbf{X}_i(\mathbf{V}), \mathbf{Y}_i(\mathbf{V}), \mathbf{Z}_i(\mathbf{V})$ axes of coordinates;
- $k_{x_i}, k_{y_i}, k_{z_i}$ are scaling factors with respect to local $\mathbf{X}_i(\mathbf{V}), \mathbf{Y}_i(\mathbf{V}), \mathbf{Z}_i(\mathbf{V})$ axes of coordinates;

An arbitrary length state vector has a translation distance - \mathbf{L} associated with change of coordinates of $\mathbf{DX}_i(\mathbf{V}), \mathbf{DY}_i(\mathbf{V}), \mathbf{DZ}_i(\mathbf{V})$. The distance \mathbf{L} can be computed as follows

$$\mathbf{L} = \sqrt{\mathbf{DX}(\mathbf{V})^2 + \mathbf{DY}(\mathbf{V})^2 + \mathbf{DZ}(\mathbf{V})^2} \quad (4)$$

where $\mathbf{DX}_i(\mathbf{V}) = \zeta_1(\mathbf{V}) - \mathbf{X}_0(\mathbf{V})$; $\mathbf{DY}_i(\mathbf{V}) = \zeta_2(\mathbf{V}) - \mathbf{Y}_0(\mathbf{V})$; $\mathbf{DZ}_i(\mathbf{V}) = \zeta_3(\mathbf{V}) - \mathbf{Z}_0(\mathbf{V})$

If $\mathbf{L} = 0$, we define $\theta = 0$, $\varphi = 0$ which are the Euler's rotation angles,

otherwise

we compute it as follows :

$$\theta = \arctan(\mathbf{DY}_i(\mathbf{V})/\mathbf{DX}_i(\mathbf{V})) \text{ IF } \mathbf{DX}_i(\mathbf{V}) > 0$$

$$\theta = \pi + \arctan(\mathbf{DY}_i(\mathbf{V})/\mathbf{DX}_i(\mathbf{V})) \text{ IF } \mathbf{DX}_i(\mathbf{V}) < 0$$

$$\theta = \pi/2 \text{ IF } \mathbf{DX}_i(\mathbf{V}) = 0 \text{ and } \mathbf{DY}_i(\mathbf{V}) \geq 0$$

$$\theta = \pi/2 \text{ IF } \mathbf{DX}_i(\mathbf{V}) = 0 \text{ and } \mathbf{DY}_i(\mathbf{V}) < 0$$

$$\varphi = \arccos[\mathbf{DZ}_i(\mathbf{V})/\mathbf{L}]$$

These equations allow us to represent the vectors by adding the corresponding components $L, \theta, \varphi, DX_i(V), DY_i(V), DZ_i(V), \alpha, \beta, \gamma, \alpha', \alpha'', \beta', \beta'', \gamma', \gamma''$ etc. of each vector together using automatic generation of the parameter set for IPVTS properties (see Fig. 11 and equation (1)).

These parameters, constraints and rule transition table (see Fig. 1, Fig. 2, Fig. 3 and Fig. 4) define the specifications of the Visual Thinking Conceptual Parametric Prototyping Language (VTCPPPL) in equations (1), (2), (3) and (4).

In most conventional mechanical CAD systems the geometry determines the dimension parameters such as:

$$L, \theta, \varphi, DX_i(V), DY_i(V), DZ_i(V), \alpha, \beta, \gamma, \alpha', \alpha'', \beta', \beta'', \gamma', \gamma''.$$

The user must accurately define the geometric position of each entity in the drawing and changes to design are made directly in the graphics [Dori 1996]. The dimensions are treated as constraints describing relations among entities.

The IPPMS allows us to solve this problem using a visual thinking paradigm based on dimension-driven approach and a fractal homogeneous transformation.

A Bias Distance (feedback) - $BD_i(V)$ as Tool for the Global Control of Heuristics

To transform of the parametric constraints $P_i(V)$ or $C(V)$, IPPMS uses the global control heuristics for all nodes of model such as a bias distance (feedback) - $BD_i(V)$ associated with local system of coordinates (Fig.1, Fig. 3):

$$BD_i(V) = L' - L$$

where $C(V) = L'$.

If $BD_i(V)$ is not equal to the null vector and dimension - L is changed to according the bias distance $BD_i(V)$; the back transformation is considered :

$$\begin{aligned}
DX_i(\mathbf{V}) &= \sin \varphi \cos \theta \\
DY_i(\mathbf{V}) &= \sin \varphi \sin \theta \\
DZ_i(\mathbf{V}) &= \cos \varphi
\end{aligned}$$

Hypergraph Composition Model IPPMS

If the geometric model has m -vectors associated with \mathbf{V} , \mathbf{E} , $\mathbf{C}_i(\mathbf{V})$, $\mathbf{DB}_i(\mathbf{V})$, $\mathbf{PF}_i(\mathbf{V})$, we will have the general matrix transformation model that is generated automatically by IPPMS:

$$\mathbf{A}_m(\mathbf{V}) = \mathbf{A}_{01} \mathbf{A}_{12} \mathbf{A}_{23} \dots \mathbf{A}_{(m-1)m} \quad (5)$$

Note that it is possible that two types of notation of the resulting matrix are equivalent:

$$\mathbf{A}_m(\mathbf{V}) = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{matrix} x \\ y \\ z \end{matrix} = \begin{bmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{bmatrix} \quad (6)$$

Equation (6) is a result of multiplying matrices for m elements of a composite complex matrix (5). Therefore, as a result, both types (5) and type (6) can be used. In another case which can be considered, also the equations systems also can be written for each cell of the dynamic discrete model:

$$\begin{aligned}
x' &= \alpha x + \beta y + \gamma z \\
y' &= \delta x + \epsilon y + \zeta z \\
z' &= \eta x + \theta y + \iota z
\end{aligned} \quad (7)$$

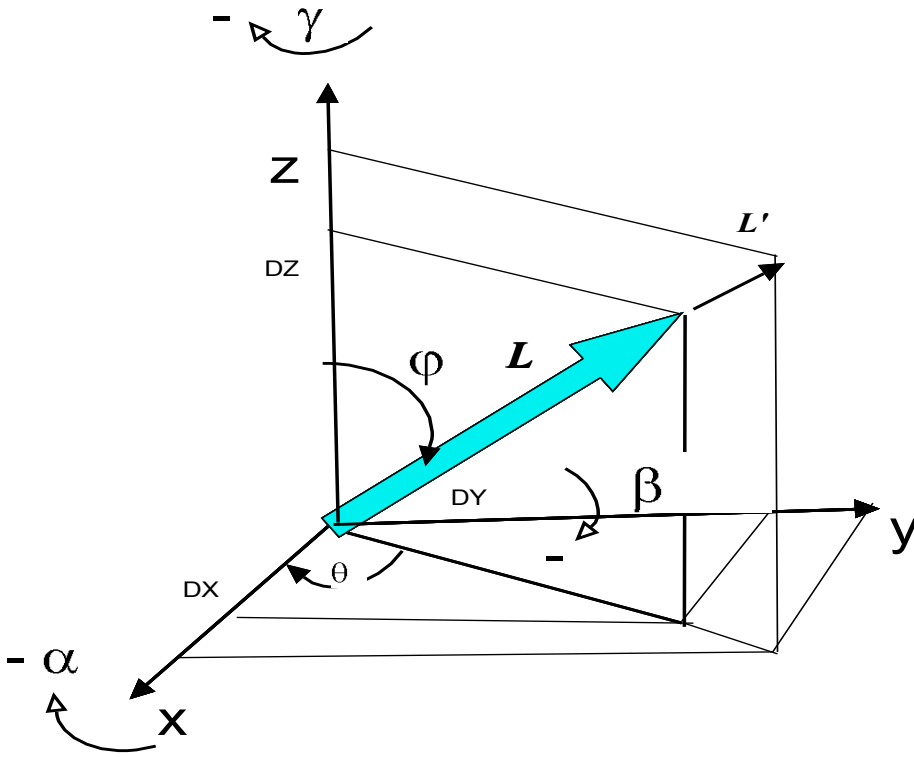


Fig. 3 Graphic representation of arbitrary vector. An arbitrary length state vector has a translation distance - L associated with change of coordinates of $DX_i(V)$, $DY_i(V)$, $DZ_i(V)$ and the θ , ϕ Euler's angles and rotation angles.

We use the hypergraph composition model IPPMS to describe a vector representation of a single cell and relative position and orientation in discrete dynamical networks. In order to describe the location and orientation of each object and primitive relative to its neighbor, a frame is attached to each cells, then we specify a set of parameters that characterize this frame such as $HF_i(V)$ -heuristic functions and rules, $P_i(V)$ - parametric constraints, $PF_i(V)$ - parametric function; bias distance - $BD_i(V)$, $TO_i(V)$ -text; $W_i(V)$ - connection weighs; $PB_i(V)$ -local feedback function, and $G_i(V)$ -genetic code (see equation (1)).

In reference to other work, we provide to generate automatically the graphical specification and include the feature of translating, rotational, scaling and general displacement matrix operators which can easily be programmed on a computer for source code to manipulate automatically the complex object.

The existing work that describes the key problem using matrix transformation methods is still based on an informal manual description of the components involving homogeneous transformation matrices (Lin and Tsai, 1996) and uses queries which may not be posed in standard programming notation. Hence it is then automatically translated into algebraic notation and visual parametric specification using a 3D CAD/CAM model. The position, structure, topological links and orientation of the k th frame, $(xyz)_k$, with respect to frame $(xyz)_j$, also, cannot be written automatically in the CAD/CAM model. The resulting models also cannot be directly imported into feature-based CAD systems without a loss of the semantics and topological information inherent in feature-based representations. IPVTS allows us to solve this problem based on a high-level visual representation of the objects in a successful manner.

Physical Setup and System Equations for the PHANToM Mechanism

Geometric Modeling System (GMS) for the PHANToM mechanism is described in this section which allows us to build a 3D vector parametric model with relationships and parameters (dimension-driven wireframe). GMS uses the heuristic modeling algorithm and coordinate conversion methods (see Fig. 2) that permit us to build the optimal parametric history and "intelligent" objects. The heuristic modeling algorithm should be capable of handling both goal-driven and data-driven strategies. It presents the

mathematical calculations used by OpenGL and MS Visual C++ for high-level object-oriented design.

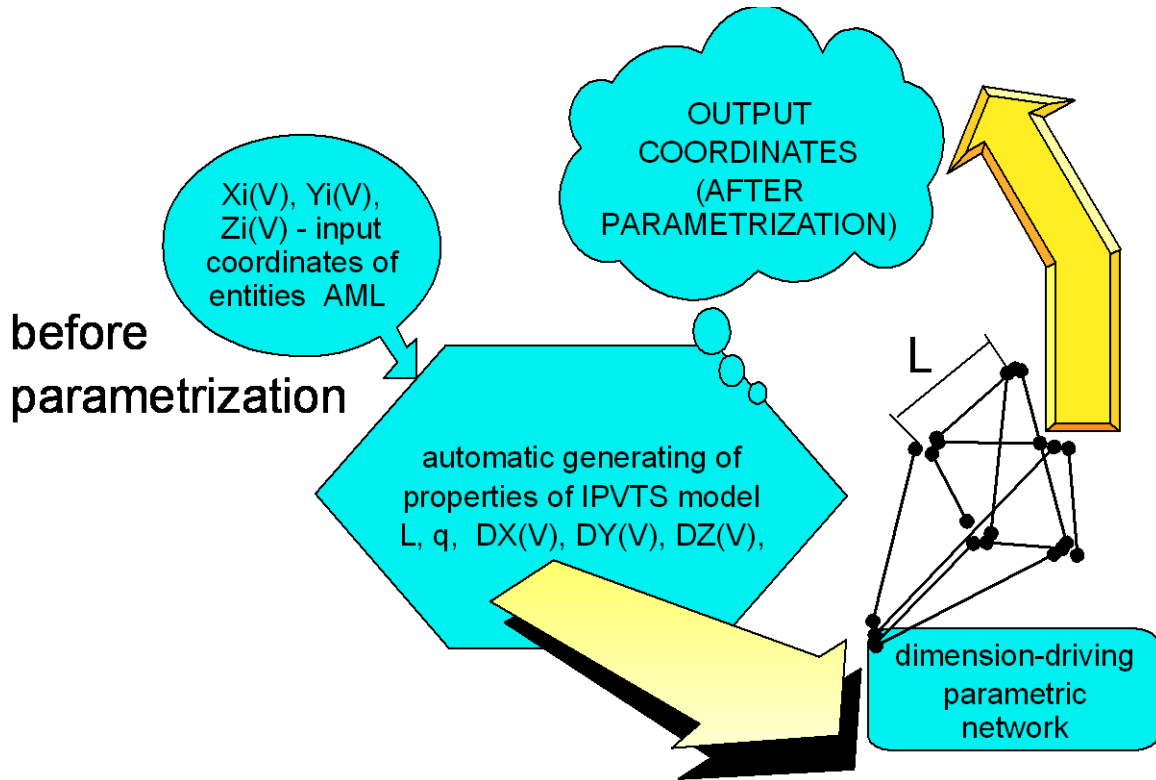


Fig 4. The automatic generation of the dimension-driving parametric network

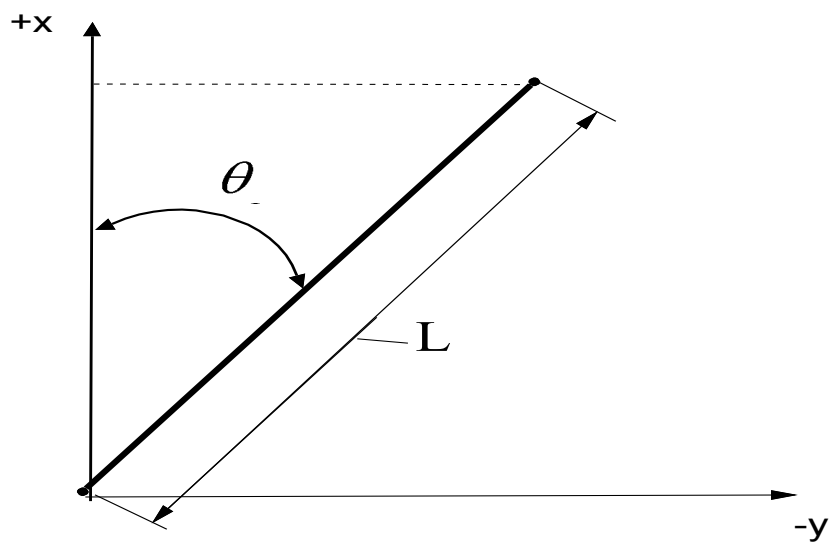


Fig. 5 Rotation of Projected X Y through the angle θ

$$\begin{aligned}
L &= l_1 \cos\theta + l_2 \cos\theta \\
X &= \cos\theta (L_1 \cos\theta + l_2 \cos\theta) \\
Y &= \sin\theta (L_1 \cos\theta + l_2 \cos\theta)
\end{aligned} \tag{8}$$

Computing the forces with respect to x, y, z axes :

$$\begin{aligned}
P_X &= P_X' \cos\theta = l_1 c_2 + l_2 c_{23} = l_1 c_2 l_1 + l_2 c_1 c_{23} \\
P_Y &= P_X' \sin\theta = l_1 c_2 + l_2 c_{23} = l_1 c_2 s_1 + l_2 c_{23} s_1 \\
P_Z &= -l_1 \sin\theta - l_2 \sin\theta
\end{aligned} \tag{9}$$

Define the Jacobian Matrix:

$$\begin{aligned}
\frac{\partial P_X}{\partial \theta_1} &= -c_2 s_1 - l_2 c_{23} s_1 & \frac{\partial P_Z}{\partial \theta_1} &= 0 & \frac{\partial P_Y}{\partial \theta_1} &= l_1 c_2 c_1 + l_2 c_{23} c_1 \\
\frac{\partial P_X}{\partial \theta_2} &= -c_1 s_2 - l_2 c_1 s_{23} & \frac{\partial P_Z}{\partial \theta_2} &= -c_2 - l_2 c_{23} & \frac{\partial P_Y}{\partial \theta_2} &= -s_1 s_2 - l_2 s_1 s_{23} \\
\frac{\partial P_X}{\partial \theta_3} &= -l_1 s_{23} & \frac{\partial P_Z}{\partial \theta_3} &= -l_2 c_{23} & \frac{\partial P_Y}{\partial \theta_3} &= -l_1 s_1 s_{23}
\end{aligned} \tag{10}$$

Computing the forces respect to x, y, z axes :

$$J = \begin{bmatrix} -l_1 c_2 s_1 - l_2 c_{23} s_1 & -l_1 c_1 s_2 - l_2 c_1 s_{23} & -l_2 c_1 s_{23} \\ l_1 c_2 c_1 + l_2 c_{23} c_1 & -l_1 s_1 s_2 - l_2 s_1 s_{23} & -l_2 s_1 s_{23} \\ 0 & -l_1 c_2 - l_2 c_{23} & -l_2 c_{23} \end{bmatrix}$$

Then the torque is specified via:

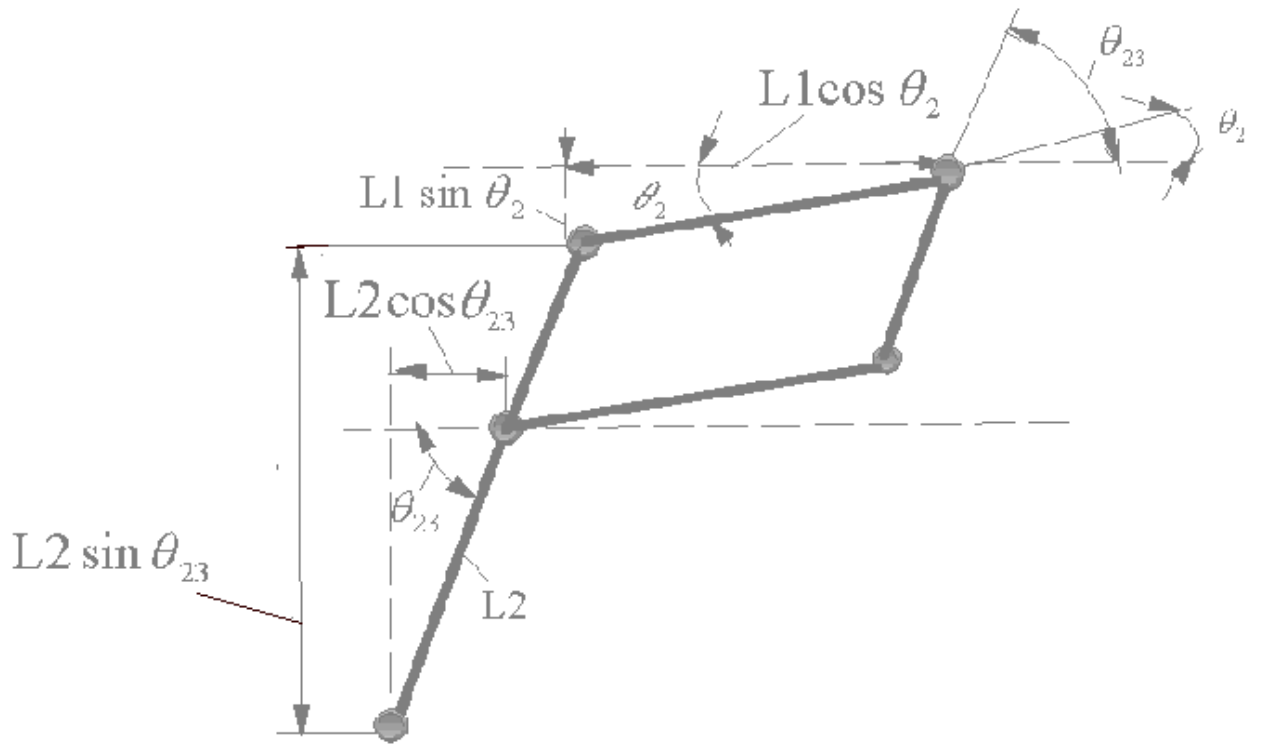
$$\tau = J^T F$$

$$\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} * \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

Finally, we write the torque equation, which is the algorithm used in the modeling

$$\begin{aligned}
\tau_x &= (l_{11} F_{mx} + l_{12} F_{my} + l_{13} F_{mz}) \text{ gradual} * k \\
\tau_y &= (l_{21} F_{mx} + l_{22} F_{my} + l_{23} F_{mz}) \text{ gradual} * k \\
\tau_z &= (l_{31} F_{mx} + l_{32} F_{my} + l_{33} F_{mz}) \text{ gradual} * k
\end{aligned} \tag{11}$$

For a description of the torque equations, the Jacobian Matrix Method proposed by Massie and Salisbury [Installation & Tech. Manual, Massie 1994] for the



$$\theta_{23} = \theta_2 + \theta_{23}$$

$$Z = r_1 \sin \theta_2 + r_2 \sin \theta_{23}$$

Fig. 6 PHANToM Links Kinematics and Geometric Parameters as Components of Structure.

PHANToM was employed. However, the IPVTS can generate a more effective hypergraph model based on homogenous matrix transformations and high-level logic.

ALGORITHM OF BASIC TELEOPERATION

For the virtual reality environment with a virtual force F_e (see Fig 7) the following equations of motion exist for the x axis:

$$F_e = M_e \ddot{x}_e(t) + B_e \dot{x}_e(t) + \zeta_e x_e(t) \quad (12)$$

where $x_e(t)$ denote the displacement of the slave arm in the e environment, $\dot{x}_e(t)$

velocity and $\ddot{x}_e(t)$ acceleration are of the slave arm in the e environment and

M_e , B_e and K_e are the inertia, damping, and stiffness, respectively. In this equation, B_e is physical constant that characterizes the transmission line as the damper (friction) in the environment.

Let $F_e = F_s$, resulting in the following equation :

$$F_s = M_s \ddot{x}_s(t) + B_s \dot{x}_s(t) + K_s x_s(t) \quad (13)$$

For a master, we have respectively:

$$F_m = M_m \ddot{x}_m(t) + B_m \dot{x}_m(t) + K_m x_m(t) \quad (14)$$

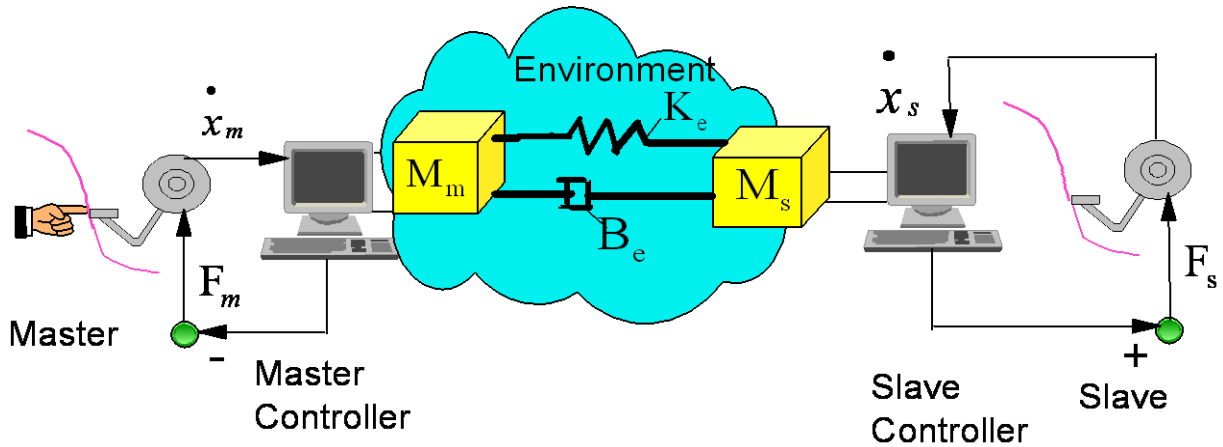


Fig. 7 Basic Teleoperator without Time Delay based on two inertia - M_m , M_s connected via a spring - K_e and damper B_e in the environment.

A bilateral teleoperation system can be represented as a translation dynamic system based on a three dimensional model:

$$F_s = -F_m = M_e [\ddot{x}_m(t) - \ddot{x}_s(t)] + B_e [\dot{x}_m(t) - \dot{x}_s(t)] + K_e [x_m(t) - x_s(t)] \quad (15)$$

where $M_e = M_m = M_s$, $B_e = B_m = B_s$ and $K_e = K_m = K_s$

K_s is the virtual stiffness of the slave, K_e is virtual stiffness and K_m is a stiffness of the operator's hand

$$\begin{aligned}
F_{sx} = -F_{mx} &= M_e [\ddot{x}_m(t) - \ddot{x}_s(t)] + B_e [\dot{x}_m(t) - \dot{x}_s(t)] + K_e [x_m(t) - x_s(t)] \\
F_{sy} = -F_{my} &= M_e [\ddot{y}_m(t) - \ddot{y}_s(t)] + B_e [\dot{y}_m(t) - \dot{y}_s(t)] + K_e [y_m(t) - y_s(t)] \\
F_{sz} = -F_{mz} &= M_e [\ddot{z}_m(t) - \ddot{z}_s(t)] + B_e [\dot{z}_m(t) - \dot{z}_s(t)] + K_e [z_m(t) - z_s(t)]
\end{aligned} \tag{16}$$

A bilateral teleoperation system can be represented:

$$\begin{aligned}
F_{sx} = -F_{mx} &= -M_e [\ddot{x}_m(t) - \ddot{x}_s(t)] - B_e [\dot{x}_m(t) - \dot{x}_s(t)] - K_e [x_m(t) - x_s(t)] \\
F_{sy} = -F_{my} &= -M_e [\ddot{y}_m(t) - \ddot{y}_s(t)] - B_e [\dot{y}_m(t) - \dot{y}_s(t)] - K_e [y_m(t) - y_s(t)] \\
F_{sz} = -F_{mz} &= -M_e [\ddot{z}_m(t) - \ddot{z}_s(t)] - B_e [\dot{z}_m(t) - \dot{z}_s(t)] - K_e [z_m(t) - z_s(t)].
\end{aligned} \tag{17}$$

The master will reflect the virtual contact force of the slave back to the operator and the slave will follow the position of the master exactly because there is no time delay in this case.

$$\begin{aligned}
F_{mx} = -F_{sx} &= M_e [\ddot{x}_m(t) - \ddot{x}_s(t)] + B_e [\dot{x}_m(t) - \dot{x}_s(t)] + K_e [x_m(t) - x_s(t)] \\
F_{my} = -F_{sy} &= M_e [\ddot{y}_m(t) - \ddot{y}_s(t)] + B_e [\dot{y}_m(t) - \dot{y}_s(t)] + K_e [y_m(t) - y_s(t)] \\
F_{mz} = -F_{sz} &= M_e [\ddot{z}_m(t) - \ddot{z}_s(t)] + B_e [\dot{z}_m(t) - \dot{z}_s(t)] + K_e [z_m(t) - z_s(t)]
\end{aligned} \tag{18}$$

The Virtual Force Reflecting Algorithm is Based on a Physics Model and on the Torque, Force, Position, and Velocity Limits of PHANToM.

The PHANToM represents a new technology based on force reflecting, robotic, computer input devices that can actually feel virtual objects [Installation & Technical Manual 1995, Massie 1994]. The PHANToM contains three motor-encoders which

control the reflected F_x , F_y , F_z reaction's forces computed via the 3D physics-based model. An example such as Hooke's Law is given as follows:

$$\begin{aligned} F_x &= k * \Delta_x(t) \\ F_y &= k * \Delta_y(t) \\ F_z &= k * \Delta_z(t) \end{aligned} \quad (19)$$

where the k - stiffness of the virtual rigid wall (box or arbitrary virtual objects); $\Delta_x(t), \Delta_y(t), \Delta_z(t)$ are the depth of penetration during interaction of the user point (the effector PHANToM) with virtual objects with respect to the x, y, z axis of the force-reflecting environment. The encoder motor represents the $\Delta_x(t), \Delta_y(t), \Delta_z(t)$ position information for simulation of the virtual forces that the user feels. Fig. 8 and Fig. 9 describe the basic conception of the virtual force PHANToM robot for the example of a virtual wall. The PHANToM devices can sense the $\Delta_x(t), \Delta_y(t), \Delta_z(t)$ position information with a finite frequency only with correspondence of the encoding signals - e_0, e_1, e_2 (see Fig. 18) and determine the F_x, F_y, F_z reflected forces respectively (Fig. 8, Fig. 9 and Fig. 19 display these virtual forces on our virtual reality scene).

After this calculation, but before transmitting these values to the Phantom, the controller checks to see if these values are within the range from $\{-MAX_Tm\}$ to $\{MAX_Tm\}$ specified by the PHANToM instructions the largest values are the limit $MAX_Tm = \pm 000$ for protection of the motor-encoders (the saturating motor condition). If not, the controller scales the torque to keep it within range from $\{-MAX_Tm\}$ to $\{MAX_Tm\}$ (see Fig. 8 and Fig. 9).

The potential energy gain of a virtual spring leads to instability if the $k \rightarrow$ stiffness coefficient is increased and the torque limit MAX_Tm can be violated.

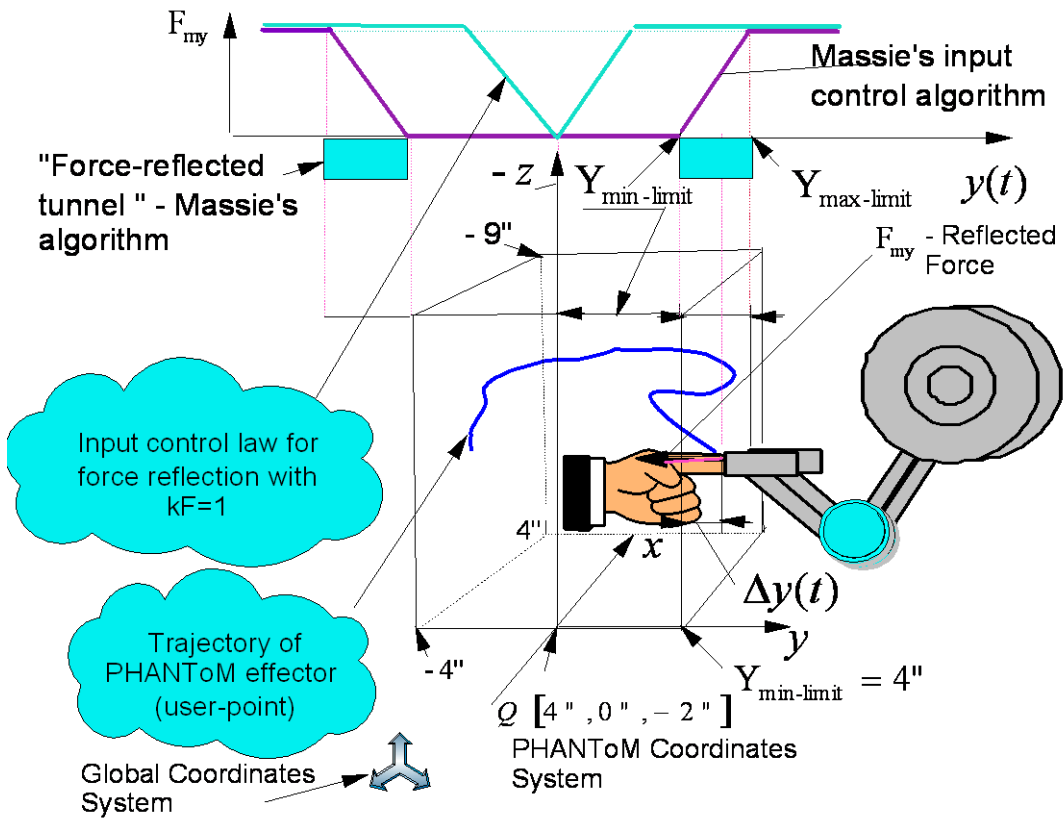


Fig. 8 The interaction of PHANToM effector with “virtual wall”. The virtual force reflects the reaction of virtual object in “force-reflection tunnel”.

See equations (20) for a check of a saturating motor below. An experimental example based on visual modeling on Fig. 30a, in the sequel, illustrates the instability of the control for PHANToM system when the torque limit MAX_Tm has been violated for all x, y, z axes.

On the other hand, a low value of stiffness coefficient represents an elastic virtual wall which can be desirable an example for modeling of a pneumatic system.

$$\text{if } (Tm[0] > MAX_Tm) \text{ } Tm[0] = MAX_Tm;$$

$$\text{if } (Tm[0] < -MAX_Tm) \text{ } Tm[0] = -MAX_Tm;$$

$$\begin{aligned}
& \text{if } (Tm[1] > MAX_Tm) \ Tm[1] = MAX_Tm; \\
& \text{if } (Tm[1] < -MAX_Tm) \ Tm[1] = -MAX_Tm; \\
& \text{if } (Tm[2] > MAX_Tm) \ Tm[2] = MAX_Tm; \\
& \text{if } (Tm[2] < -MAX_Tm) \ Tm[2] = -MAX_Tm;
\end{aligned} \tag{20}$$

The Massie’s algorithm for simulation of virtual reaction forces is based on standard virtual objects such as virtual walls, virtual planes, virtual spheres and virtual cubes. In our experiments we will use the conception of a virtual wall. However, we believe what is needed in the future is an universal computer model in the future for arbitrary virtual model based on IPVTS.

Fig. 9 illustrates the graphical interpretation of the Massie’s algorithm for computing the PHANToM virtual forces. As it is portrayed, the “force-reflection tunnel” represents the force responses only inside the virtual wall because this algorithm considers the space inside the small virtual box as “free space” with the stiffness $k=0$ and the virtual reflected force is “zero” in this case.

The virtual forces inside the virtual wall, after transformation of equations (19), can be written as:

$$\begin{aligned}
F_x &= [X_{\text{min-limit}} - \cdot(t)] \\
F_y &= [Y_{\text{min-limit}} - \cdot(t)] \\
F_z &= [Z_{\text{min-limit}} - \cdot(t)]
\end{aligned} \tag{21}$$

where $X_{\text{min-limit}}$, $Y_{\text{min-limit}}$, $Z_{\text{min-limit}}$ are the lower limits of the virtual wall when an intersection of these restrictions by user the user-point appears in the “force-reflection tunnel”.

Microsoft Visual C++ code from Massie’s algorithm describing the “force-reflection tunnel” in the virtual wall is displayed in Fig.10.

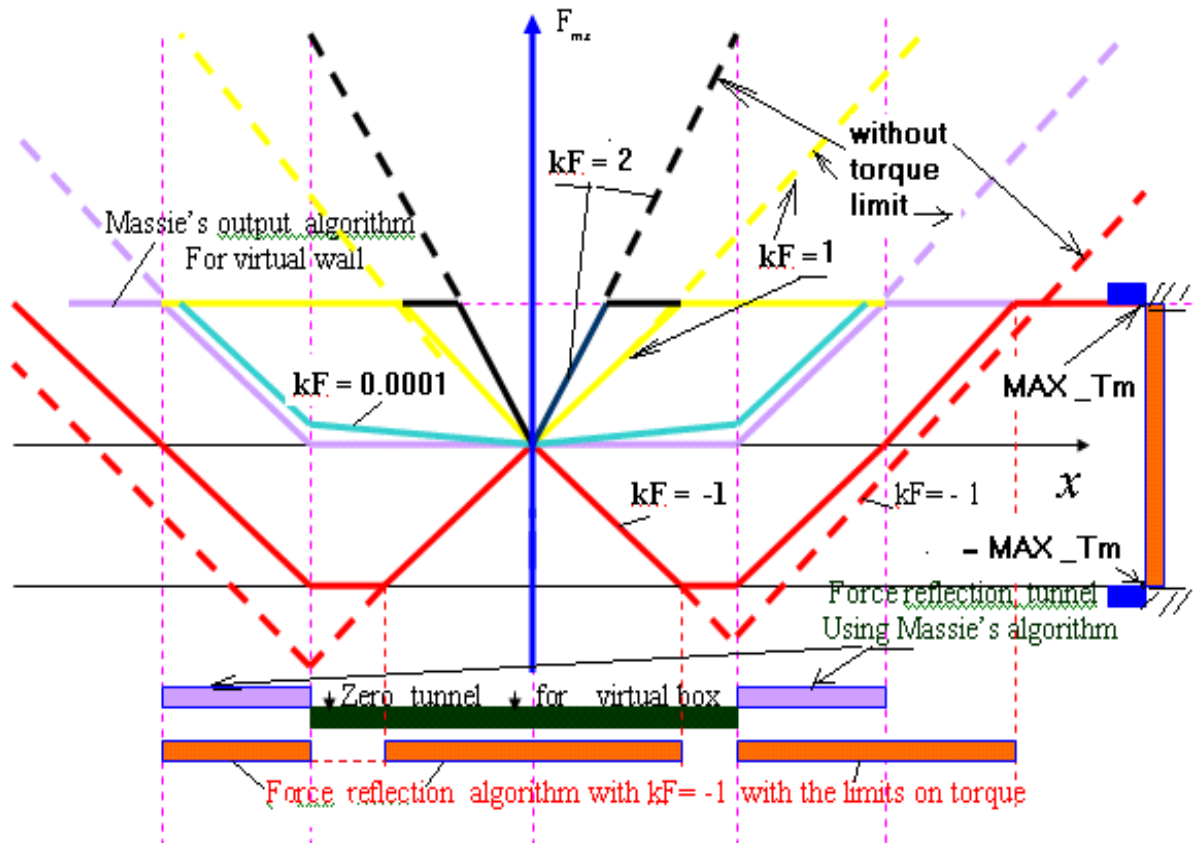


Fig. 9 The graphical interpretation of the Massie's algorithm and our input control laws for computing the PHANToM virtual forces

The simulation results, with this condition, indicate (see Fig. 12) that virtual forces F_{mx}, F_{my}, F_{mz} have the positive values for negative positions $x(t), y(t), z(t)$ and approximately “zero” values for these forces involving positive $x(t), y(t), z(t)$ positions in free “non-force-reflection” space. Here the virtual force is not practically controlled. Although this case can represent some interest involving real control laws, we will consider as more useful other input control laws for modeling. Control algorithms on Fig. 11 represented in the work [Repperger 1997] for force responses involving a different input control law of the PHANToM device.

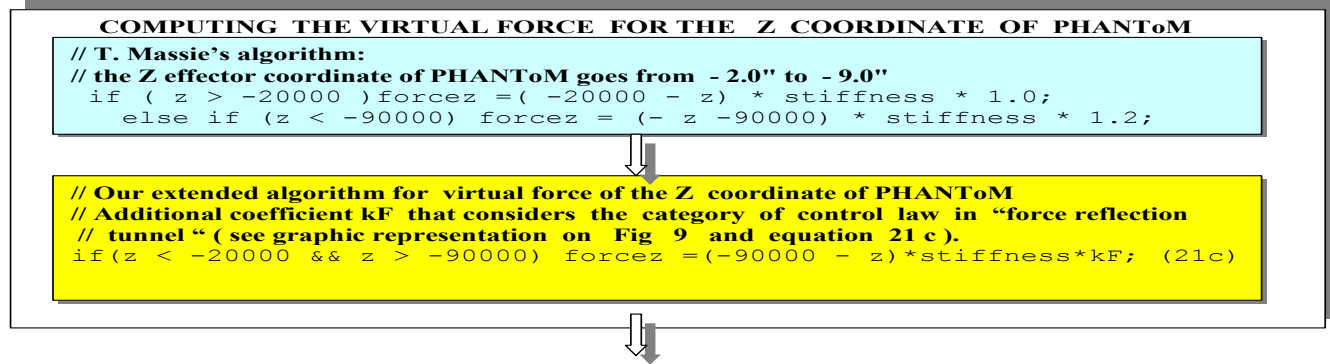
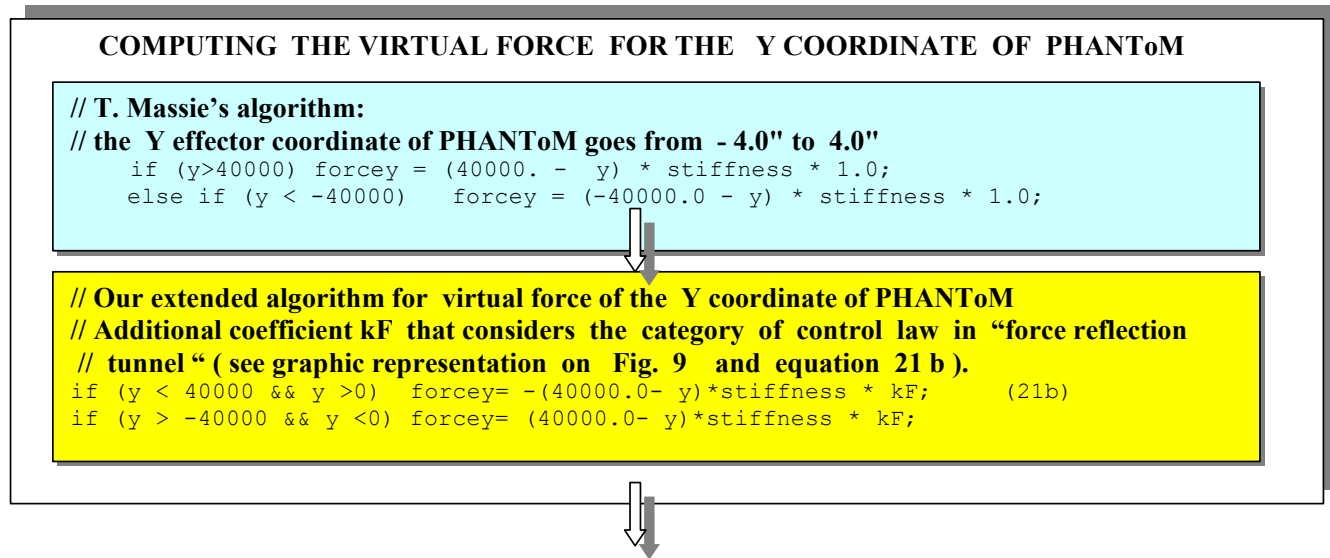
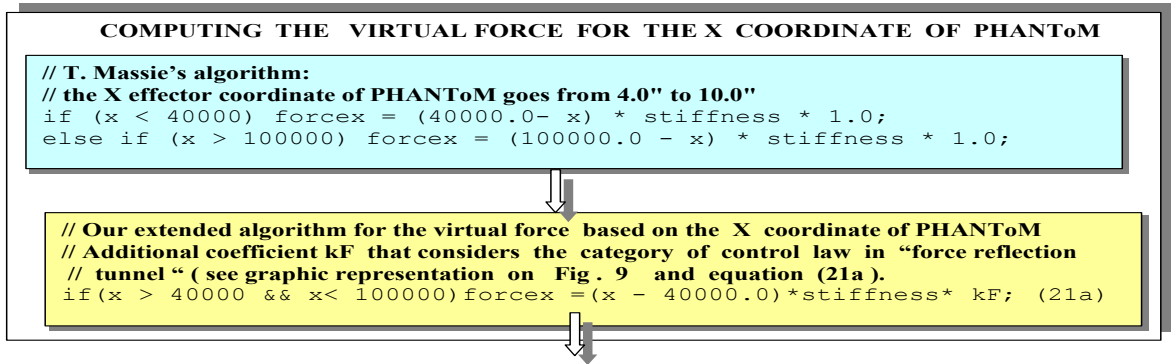


Fig. 10 The algorithm for computing of virtual forces considering the “force-reflection tunnel” for different control strategies. See the chart related with this algorithm in Fig. 9

We use for this goal, the additional coefficient k_F , that considers the category of the control law (see graphic representation on Fig. 8 , Fig 9 and Fig. 10 and equations 21a , 21b, 21c on Fig. 10)

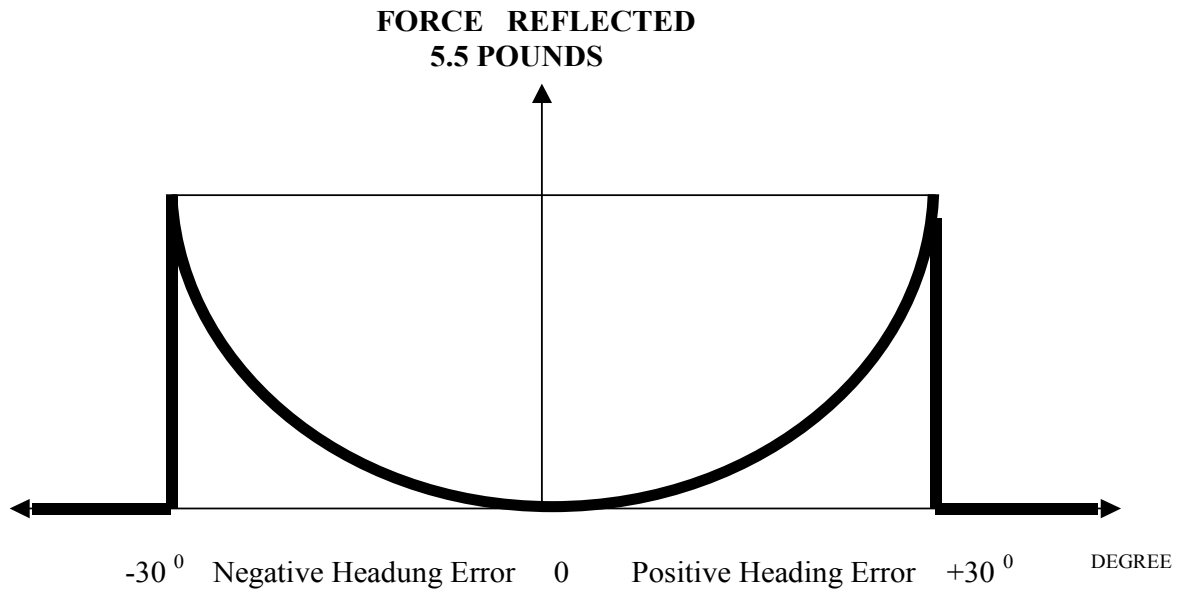


Fig. 11 Force reflected versus heading error $F_R = a|\Theta - 00^\circ|^3$ according to work [Repperger,1997]

We are extending the Massie's algorithm for computing the Phantom virtual forces and add the condition when the "force-reflected tunnel" occurs in a small virtual box also (see Fig. 9) where additional conditions indicated in the yellow color block).

Analysis of Input Control Laws for Nonlinear Real-Time Feedback Strategies

For evaluating the performance of the PHANToM control strategies based on additional coefficient k_F we analyze several cases using experimental real-time modeling:

- *In case $k_F \approx 1$, an example $k_F = 0.0000$, we utilize in the input control law which is similar to the control strategy in Massie's algorithm . The real-time*

results for the $F_{mx}(t)$, $F_{sy}(t)$, $F_{mz}(t)$ reflected forces on Fig. 12 display these forces that have positive values for negative positions $x(t), y(t), z(t)$ and almost “zero” values these forces for positive $x(t), y(t), z(t)$ positions in free “non-force-reflection” space. Here the virtual force is not practically controlled. In this “non-force-reflection” interval control strategy we will not apply a force and torque until the velocity and position comes within the set limits. It causes unstable dynamics of the system because the pulsation as the force response occurs as the torque has a wide interval change from zero to a maximum limit of the torque (see Fig. 29). Fig. 29 illustrates that the system is unstable despite a small value of torque on the master $\tau_{\dots x}$ and on the slave $\tau_{\dots x}(t - \tau)$ with time delay $T=0.1$ sec. The real-time experiments show, also, that the more pulsation of the reflected forces and torque occurs with time delay. The similar results were received for small negative coefficient k_F , as an example

$$k_F = -0.00001.$$

- ***In case $k_F = 1$ we have a proportional input control law*** in small virtual box and the “force-reflected tunnel” has a more wide interval. Fig. 13 displays these forces which have the positive values for negative positions $x(t), y(t), z(t)$. Almost positive values for these forces OCCUR for positive $x(t), y(t), z(t)$ positions in extended “force-reflection” space. Here the virtual force is also controlled and causes additional amplitude for reflected forces in a virtual box space. This strategy allows the control of the force inside a small virtual box interval and to improve the dynamics.

[lbs x 10000] for force value

[in x 10000] for position value

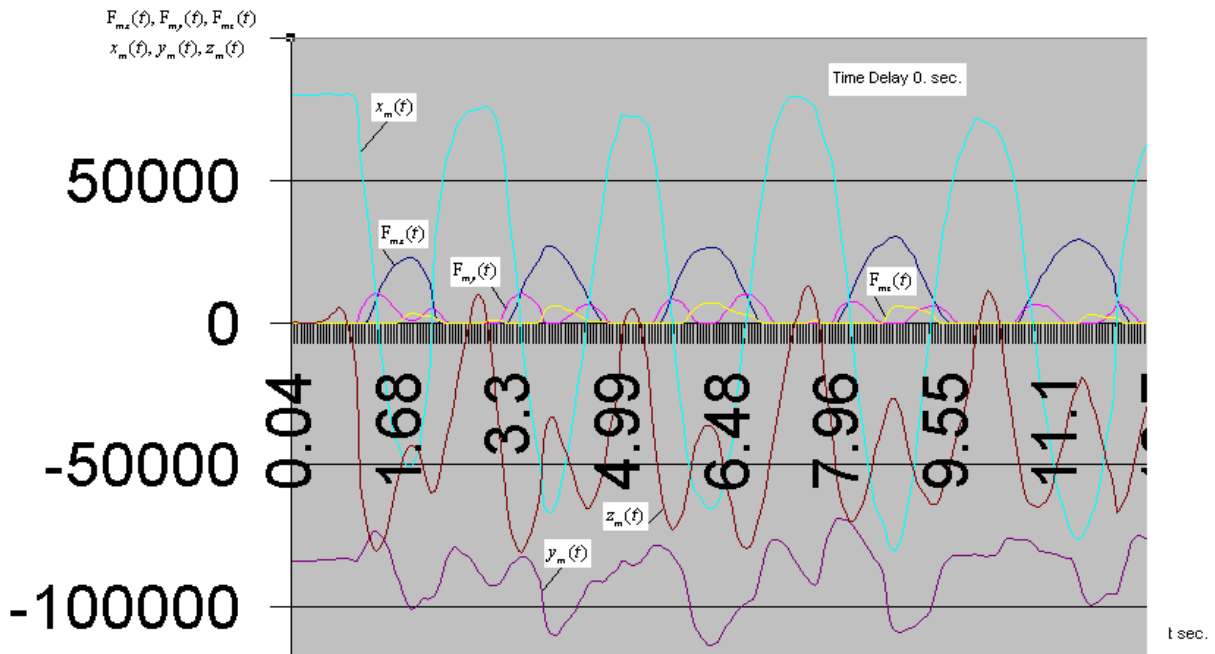


Fig. 12 The $F_{mx}(t), F_{my}(t), F_{mz}(t)$ virtual forces response with respect to the $x_m(t), y_m(t), z_m(t)$ position varying. The coefficient $k_F = 0$ for force computation in the algorithm on Fig 10 .Time Delay T=0 sec.

- ***In case $k_F = 2$ we have proportional input control law*** also in small virtual box.

Fig. 11 displays these forces which have the positive values $F_{mx}(t)$ for negative positions $x(t)$ and positive values for these forces for positive $x(t)$ positions.

In extended “force-reflection” space, the virtual force has adouble additional amplitude for reflected forces in THE virtual box space. However, it is necessary

to already note that for $F_{my}(t)$ force, we have double additional amplitude for k_F

= 1 .

[lbs x 10000] for force value
[in x 10000] for position value

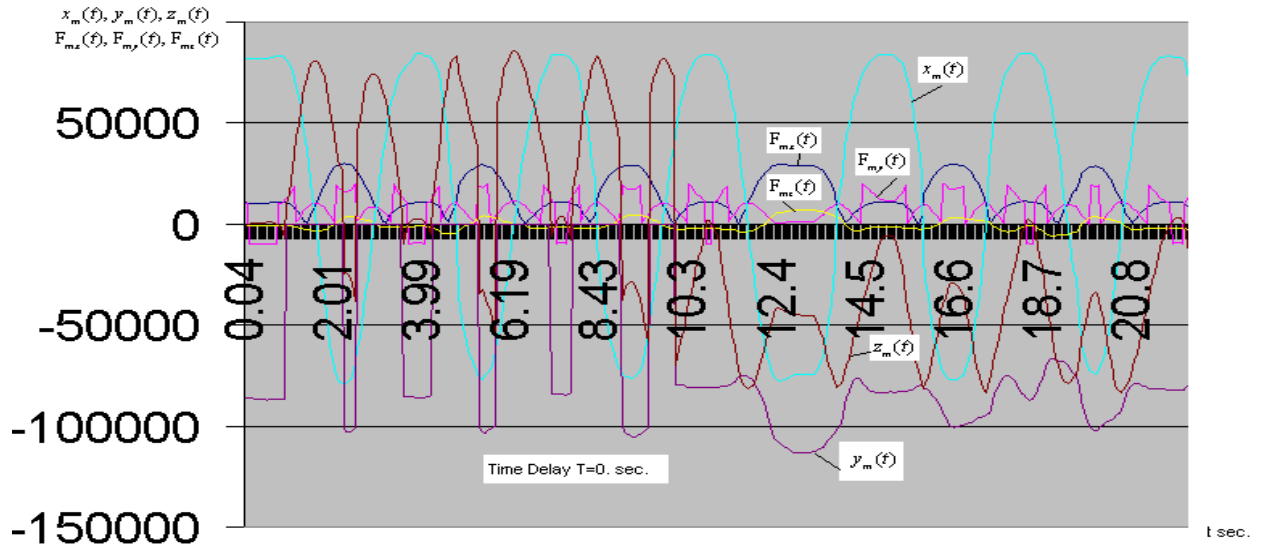


Fig. 13 The $F_{mx}(t), F_{my}(t), F_{mz}(t)$ virtual forces responses with respect to the $x_m(t), y_m(t), z_m(t)$ position varying . The coefficient $k_F = 1$ for force computing in algorithm on Fig 10 . Time Delay $T=0$ sec.

- ***In case $k_F = 3$ we have proportional input control law*** also in small virtual box and Figure 16 displays these forces which have the positive values for the negative positions $x(t), y(t), z(t)$.

Also positive values exist for these forces for positive $x(t), y(t), z(t)$ positions in extended “force- reflection” space where the virtual force has double additional amplitudes for reflected forces in a virtual box space. However, it is necessary to already note that for $F_{my}(t)$ force we have double additional amplitude for $k_F = 1$.

- ***In case $k_F = -1$ we have proportional input control law*** which is similar to Hooke’s Law

(see Fig. 9) for virtual force in “force-reflected tunnel” both inside a small box and outside i.e. in a “virtual wall” with more additional stiffness coefficients. The PHANToM virtual forces characteristics are positive for negative values of

$x(t), y(t), z(t)$ positions (see Fig. 16) and for negative values of forces involving positive values of positions. This provides a more successful application of control for virtual and real objects. As is shown in Fig. 16, these results are similar to results for virtual forces inside the “force-reflected tunnel” described in the work [Repperger 1997]. It is also useful to note that despite an application of proportional law for these inputs, we have a cubic-like law for the output as it is shown in the work [Repperger 1997] – see Fig. 11.

This method provides more flexible signal processing for control in our case and more stable dynamic characteristics. Outside the virtual wall, when the condition of the saturating motor is based on torque limits which are violated, we have the space in environment with absolutely rigid characteristics, such as it is shown in equations (20). In this case we have following strategies for control:

- ***The Phantom controller scales the torque, the positions, velocities and forces*** to keep it within range (the virtual force is calculated from the following values of the positions and velocities):

$$\begin{aligned}
 & \text{from } T_m[0] = -MAX_Tm \text{ to } T_m[0] = MAX_Tm \\
 & \text{from } T_m[1] = -MAX_Tm \text{ to } T_m[1] = MAX_Tm; \\
 & \text{from } T_m[2] = -MAX_Tm \text{ to } T_m[2] = MAX_Tm; \\
 & x(t) = \max(|\dot{x}|) \text{ for } T_m[0] < -MAX_Tm \text{ or } T_m[0] > MAX_Tm \quad (22) \\
 & y(t) = \max(|\dot{y}|) \text{ for } T_m[2] < -MAX_Tm \text{ or } T_m[2] > MAX_Tm \\
 & z(t) = \max(|\dot{z}|) \text{ for } T_m[1] < -MAX_Tm \text{ or } T_m[1] > MAX_Tm \\
 & \dot{x}(t) = \max(|\dot{x}|) \text{ for } T_m[0] < -MAX_Tm \text{ or } T_m[0] > MAX_Tm \\
 & \dot{y}(t) = \max(|\dot{y}|) \text{ for } T_m[2] < -MAX_Tm \text{ or } T_m[2] > MAX_Tm \\
 & \dot{z}(t) = \max(|\dot{z}|) \text{ for } T_m[1] < -MAX_Tm \text{ or } T_m[1] > MAX_Tm
 \end{aligned}$$

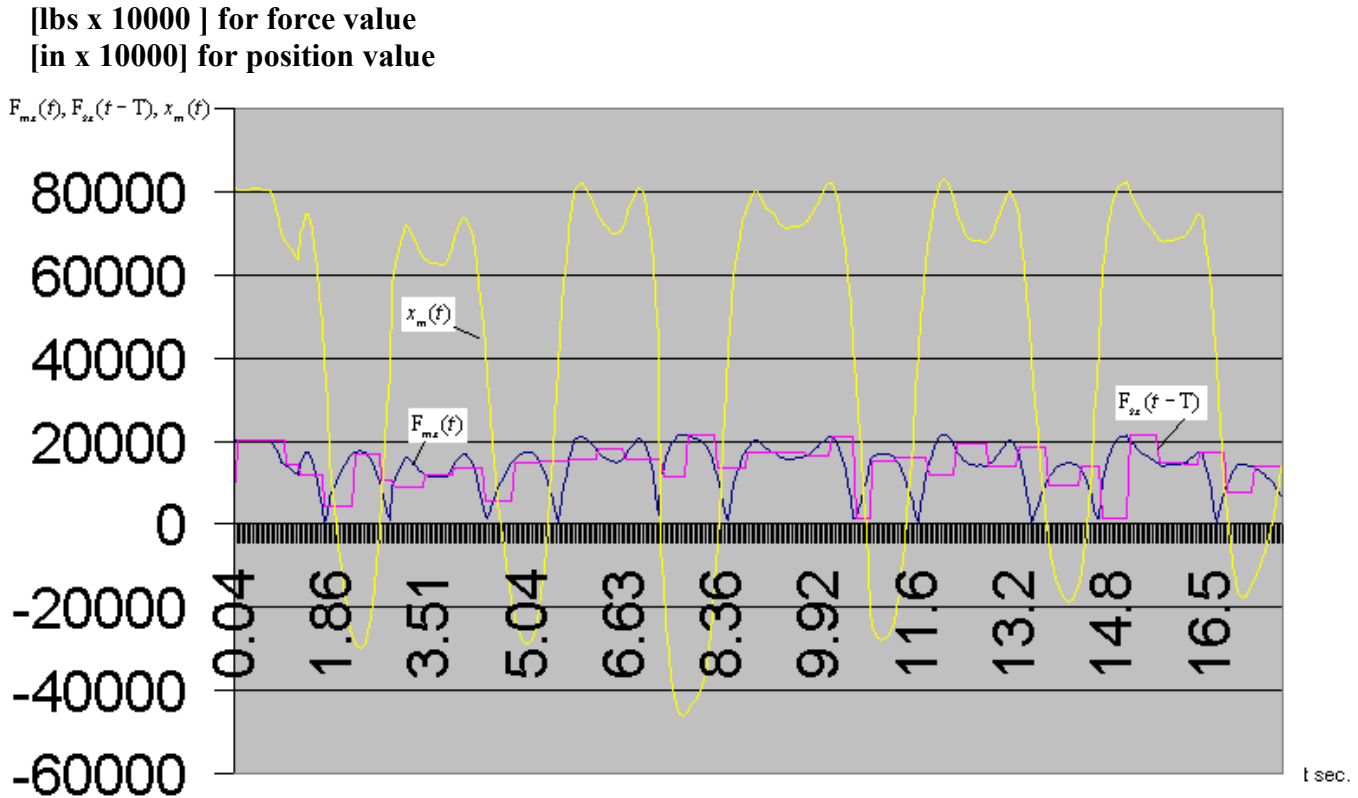


Fig. 14. The $F_{mx}(t), F_{my}(t), F_{mz}(t)$ virtual forces responses respect to the $x_m(t), y_m(t), z_m(t)$ position varying. The coefficient $k_F = 2k_F$ for force computing in algorithm represented on Fig 10 and Time Delay $T=0$ sec.

We have rigid restrictions for torque, positions, velocities and forces in this situation. If the torque, the velocity, and position are outside of the “force-reflected tunnel” supplied by the user, the Phantom controller will not apply any additional force. Any additional torque will be not applied until the velocity and positions return within the set limits.

- ***The Phantom controller scales the torque , the positions , velocities and forces*** to keep it within range (the virtual force is calculated from the values of the positions and velocities):

$$\begin{aligned}
& \text{from } T_m[0] = -MAX_Tm \text{ to } T_m[0] = MAX_Tm \\
& \text{from } T_m[1] = -MAX_Tm \text{ to } T_m[1] = MAX_Tm; \\
& \text{from } T_m[2] = -MAX_Tm \text{ to } T_m[2] = MAX_Tm; \\
x(t) &= \begin{cases} \text{for } T_m[0] < -MAX_Tm \text{ or } T_m[0] > MAX_Tm \\ \end{cases} & (23) \\
y(t) &= \begin{cases} \text{for } T_m[2] < -MAX_Tm \text{ or } T_m[2] > MAX_Tm \\ \end{cases} \\
z(t) &= \begin{cases} \text{for } T_m[1] < -MAX_Tm \text{ or } T_m[1] > MAX_Tm \\ \end{cases} \\
\dot{x}(t) &= \begin{cases} \text{for } T_m[0] < -MAX_Tm \text{ or } T_m[0] > MAX_Tm \\ \end{cases} \\
\dot{y}(t) &= \begin{cases} \text{for } T_m[2] < -MAX_Tm \text{ or } T_m[2] > MAX_Tm \\ \end{cases} \\
\dot{z}(t) &= \begin{cases} \text{for } T_m[1] < -MAX_Tm \text{ or } T_m[1] > MAX_Tm \\ \end{cases}
\end{aligned}$$

We have the rigid restrictions for torque and “absolutely zero” values of positions, velocities and forces for these restrictions outside the “force-reflected tunnel” .

[lbs x 1000] for force value

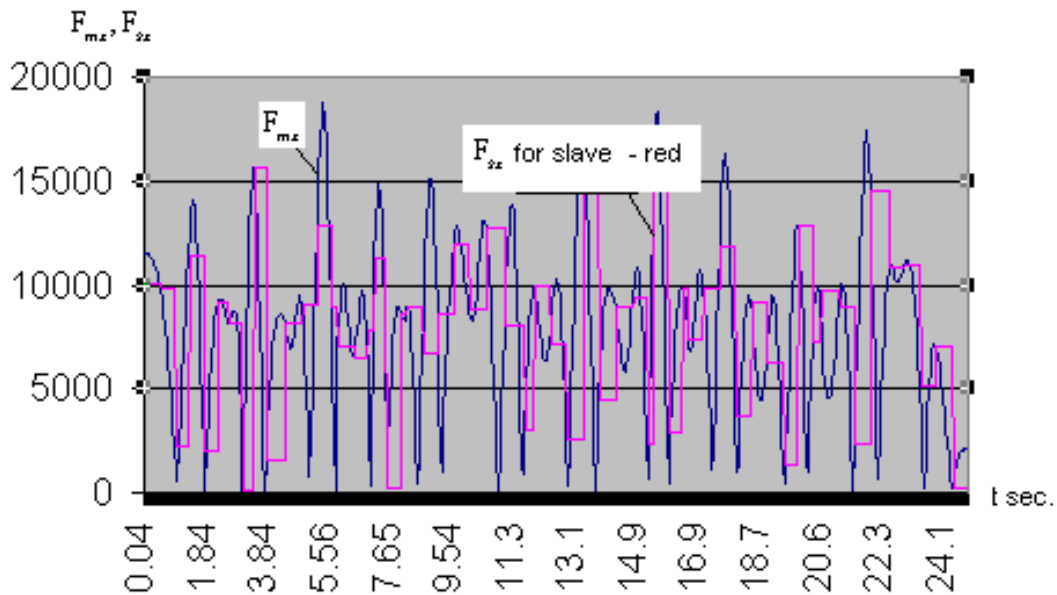


Fig. 15 The virtual force response on master and slave for very large coefficient $k_F = 3$ for force computing in algorithm represented on Fig 10 and Time Delay $T = 0$ sec. System is unstable even for Time Delay $T = 0$ sec.

- ***The Phantom controller scales the torque to keep it only within range and does not scale*** the positions, velocities, and forces to keep it within any range (the virtual force is calculated from the values of the positions and velocities):

$$\begin{aligned}
 &\text{from } T_m[0] = -MAX_Tm \text{ to } T_m[0] = MAX_Tm \\
 &\text{from } T_m[1] = -MAX_Tm \text{ to } T_m[1] = MAX_Tm; \\
 &\text{from } T_m[2] = -MAX_Tm \text{ to } T_m[2] = MAX_Tm; \tag{24}
 \end{aligned}$$

$x(t)$ has not any restrictions

$y(t)$ has not any restrictions

$z(t)$ has not any restrictions

$\dot{x}(t)$ has not any restrictions

$\dot{y}(t)$ has not any restrictions

$\dot{z}(t)$ has not any restrictions

We have the “absolutely rigid” restriction on the torque and “absolutely soft” characteristics on the positions, the velocities and the reflected forces for master robot as well as for the slave robot.

- ***The Phantom controller does not scale any values of torque, positions, velocities and forces***

to keep it within any range (the virtual force is calculated from the values of the positions and velocities). This is system with “absolutely soft” characteristics that works in free space and has not any restrictions for control parameters as for the master and for the slave. Considering the instructions for the PHANToM, we must provide for absolute torque limits at all times [Installation & Tech. Manual 1995, Massie 1994].

[lbs x 10000] for force value

[in x 10000] for position value

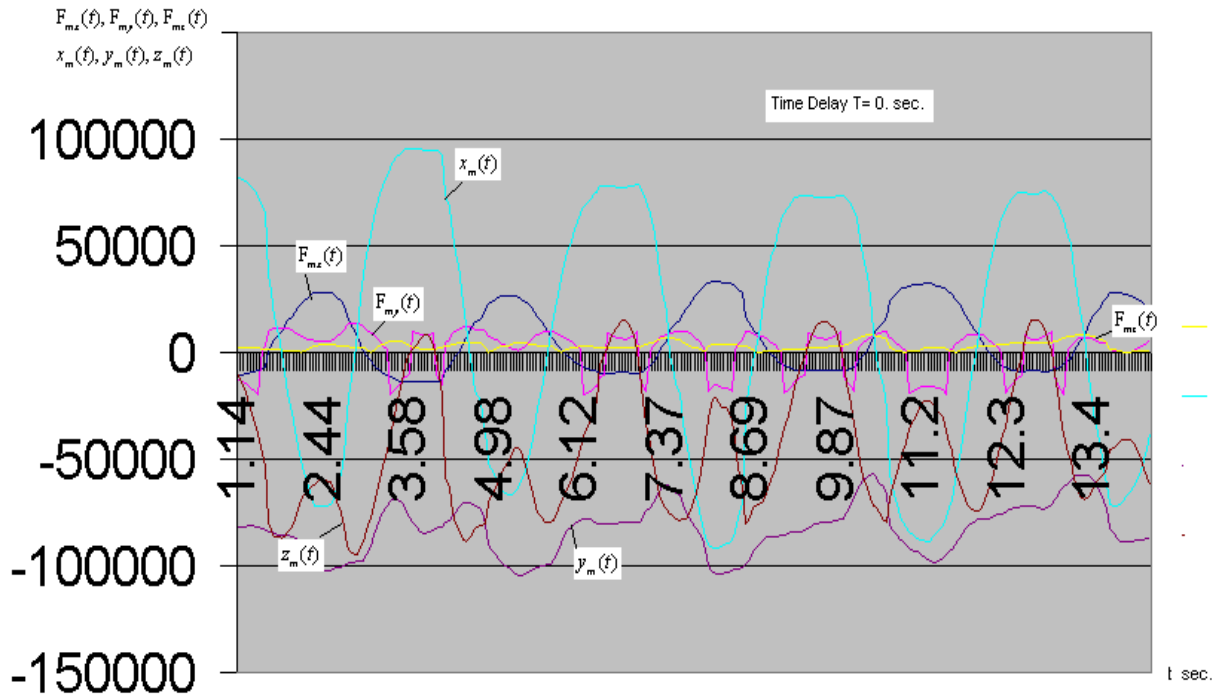


Fig. 16 The $F_{mx}(t), F_{my}(t), F_{mz}(t)$ forces reflected as response on positions $x_m(t), y_m(t), z_m(t)$. KF= -1 for force computing, Time Delay T=0 sec.

Thus this situation can be in practice only if the values of torques, positions, velocities and forces are kept small and they can not be at maximum values or limits (or reach the limits in rare cases). We can consider for this 3-dimensional scene the torque that the transpose of the Jacobian multiplies forces [Installation & Techn. Manual 1995, T. Massie 1994]. The Phantom algorithm converts *the* F_{mx}, F_{my}, F_{mz} force vectors supplied by the user, by the program, internally as responses on the position and to the torques:

$$\begin{aligned}
 \tau_{_x} &= \left({}^1_{11}F_{mx} + {}^1_{12}F_{my} + {}^1_{13}F_{mz} \right) * gradual * k \\
 \tau_{_y} &= \left(F_{mx} + {}^1_{12}F_{my} + {}^1_{13}F_{mz} \right) * gradual * k \\
 \tau_{_z} &= \left({}^1_{11}F_{mx} + {}^1_{12}F_{my} + {}^1_{13}F_{mz} \right) * gradual * k
 \end{aligned}
 \tag{25}$$

or the source code for the Visual C++ program has been developed for integrating in IPVTS in the following (this routine based on equations from Massie's publications) [Installation & Techn. Manual 1995, T. Massie 1994]. Thus we can provide for absolute torque (current) limits.

Fig. 19 displays the spatial representation of virtual torque values $Tm[0]$, $Tm[1]$, $Tm[2]$ on a visual scene generated our program .

Note that Massie and other scientists, in their publications, did not consider the inertia component and the force F_{mz} for determining the torque $\tau_{.x} = Tm [0]$. Thus we will extend the dynamic model which defines the inertia characteristics. After finding the parameters of the visual model, we also used the following algorithm and parametric functions described by T. Massie and others [Installation & Techn. Manual 1995, T. Massie 1994] .

It is important consider for the PHANToM a virtual model involving the principle of multiple springs acting together which specify the forces, general torque and all the torques with respect to all the 3 axes. In our model the principle of multiple springs is based on a non-linear model involving Massie's algorithm to control the correct position of the end-effector during the reflected force feedback loop cycles.

Modeling for the rotation conceptions of kinematics and dynamics of PHANToM robots is based on the following equations:

$$\begin{aligned} \tau_{.x} = -\tau_{.ix} &= -J_e [\theta_{.x}^{\ddot{}}(t) - \theta_{.x}^{\dot{}}(t)] - B_e [\theta_{.mx}^{\dot{}}(t) - \theta_{.x}^{\dot{}}(t)] - K_e[\theta_{.x}(t) - \theta_{.x}(t)] \\ \tau_{.y} = -\tau_{.iy} &= -J_e [\theta_{.y}^{\dot{}}(t) - \theta_{.y}^{\dot{}}(t)] - B_e [\theta_{.my}^{\dot{}}(t) - \theta_{.y}^{\dot{}}(t)] - K_e[\theta_{.y}(t) - \theta_{.y}(t)] \quad (26) \\ \tau_{.z} = -\tau_{.iz} &= -J_e [\theta_{.z}^{\dot{}}(t) - \theta_{.z}^{\dot{}}(t)] - B_e [\theta_{.z}^{\dot{}}(t) - \theta_{.z}^{\dot{}}(t)] - K_e[\theta_{.z}(t) - \theta_{.z}(t)] \end{aligned}$$

```

//Computing torque based on multiple spring idea for all 3 axes
// Torque transpose of Jacobian times forces:

Tm[0]=(long) (gradual*k1*(-s1 clg * forcex + c1 * clg * forcey));

Tm[2] = (long) (gradual * k2 *(-c1 * L2 * s23 * forcex - s1 * L2 * s23 * forcey +
L2 * c23 * forcez) );

Tm[1] = -Tm[2] + (long) (gradual * k2 *(-c1 * slg * forcex - s1 * slg *
forcey + clg * forcez));

```



```

// check for saturating motor

if (Tm[0]> MAX_Tm) Tm[0]= MAX_Tm;
if (Tm[0]<-MAX_Tm) Tm[0]=-MAX_Tm;

if (Tm[1]> MAX_Tm) Tm[1]= MAX_Tm;
if (Tm[1]<-MAX_Tm) Tm[1]=-MAX_Tm;

if (Tm[2]> MAX_Tm) Tm[2]= MAX_Tm;
if (Tm[2]<-MAX_Tm) Tm[2]=-MAX_Tm;

```



```

// Write to torque motors...

torque(0,Base_Addr, (long) (Tm[0] + 32768.0));
torque(1,Base_Addr, (long) (Tm[2] + 32768.0));
torque(2,Base_Addr, (long) (Tm[1] + 32768.0));

```



Fig 17 The algorithm for computing torque and checking for a saturating torque limit on the motor based on T. Massie's Control Program [Installation & Techn. Manual 1995, Massie 1994]

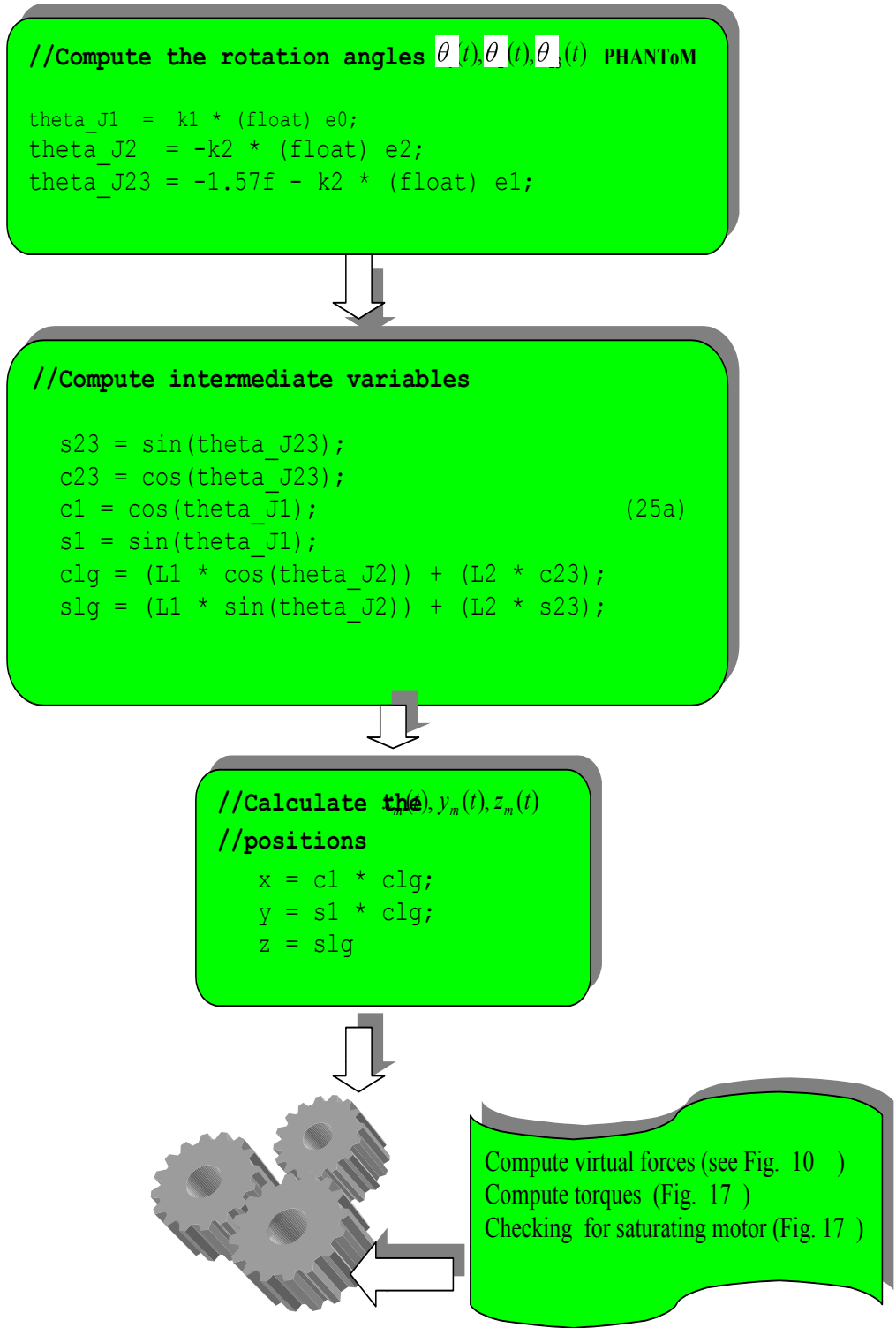


Fig. 18 The Massie's Algorithm [Installation & Technical Manual ,1999] that is used as Basic Control Function Prototype in the virtual model of IPVTS.

Using these control laws, we can determine the optimal characteristics between the operating force and the external force. For this, a desired impedance characteristic is specified for the master and slave such as $M_e = M_m = M_s$, $B_e = B_m = B_s$ and $K_e = K_m = K_s$ or the alternative set of the desired parameters such as $k_e M_e = k_m M_m = k_s M_s$, $k_{be} B_e = k_{bm} B_m = k_{bs} B_s$ and $k_{ke} K_e = k_{km} K_m = k_{ks} K_s$.

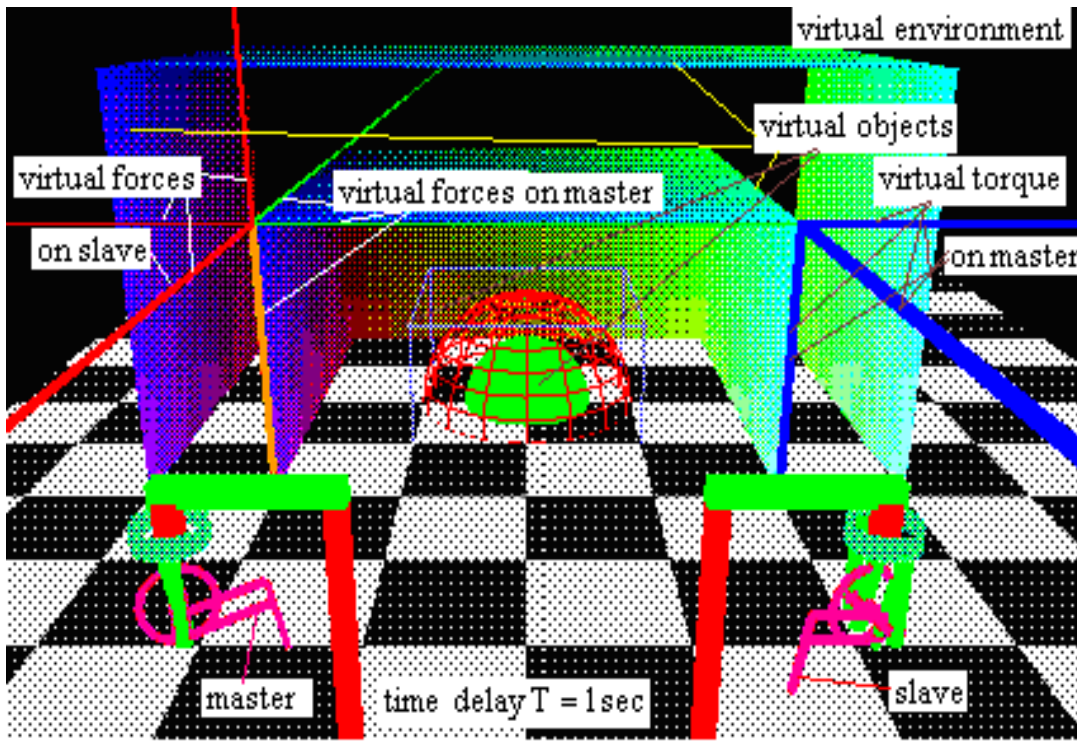


Fig. 19 The virtual scene that represents the teleoperation process using the PHANTOM master controlled by the hand of the operator and the remote virtual slave which gets the control signal from the master with Time Delay $T=1.0$ sec. This visual scene also demonstrates reflected virtual forces on the master and on the slave (red color), which virtual torques on the master (blue color). It is shown that the slave position has a large position error which is caused by the delay time.

We can also use non- equal parameters in these laws such as $k_e M_e \neq k_m M_m \neq k_s M_s$, $k_{be} B_e \neq k_{bm} B_m \neq k_{bs} B_s$ and $k_{ke} K_e \neq k_{km} K_m \neq k_{ks} K_s$. By combining these parameters we can create the control system with desired characteristics

of the operating utilizing external forces with position. But for non-optimal values of these non - equal parameters can cause instability as it is shown in Fig. 30, Fig. 37 and Fig 38. In general, it is a multiple optimization task which can be solved via a heuristic algorithm of the IPVTS.

In the work [Buttolo 1994], one of the control laws must be selected involving a negative feedback term proportion to the controller depending on the values of the desired parameters $k_e M_e \neq k_m M_m \neq k_s M_s$, $k_{be} B_e \neq k_{bm} B_m \neq k_{bs} B_s$ and $k_{ke} K_e \neq k_{km} K_m \neq k_{ks} K_s$ which can increase the stability of system. Based on these control laws we can note that as positive and negative values of signs (“+” and “-”) occur in the balance between the elements of equations (18) or (26) occur because, in the general case, this sign depends on the values of parameters $[\theta_{\dots x}^{\ddot{}}(t) - \theta_{\dots x}^{\dot{}}(t)]$, $[\theta_{\dots y}^{\dot{}}(t) - \theta_{\dots y}^{\dot{}}(t)]$, $[\theta_{\dots y}(t) - \theta_{\dots y}(t)]$, $k_e M_e \neq k_m M_m \neq k_s M_s$, $k_{be} B_e \neq k_{bm} B_m \neq k_{bs} B_s$ and $k_{ke} K_e \neq k_{km} K_m \neq k_{ks} K_s$ which can be changed in this system considering alternative control laws and feedbacks.

For example, in addition to the above represented equations we can obtain the following combinations of equations with alternative positive and negative feedbacks:

$$\begin{aligned}
 F_{sx} &= -F_{mx} = -M_e [x_m^{\ddot{}}(t) - x_s^{\ddot{}}(t)] + B_e [x_m^{\dot{}}(t) - x_s^{\dot{}}(t)] - K_e [x_m(t) - x_s(t)] \\
 F_{sy} &= -F_{my} = -M_e [y_m^{\dot{}}(t) - y_s^{\dot{}}(t)] + B_e [y_m^{\dot{}}(t) - y_s^{\dot{}}(t)] - K_e [y_m(t) - y_s(t)] \quad (27) \\
 F_{sz} &= -F_{mz} = -M_e [z_m^{\dot{}}(t) - z_s^{\dot{}}(t)] + B_e [z_m^{\dot{}}(t) - z_s^{\dot{}}(t)] - K_e [z_m(t) - z_s(t)]
 \end{aligned}$$

$$\begin{aligned}
 F_{sx} &= -F_{mx} = -M_e [x_m^{\ddot{}}(t) - x_s^{\ddot{}}(t)] - B_e [x_m^{\dot{}}(t) - x_s^{\dot{}}(t)] + K_e [x_m(t) - x_s(t)] \\
 F_{sy} &= -F_{my} = -M_e [y_m^{\dot{}}(t) - y_s^{\dot{}}(t)] - B_e [y_m^{\dot{}}(t) - y_s^{\dot{}}(t)] + K_e [y_m(t) - y_s(t)] \quad (28) \\
 F_{sz} &= -F_{mz} = -M_e [z_m^{\dot{}}(t) - z_s^{\dot{}}(t)] - B_e [z_m^{\dot{}}(t) - z_s^{\dot{}}(t)] + K_e [z_m(t) - z_s(t)]
 \end{aligned}$$

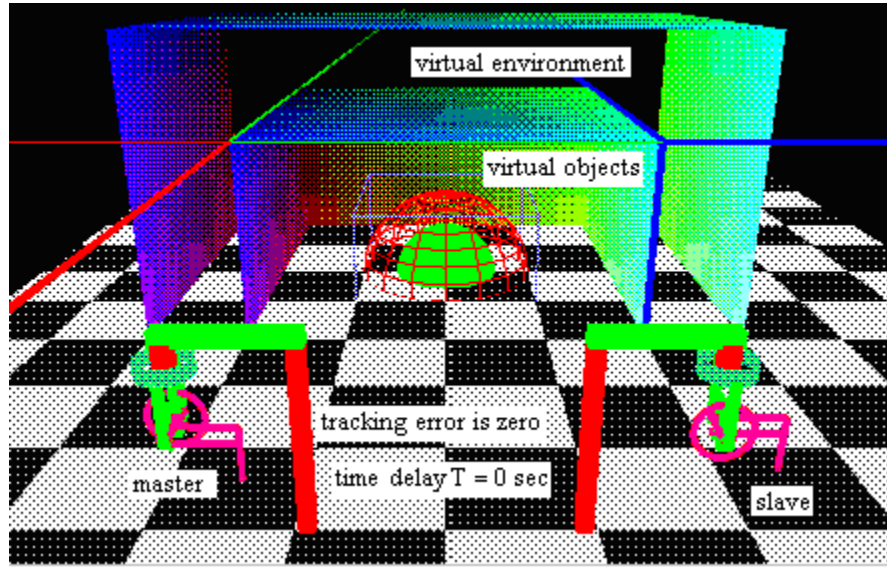


Fig. 20 The virtual scene that represents the teleoperation process using the PHANToM master controlled by the hand of operator to the remote virtual slave which obtains the control signal from the master without Time Delay $T=0$ sec. It is shown that the slave position has a small position error that is explained because the delay time $T=0$.

$$\begin{aligned}
 F_{sx} = -F_{mx} &= +M_e [x_m''(t) - x_s''(t)] - B_e [x_m'(t) - x_s'(t)] + K_e [x_m(t) - x_s(t)] \\
 F_{sy} = -F_{my} &= +M_e [y_m'(t) - y_s'(t)] - B_e [y_m(t) - y_s(t)] + K_e [y_m(t) - y_s(t)] \quad (29) \\
 F_{sz} = -F_{mz} &= +M_e [z_m'(t) - z_s'(t)] - B_e [z_m(t) - z_s(t)] + K_e [z_m(t) - z_s(t)]
 \end{aligned}$$

Based on the above described methodology, the IPVTS system was developed. This high-level integrated system is based on a general methodology and algorithms represented in Fig. 1, Fig. 2, algorithms of PHANToM robot on Fig. 8, Fig. 9, Fig. 17 and Fig. 18, equations (1).....(55). This visual environment system is based on a computer program using MS Visual C++ with MFC, OpenGL high-level language for graphic environment based on the work [Fosner 1997], driver programs for PHANToM and MS Excel for graphic charts integrated with the IPVTS.

This virtual scene for research consists of a PHANToM master robot represented both the virtual model and a physical real device controlled by the hand of an operator, virtual

slave robot based on a model and virtual objects. All events and actions in the virtual environment were controlled via the operator of the PHANTOM device and computer mouse as additional control devices.

This virtual environment allows us to solve the tasks related with animation of virtual objects, interactions of objects with reflected forces, control tasks for virtual objects in Real-Time and simulation of time delay from 1...10 sec. and more.

Conclusion: It is clearly shown that in general virtual models the control strategies must select the optimal structure and parameters based on rules for a heuristic search in a high-level discrete dynamic model.

USING IPVTS FOR TELEOPERATION BASED ON TIME DELAY COMMUNICATIONS

Niemeyer and Slotine [Niemeyer and Slotine 1991] demonstrated that stability of bilateral teleoperation in the presence of irregular time delay can be preserved through the systematic use of a wave-variable formulation.

As it shown in Niemeyer's work [Niemeyer 1996], the time delay can be considered in the following equations for the slave robot and the master robot respectively :

$$F_{sx} = B_e [\dot{x}_m(t-T) - \dot{x}_s(t)] + K_e [x_m(t-T) - x_s(t)] \quad (30)$$

$$F_{mx} = - B_e [\dot{x}_m(t) - \dot{x}_s(t-T)] - K_e [x_m(t) - x_s(t-T)] \quad (31)$$

For our case we have the following equations based on bilateral control system with negative feedback force $-F_{sx}$:

$$\begin{aligned} F_{mx} = -F_{sx} &= -M_e [\ddot{x}_m(t) - \ddot{x}_s(t-T)] - B_e [\dot{x}_m(t) - \dot{x}_s(t-T)] - K_e [x_m(t) - x_s(t-T)] \\ F_{my} = -F_{sy} &= -M_e [\ddot{y}_m(t) - \ddot{y}_s(t-T)] - B_e [\dot{y}_m(t) - \dot{y}_s(t-T)] - K_e [y_m(t) - y_s(t-T)] \\ F_{mz} = -F_{sz} &= -M_e [\ddot{z}_m(t) - \ddot{z}_s(t-T)] - B_e [\dot{z}_m(t) - \dot{z}_s(t-T)] - K_e [z_m(t) - z_s(t-T)] \end{aligned} \quad (32)$$

For the case based on positive feedback we have:

$$\begin{aligned}
 F_{mx} = -F_{sx} &= M_e [x_m''(t) - x_s''(t-T)] + B_e [\dot{x}_m(t) - \dot{x}_s(t-T)] + K_e [x_m(t) - x_s(t-T)] \\
 F_{my} = -F_{sy} &= M_e [y_m''(t) - y_s''(t-T)] + B_e [\dot{y}_m(t) - \dot{y}_s(t-T)] + K_e [y_m(t) - y_s(t-T)] \quad (33) \\
 F_{mz} = -F_{sz} &= M_e [z_m''(t) - z_s''(t-T)] + B_e [\dot{z}_m(t) - \dot{z}_s(t-T)] + K_e [z_m(t) - z_s(t-T)]
 \end{aligned}$$

It is very important to study the modeling for the time delay $T(V) = T$ into the control system and the force feedback response from the slave robot. The experiment to be discussed in next section will be conducted using the PHANTOM and the virtual modeling technique based on the IPVTS with different time delays: $T = 0$ s, $T = 0.5$ s, $T = 1.0$ s, $T = 1.5$ s. The goal of these experiments is to compare the stability of the control system with and without the time delay.

It is needed to repeat this experiment with the force reflection and the force feedback. Here the force feedback response can be defined as a feedback function (as bias distance into virtual model) $BD_i(V) = f [F = P_i(V)]$ or $BD_i(V) = f [F = PB_i(V)]$ where $F = P_i(V)$ and $F = PB_i(V)$ can be represented by “virtual force” into the visual model. Alternative controllers can be used such that a computer mouse, 3D-Audio control system, MEMS, or computer control software. The goal of these experiments is to compare the stability of the control system with the time delay and without the time delay using the different types of sensory information and when the cues work together with the intelligent m-dimensional parametric control. It allows the use of the advantage of the intelligent m-dimensional parametric control based on IPVTS.

Examples of Time Delay-Induced Instability

This research is demonstrated in a variety of experiments based on virtual reality modeling, and explained below using MS Excel Graphical Charts related with electronic tables based on output files from the IPVTS.

Both manipulators are connected to the Model-Based Adaptive Control System with negative Feedback from the virtual PHANToM slave that reflects and transmits the force on a real PHANToM master robot. Fig. 21 portrays the curves of positions $x(t), y(t), z(t)$ for the master and slave $x(t - \tau), y(t - \tau), z(t - \tau)$ with time delay $T = 1.0$ in a Cartesian coordinates system. Analysis of the graphical results provides the following conclusions:

- The time delay causes the tracking errors on every axis that can be computed as

$$\begin{aligned} \Delta_x &= x_m(t) - x_s(t - \tau) \\ \Delta_y &= y_m(t) - y_s(t - \tau) \\ \Delta_z &= z_m(t) - z_s(t - \tau) \end{aligned} \tag{34}$$

[in x 10000] for position value

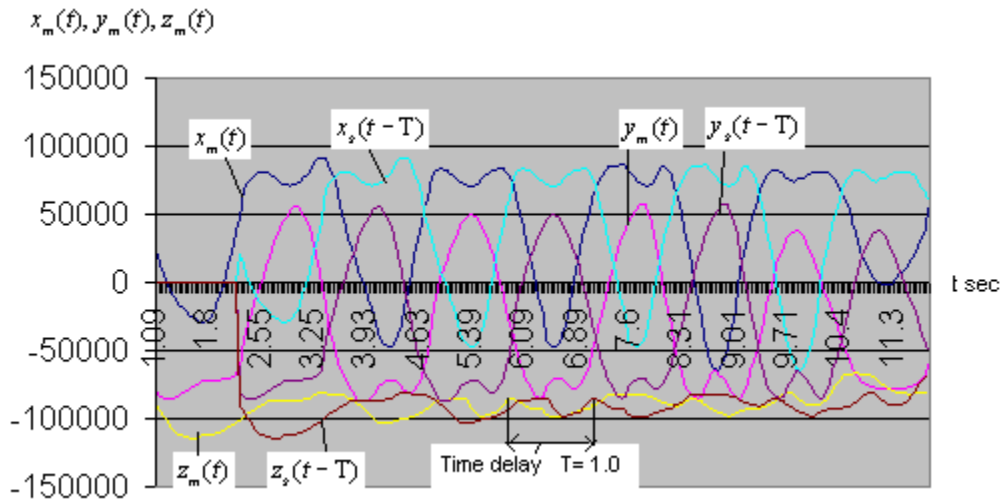


Fig. 21 The modeled experimental results that show the $x(t), y(t), z(t)$ positions of PHANToM's effector on master and on slave $x(t - \tau), y(t - \tau), z(t - \tau)$ with time delay $T = 1.0$.

As it is shown on Fig. 21 this error can get very large (see Δ) and defines the value close to the maximum magnitude of $x(t)$ position variations . This fact can be the reason for a critical event for control in real-time system

- The position $x(t)$ is associated with the $y(t)$ position and the $z(t)$ position at the same time both on in the master and on the slave. If it excludes the random variations of $x(t)$, $y(t)$, $z(t)$ in the control process, we can note that these positions have similar phase for a varying law. An example, Fig. 21, demonstrates the case if the $x(t)$ increases to maximum, the $y(t)$ increases up to a maximum limit in the same time. The equations (25a) point out that the computing of $x(t)$, $y(t)$, $z(t)$ is related to the c_1, s_1, c_{1g}, s_{1g} parameters which are based on the similar values of the $\theta_1(t), \theta_2(t), \theta_3(t)$ for the PHANTOM (see Fig. 18). For small time delays, an example $T= 0.1$ also causes the errors in the positions. Fig. 22 shows the curve of angles $\theta_1(t), \theta_2(t), \theta_3(t)$ for the master and $\theta_1(t-T), \theta_2(t-T), \theta_3(t-T)$ for the slave with small time delay $T=0.1$.

The data clearly show that the time delay $T = 0.1$ sec. for these angles causes similar errors for angles as in case for the $x(t)$, $y(t)$, $z(t)$ positions. It is demonstrated that these errors are small for the time delay $T=0.1$ sec. The motion for slave positions are the analogs of the master with time delay 0.1 sec.

For a large time delay, for example $T=1.0$ sec, these errors are shown on Fig. 21. In this case the error for $\theta_1(t)$ consists of a very large value:

$$\Delta_{\theta} = \theta_1(t) - \theta_1(t-T) = 2.75 \text{ rad.}$$

Rotation angles in radians

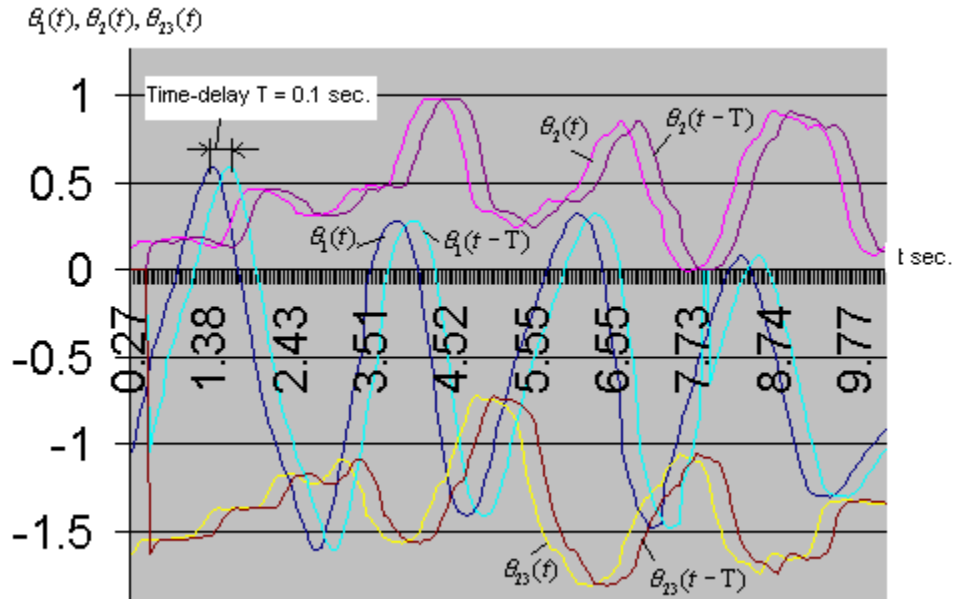


Fig. 22 Basic delayed teleoperator with a time delay $T=0.1$ sec. for the $\theta_1(t), \theta_2(t), \theta_3(t)$ rotation angles of the master and the $\theta_1(t-T), \theta_2(t-T), \theta_3(t-T)$ rotations angles of slave.

Note that the slave works in anti-phase with the master for this angle-based situation. It causes to unstable dynamics and the operator loses the control.

In next experiments, discussed below, we will study how the reflected forces on the master and slave can be associated with the end-effector's position $x_m(t)$, time delay, and the instability of system. In these tests, the operator strongly grips the PHANTOM effector by hand to prevent large motions and possible vibrations and damage to the robotic system.

The slave robot moves in free space without any absolute contact (it contains the spring or elastic contact only) but in many tests both robots (or single robot) have been tested with torque limits on the motor-encoders to prevent these devices from damage related to large reflected torques and forces as shown in Fig. 8 and Fig. 9.

Fig. 24 shows the visual representation of the force on the master $F_{mx}(t)$ for the effector's position $x_m(t)$ and $F_{sx}(t - T)$ reflected virtual force on the slave with a large time delay $T=1.0$ sec. and the input control law on the PHANTOM is based on the value $k_F = -1$ (see Fig. 9 explaining this case). The reflected torque on the slave is not restricted but the total torque on the master had the maximum limit $T_MAX = \pm 000$.

Rotation angles in radians

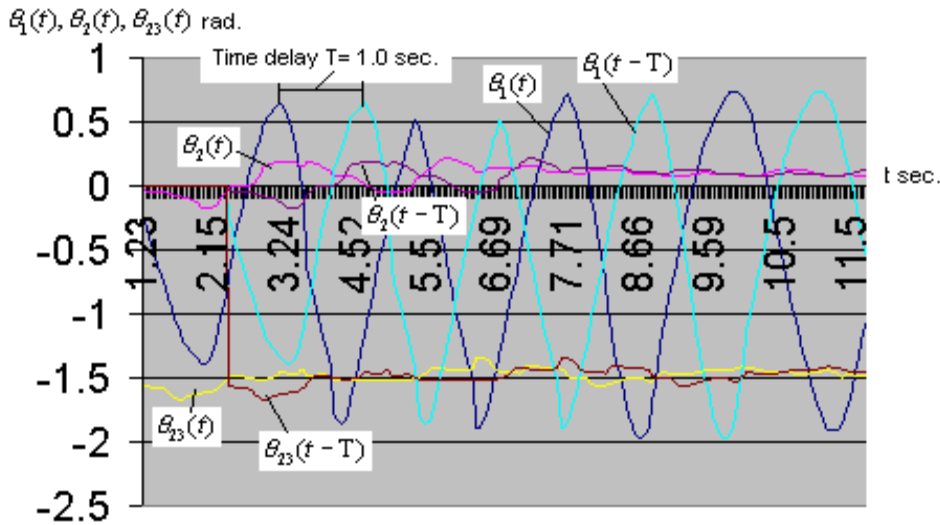


Fig. 23 Basic delayed teleoperator with a time delay $T = 1.0$ sec. for the $\theta_1(t), \theta_2(t), \theta_{23}(t)$ rotations angles of the master and the $\theta_1(t-T), \theta_2(t-T), \theta_{23}(t-T)$ rotations angles of the slave. The data clearly show the time delay $T=1$ sec. for these angles. Note that the $\theta_1(t-T), \theta_2(t-T), \theta_{23}(t-T)$ rotations angles of the slave are in opposite phase with the $\theta_1(t), \theta_2(t), \theta_{23}(t)$ rotation angles of master.

The visual real-time results in Fig. 25 and the virtual scene in Fig 26 clearly show that the reflected force on the slave $F_{sx}(t - T)$ has the time delay $T \approx 1.0$ sec with a master force $F_{mx}(t)$ and it works in anti-phase with $F_{mx}(t)$. This causes the large displacement $x_m(t)$ that operator maintains. In the processing of the control the operator feels the strongly reflected force

$F_{mx}(t)$ that increase oscillations and instability. If the operator releases the end-effector in some displacement, the oscillation, instability and displacement increase for all degrees of freedom and the system can go out of control.

In the case without time delay, when $T = 0$, we have dissimilar situations. The torque and force on the slave practically fit the torque and force on the master as illustrated in Fig. 27. The interactions of these torques compensates for the possible magnitude of oscillations and the system is stable because the total torque on the master (see Fig. 27) has a small magnitude of oscillations. The system is stable and has good dynamic characteristics. Note that for this example we have considered bilateral feedback with the “-“ minus sign appearing in equation (32) before components $K_e[x_m(t) - x_s(t - \tau)]$. We have the small input forces and torques on the master and slave that are inside of limits for $T_MAX = \pm 3000$ total torque limit (see Fig. 9 also).

Consider the interactions where the torque curves for the master and slave result in a total torque on the master for the system with a large time delay $T=1.0$ sec. (see Fig. 30). This graph clearly shows that the magnitude of the total torque is very large and it is characterized by strong and increasing oscillations. The system automatically supports this total torque inside the “torque reflection tunnel “inside the limits area $T_MAX = \pm 3000$. However, this control process causes additional instability for a large variation the of values of parameter $\Delta_\tau = T_MAX - \tau_{max}$ which modules additional oscillations,

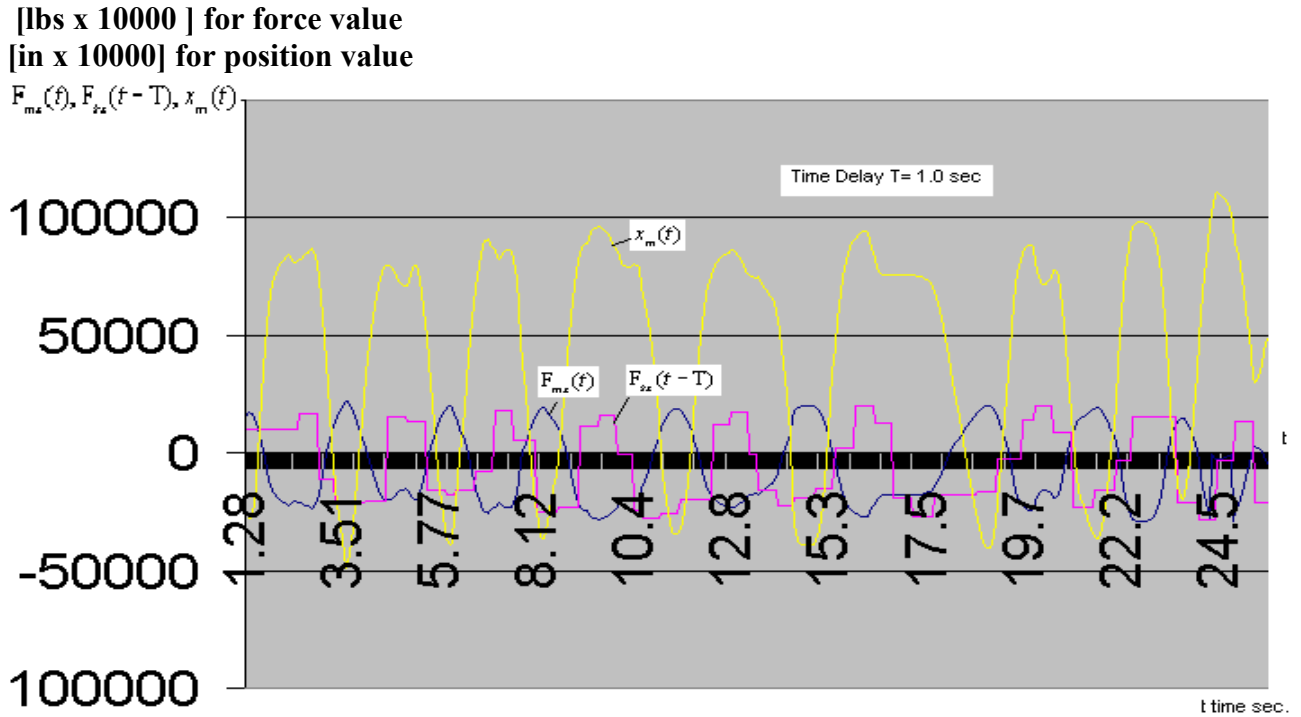


Fig. 25 Visual representation of force on master $F_{mx}(t)$ and slave $F_{sx}(t - \tau)$ and position on master - $x_m(t)$. The reflected force on master $F_{sx}(t)$ has an opposite sign to position $x_m(t)$.

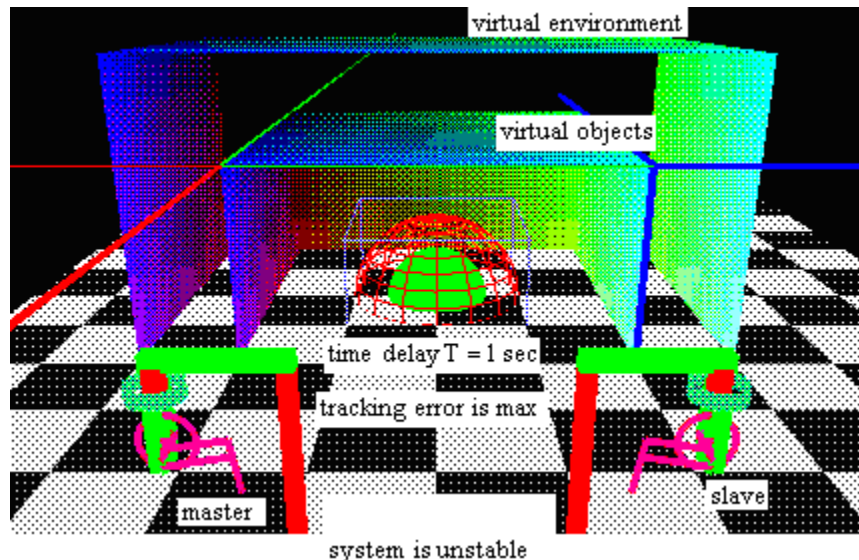


Fig. 26 The virtual scene that represents the teleoperation process using the PHANToM master controlled by hand of operator and remote virtual slave that gets the control signal from master with Time Delay $T=1.0$ sec. It is shown that the slave position has a large position error that is explained by the delay time $T=1.0$. The system is unstable.

torque value in [lbs x in x 10000]

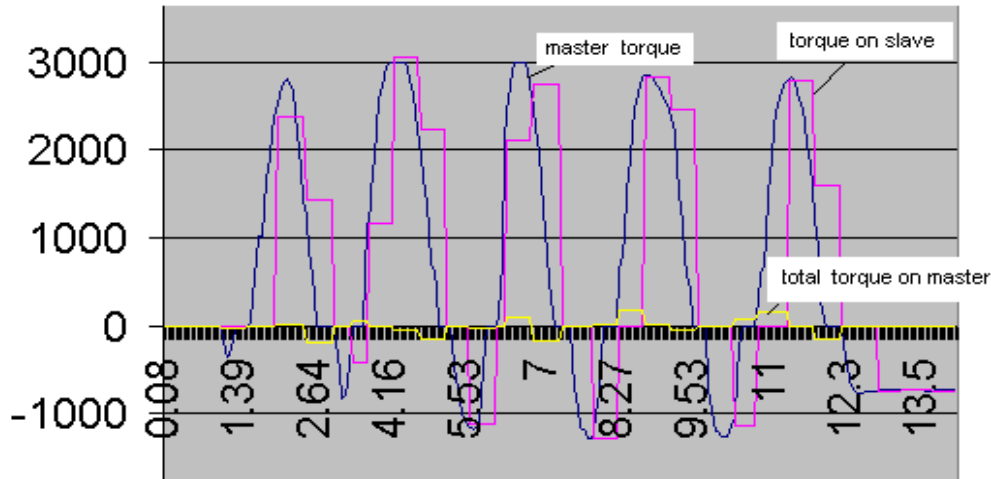


Fig. 27 Visual representation of torque curve for master – black blue line and torque curve for slave - pink (inside of interval for limits $T_MAX = \pm 3000$). The total torque on the master (yellow) has a very small interval of magnitude and system is very stable without time delay $T = 0$. Coefficient $k_F = -1$.

Torque value [lbs x in x 10000]

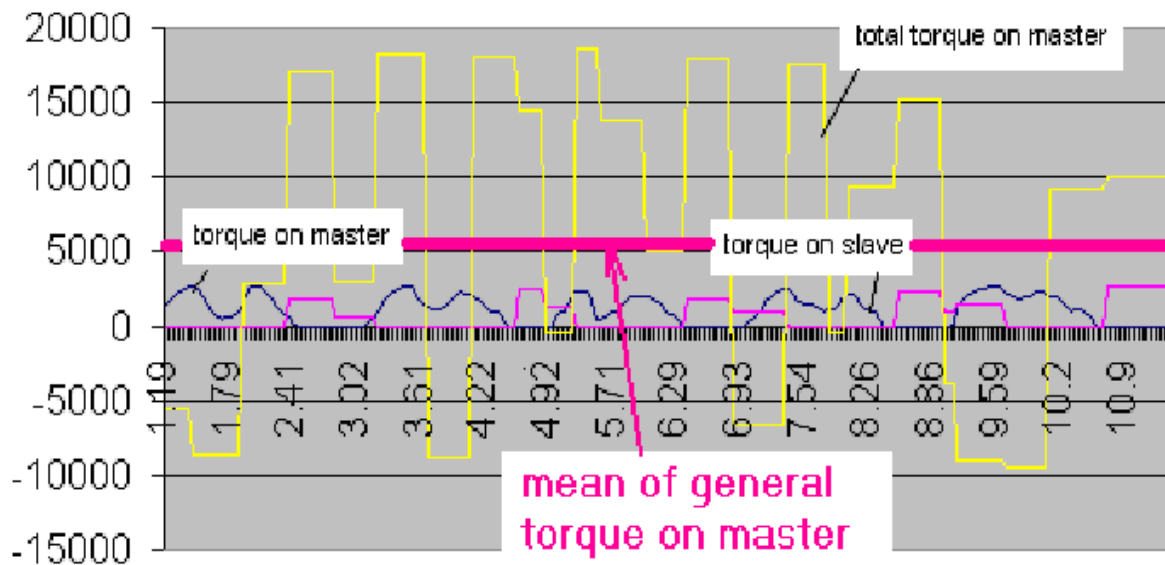


Fig. 28 Visual representation of torque curve for master – black line and torque curve for slave - pink (inside of the interval for limits $T_MAX = \pm 3000$). The total torque on the master (yellow) has a very large the interval of magnitude more than the limits $T_MAX = \pm 3000$ and the system is unstable with time delay $T = 1.0$. Coefficient $k_F = 1$, i.e. it has “dead zone” in the tunnel inside the small virtual box (see Fig. 9 for clarity).

Torque value [lbs x in x 10000]

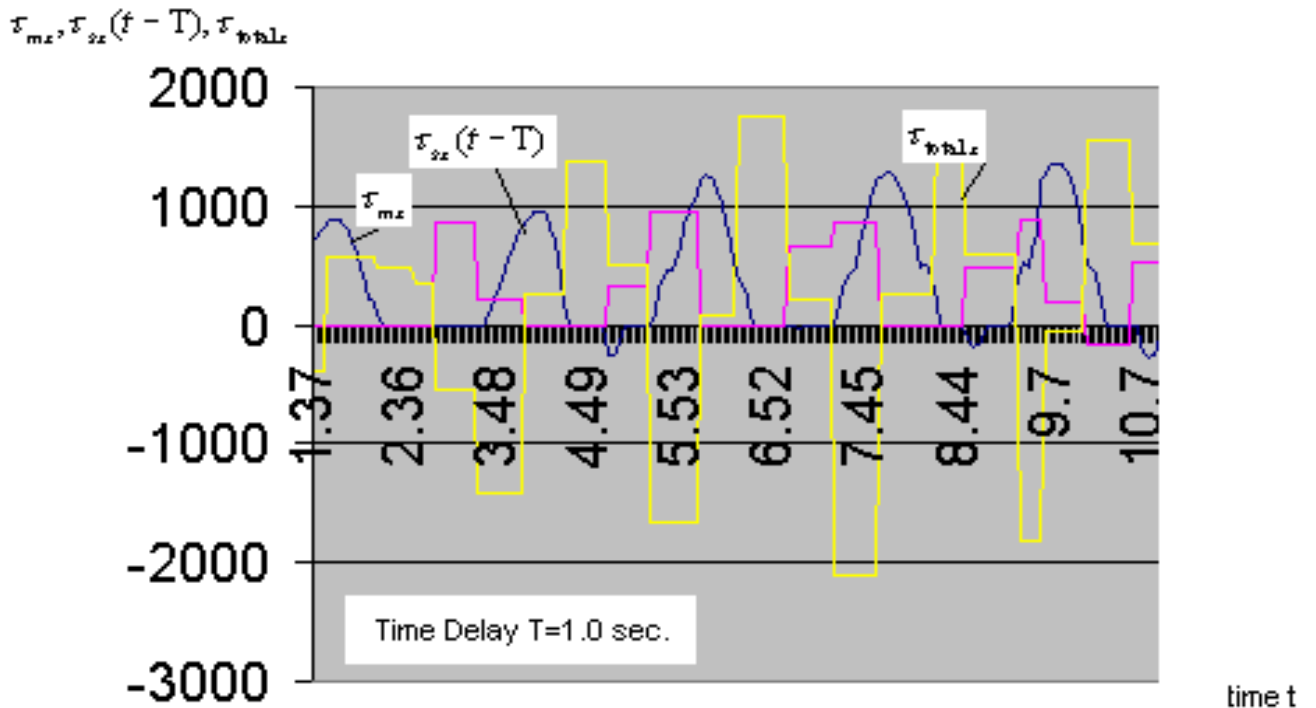


Fig. 29 The graphical representation the situation in which the system is unstable despite a small value of torque on the master τ_{ix} and on the slave $\tau_{ix}(t - T)$ with time delay

$T=0.1$ sec. The coefficient $k_F = -0.00001$

with these displacements, the operator is not able to stop increasing oscillations, forces and torques even when he prevents large motions. This instability is more clearly visible for all three degrees of freedom of the PHANToM where increasing oscillations occur of torque both on the master - $\tau_{ix}, \tau_{iy}, \tau_{iz}$ and slave - $\tau_{ix}(t-T), \tau_{iy}(t-T), \tau_{iz}(t-T)$ (see Fig. 30) as a result of the superposition for torques.

The operator cannot control the master effectively because the increasing oscillations causes unstable dynamic and can damage the device. An experimental example based on visual modeling in Fig. 30 illustrates the instability of the control of PHANToM system when torque limit MAX_Tm has been exceeded for all x, y, z axes.

Experimental Results for Analysis of Stability Using the Velocity and Position Characteristics with Time Delay

To better understand the effects of increasing instability and oscillations with time delay and basic teleoperation, consider the interaction of position $x_m(t)$ on master with negative feedback position on the slave - $x_m(t - \tau)$. Here the total value of the position on the master is computed according to equation (32) as:

$$totalx_m(t) = x_m(t) - x_m(t - \tau) \quad (35)$$

In this equation, the difference in the right side $x_m(t) - x_m(t - \tau)$ increases with greater difference as the time delay increases (from $T=0$ to $T=1.0$). An example for time delay $T=0$ (See Fig. 31) the total position of the master modules the oscillations with magnitude $totalx_m(t) = x_m(t) - x_m(t - \tau) \Rightarrow 0$. Thus this small component in equation (32), in this case, allows a stable system to be created. Similar results have been noticed for each degree of freedom independently as in this test as in all other tests discussed below.

In the situation with time delay $T=0.2$, the negative feedback position on the slave $x_m(t - \tau)$ cannot compensate for the total difference on the master - $totalx_m(t) = x_m(t) - x_m(t - \tau)$ completely and the system will have bigger oscillations for the master position. Observe the curve in yellow on Fig. 32. The potential energy of the spring described in equations (19), (21) is increased which can be the cause

Torque value [lbs x in x 10000]

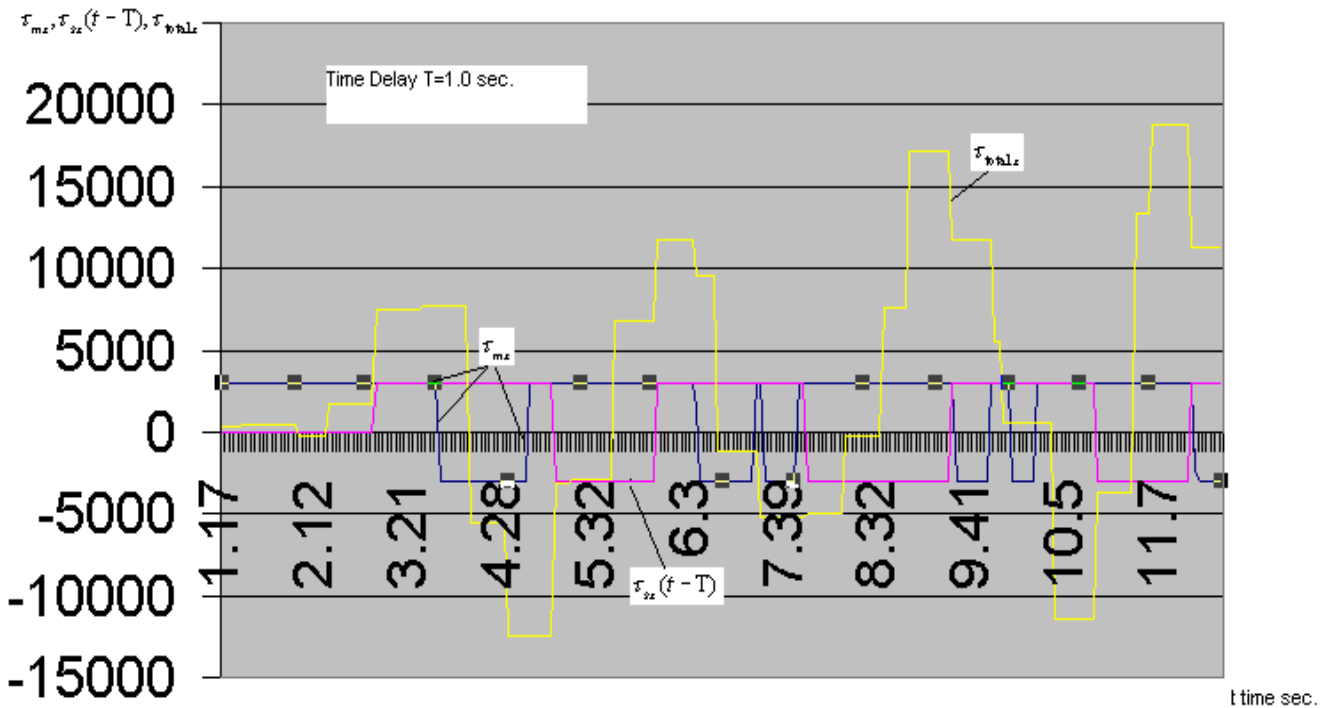


Fig 30 Visual representation of torque curve for master – black line and torque curve for slave - pink (inside of interval for limits $T_MAX = \pm 3000$). The total torque on the master (yellow) has a very large interval of magnitude more than the limits $T_MAX = \pm 3000$ and the system is unstable with time delay $T = 1.0$. The coefficient $k_F = -1$. i.e., it is “force reflection tunnel” inside small virtual box (see Fig. 9 for clarity).

of instability if the other characteristics, an example damper, cannot compensate this oscillation sufficiently,

Real –Time Modeling results are displayed in Fig, 33 for a delay $T=1$ sec. which clearly shows the violation of stability with increasing oscillations for total position where $totalx_m(t) = x_m(t) - x_m(t - \cdot)$. In this case the position $x_m(t - \cdot)$ is added with position - $x_m(t)$ for the master despite the negative feedback signal in equation (32). The slave position thus tracks the master position with a large error of phase (even in the anti-phase situation). The total amplitude $totalx_m(t)$ is increased to a large catastrophic amount, in this case, and the operator cannot prevent the oscillations.

Torque value [lbs x in x 10000]

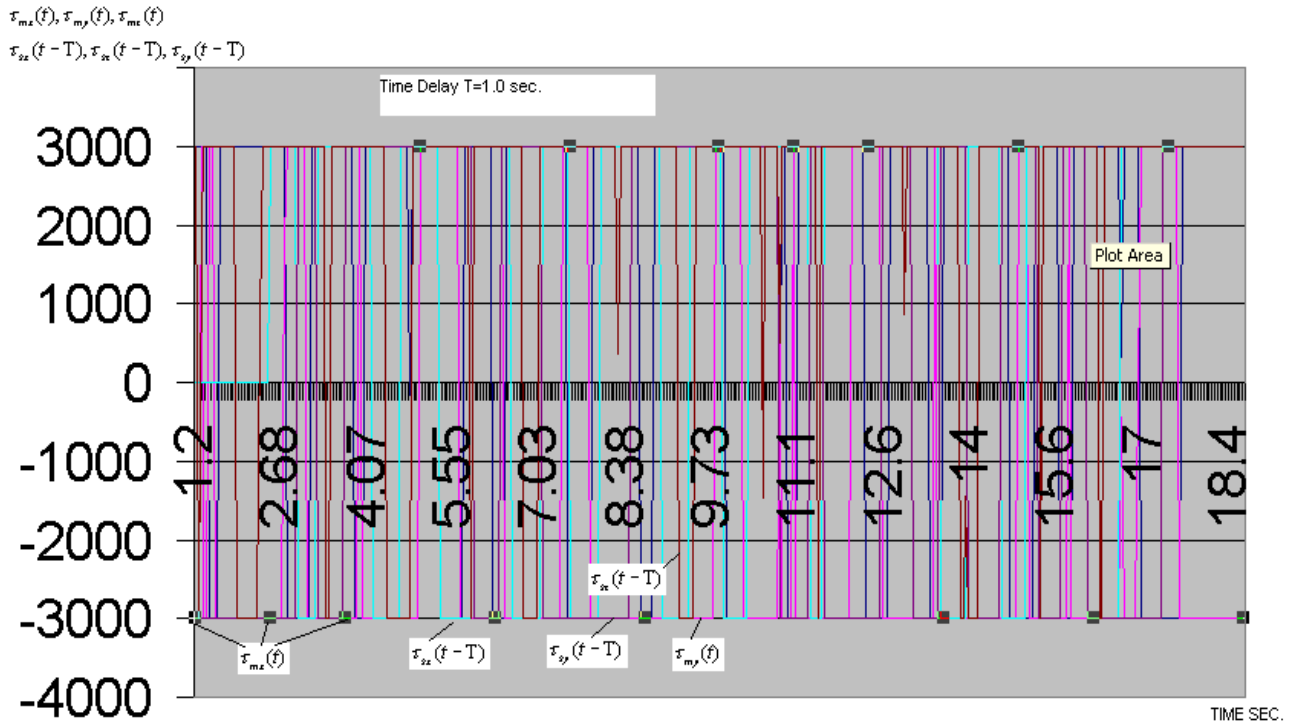


Fig. 30a Torque for all three axes with time delay $T=1.0$ sec.. System is very unstable.

The limits for torque for all axes is $T_MAX = \pm 3000$

Accordingly, the time delay communications element increases the global displacement affecting the potential energy. The instability is caused by this growing amount [Niemeyer 1996] of potential energy.

Similar results were noted for a velocity test in a teleoperation simulation for master velocity $\dot{\mathbf{x}}_m(t)$ and the slave velocity $\dot{\mathbf{x}}_m(t - \tau)$. For example, Fig. 34 shows that total velocity is approximately zero because the slave velocity $\dot{\mathbf{x}}_m(t - \tau)$ is deducted from master velocity $\dot{\mathbf{x}}_m(t)$ (both values are computed without time delay).

The system is stable in this case.

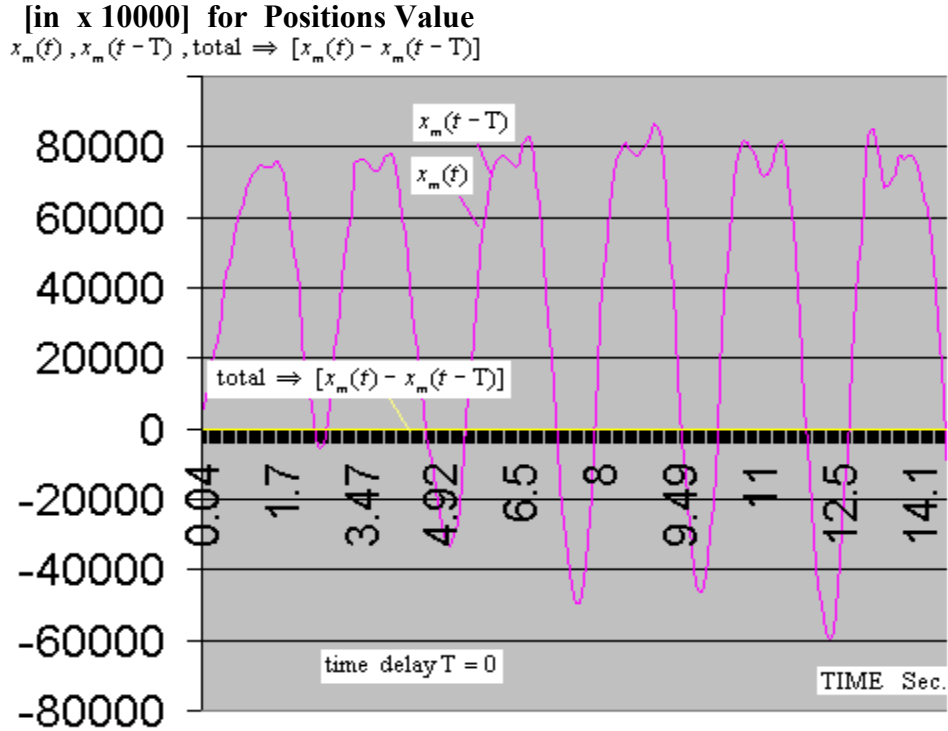


Fig. 31 The curve of total position $x_m(t) - x_m(t - \tau) \Rightarrow 0$ (yellow color line) for Time delay $T=0$ because the feedback position $x_m(t - \tau)$ compensates the position on master - $x_m(t)$. The curve $x_m(t)$ for master fits the curve $x_m(t - \tau)$ for slave on this figure. The system is stable.

Even with small time delay $T=0.2$ (Fig 35) displacement changes cause bigger difference $\dot{x}_m(t) - \dot{x}_m(t - \tau)$ between the velocity master and the velocity of the slave. As it is shown in Fig. 35 the total velocity on the master, in this case, increases and the system can be unstable due to the magnitude of the velocity and damper component in equation (35) are even greater than before.

Fig. 36 demonstrates that the larger time delay $T=1.0$ creates very a large amplitude of total velocity on the master, which produces increasing kinetic energy and causes catastrophic instability.

[in x 10000] for Positions Value

$$x_m(t), x_m(t-T), \text{total} \Rightarrow [x_m(t) - x_m(t-T)]$$

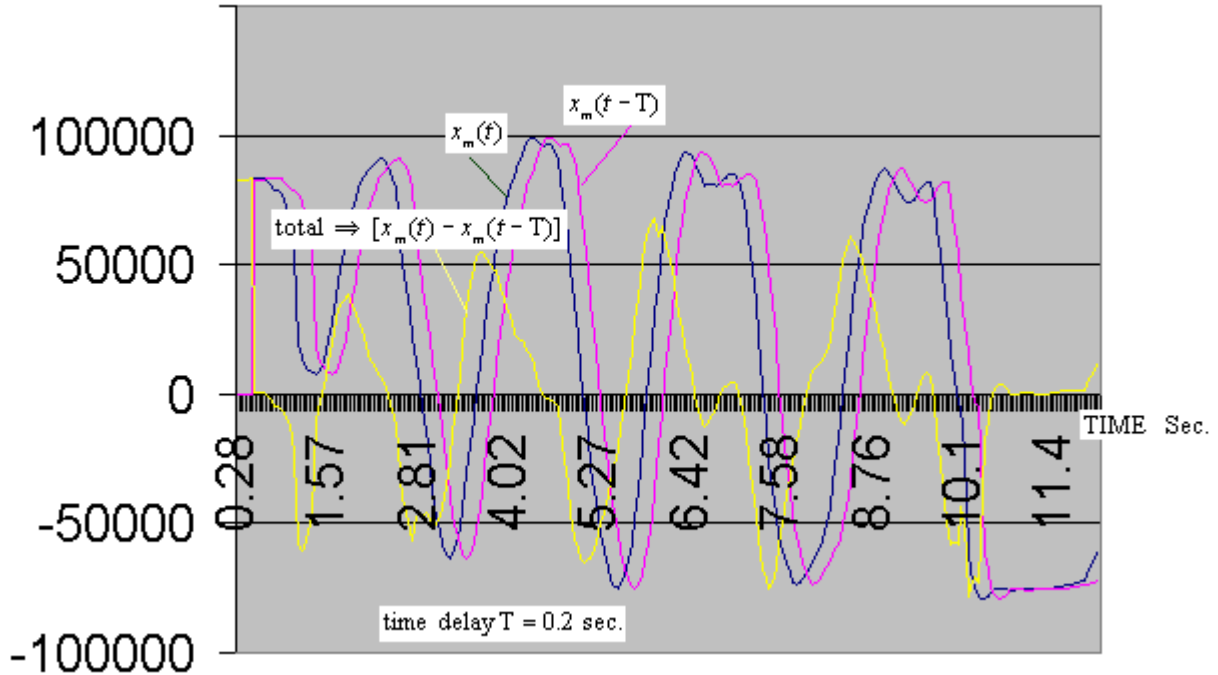


Fig. 32 The curve of total position $x_m(t) - x_m(t - \tau)$ (yellow color line) for Time delay $T=0.2$ the feedback position - $x_m(t - \tau)$ cannot compensate the position on the master - $x_m(t)$ completely. The curve $x_m(t)$ for the master does not fit the curve $x_m(t - \tau)$ for the slave on this figure and causes the position error. The magnitude of the total position $x_m(t) - x_m(t - \tau)$ is increased and the system is unstable.

Experimental results for analysis of stability using different input control strategies with time delay

- *In the previous examples for the case $k_F \approx 1$ an example $k_F = 0.00001$ we have an input control law that is similar to the control strategy in Massie's algorithm (see Fig. 17 and Fig. 18).*

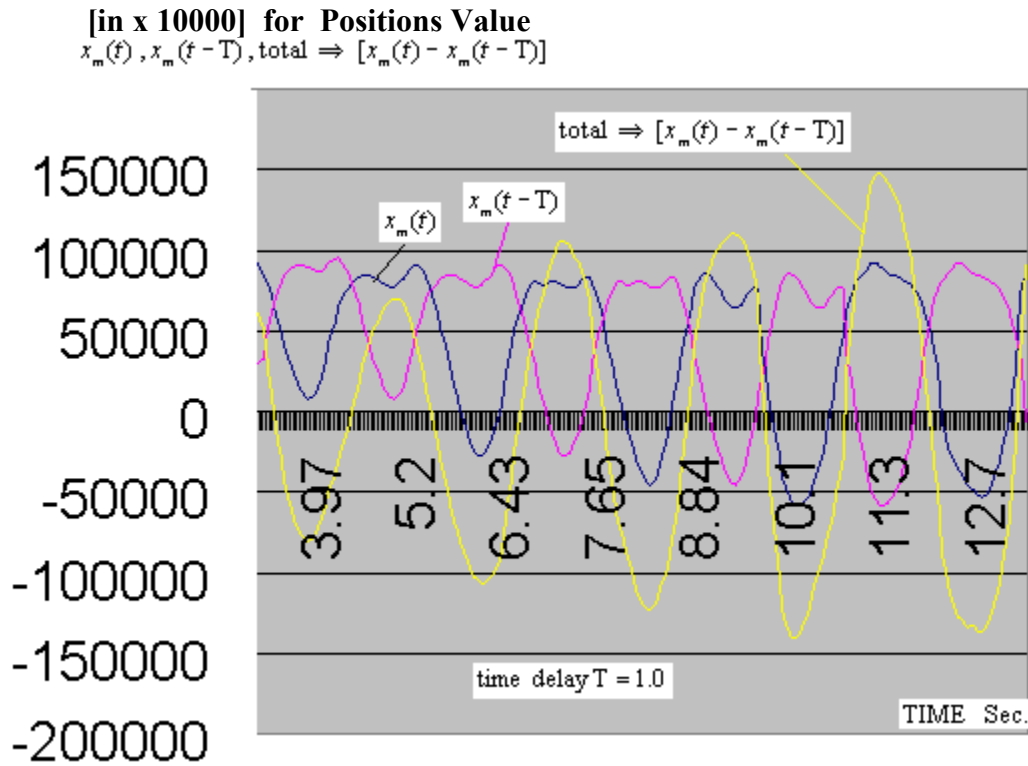


Fig. 33 The curve of total position $x_m(t) - x_m(t - \tau)$ (yellow color line) for Time delay $T=1.0$ the feedback position - $x_m(t - \tau)$ cannot compensate the position on the master - $x_m(t)$ completely. The curve $x_m(t)$ for the master does not fit the curve $x_m(t - \tau)$ for the slave on this figure and it causes a large position error (curve of changing position - $x_m(t - \tau)$ for slave is anti-phase to $x_m(t)$ for master). The magnitude of the total position $x_m(t) - x_m(t - \tau)$ is increased and the system is very unstable.

Fig. 38 gives the additional information for torque on the master and slave and total torque for PHANToM control. It causes the unstable dynamics of the system because the pulsation of the torque has a wide interval change from the maximum total value torque (an example point1) to a maximum limit of the torque (see Fig. 38). The torque responses on the slave are slower from the master and we have a superposition process of general torques on the motor-encoders. Even very small time-delays result in similar situations. The positive feedback velocity - $\dot{x}_m(t - \tau)$ is added to the velocity $\dot{x}_m(t)$ on the master. The curve $\dot{x}_m(t)$ for the master is added to the curve $\dot{x}_m(t - \tau)$

for the slave as indicated and causes large total velocity $\dot{x}_m(t) + \dot{x}_m(t - \tau)$. The magnitude of the total velocity $\dot{x}_m(t) + \dot{x}_m(t - \tau)$ is increased and the system becomes very unstable. This case matches equation (33) when we have “+” sign before the damper component because it has positive feedback.

The opposite result we have for test on Fig. 40 occurs with large time delay. Time delay $T = 1$ sec. tends to reduce the total $\dot{x}_m(t) + \dot{x}_m(t - \tau)$ because the velocity $\dot{x}_m(t)$ on the master and the velocity on slave $\dot{x}_m(t - \tau)$ have opposite signs (they are in anti-phase). Thus it provides reduced amplitude and better stability.

The velocity value in [rad/sec]

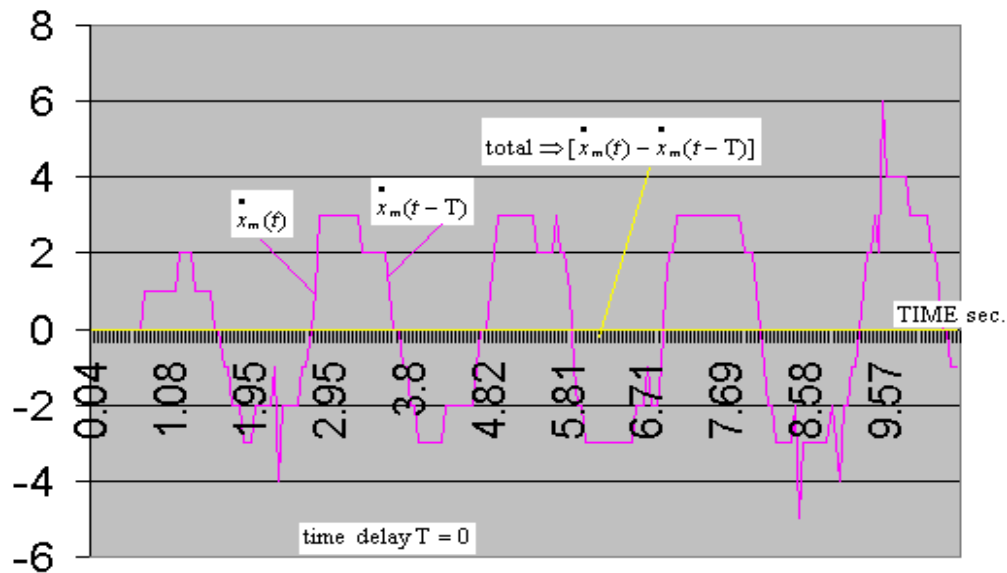


Fig. 34 The curve of changing total velocity $\dot{x}_m(t) - \dot{x}_m(t - \tau) \Rightarrow 0$ (yellow color line) for Time delay $T=0$ because the negative feedback velocity $-\dot{x}_m(t - \tau)$ compensates the position on master $\dot{x}_m(t)$. The curve velocity $\dot{x}_m(t)$ for the master fits the curve velocity $-\dot{x}_m(t - \tau)$ for the slave on this figure.

The velocity value in [rad/sec]

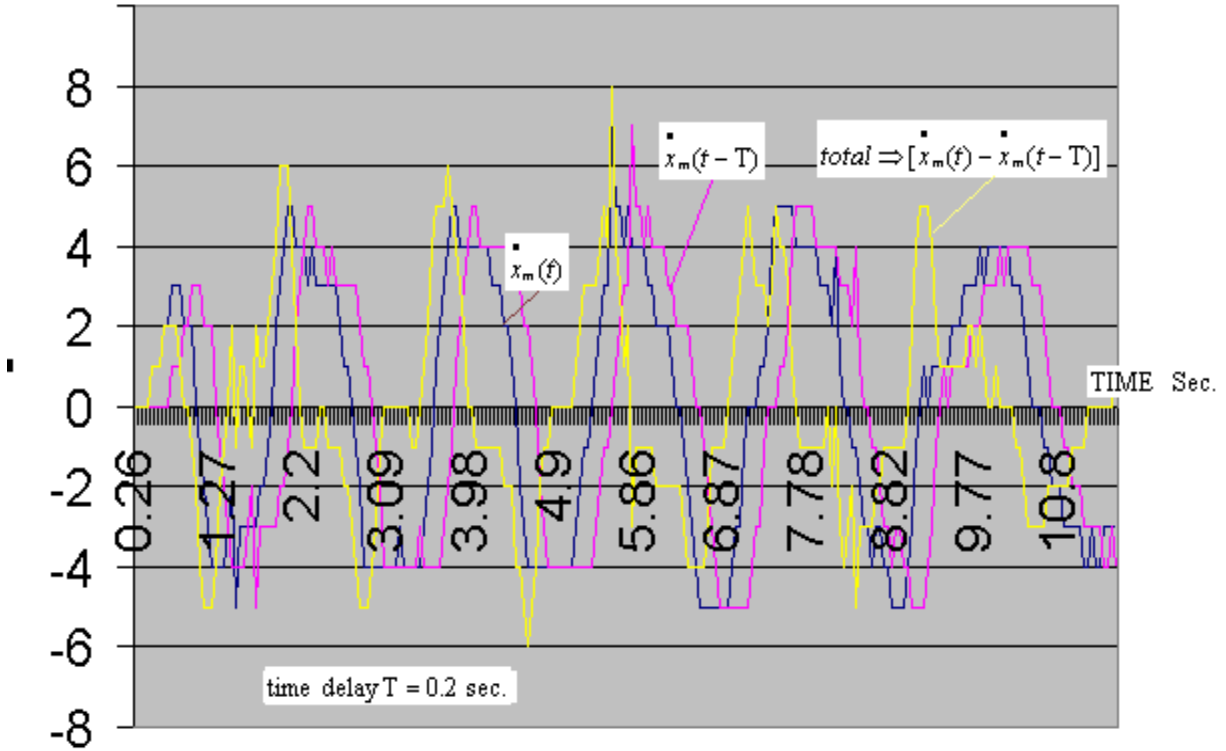


Fig. 35 The curve of total velocity $\dot{x}_m(t) - \dot{x}_m(t - \tau)$ (yellow color line) for a Time delay $T=0.2$. The feedback velocity $-\dot{x}_m(t - \tau)$ cannot compensate the velocity $\dot{x}_m(t)$ on the master completely. The curve, $\dot{x}_m(t)$ or master velocity, does not fit the curve $\dot{x}_m(t - \tau)$ for the slave in this figure and it causes the velocity error. The magnitude of the total velocity $\dot{x}_m(t) - \dot{x}_m(t - \tau)$ increases and the system is unstable.

The velocity value in [rad/sec]

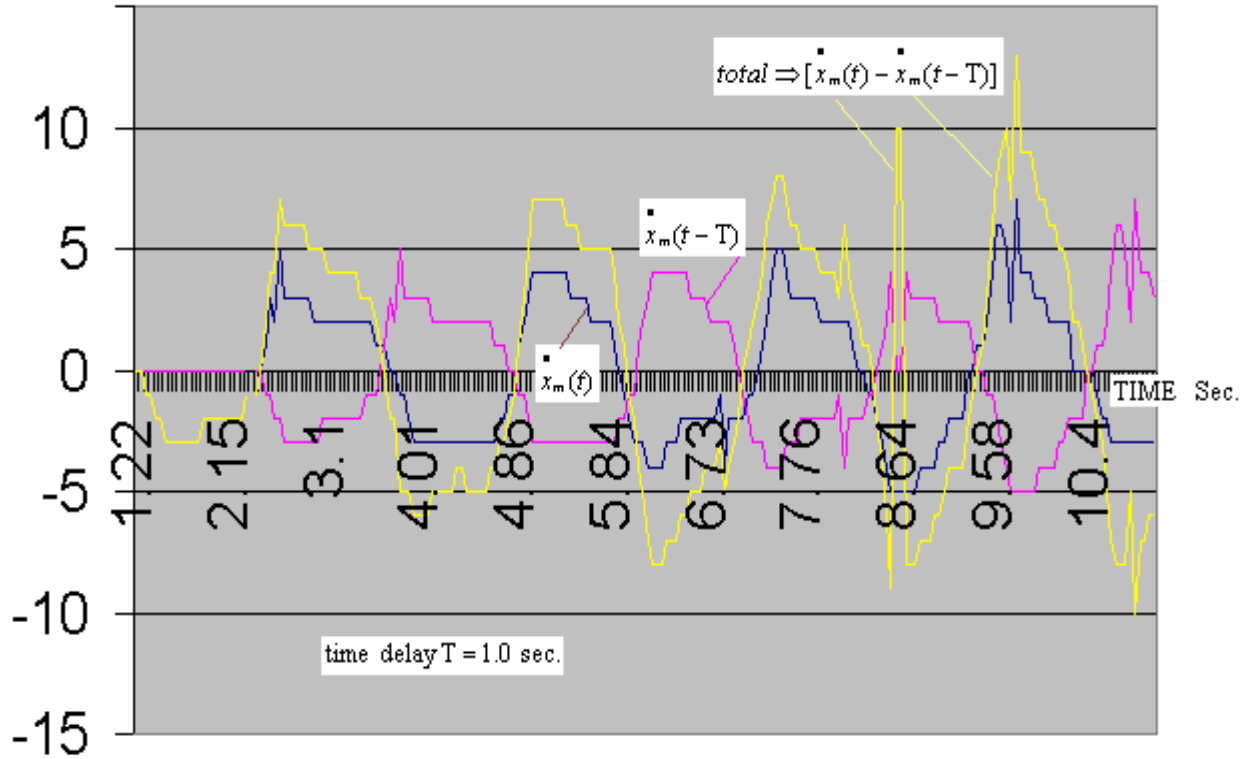


Fig. 36 The curve of total velocity $\dot{x}_m(t) - \dot{x}_m(t - T)$ (yellow color line) for Time delay $T=1.0$. The feedback velocity $\dot{x}_m(t - T)$ cannot compensate for the velocity $\dot{x}_m(t)$ on the master completely. The curve $\dot{x}_m(t)$ for the master does not fit the curve $\dot{x}_m(t - T)$ for the slave on this figure and it causes large velocity error (curve of changing velocity $\dot{x}_m(t - T)$) for the slave being out of phase with the master $\dot{x}_m(t)$. The magnitude of the total velocity $\dot{x}_m(t) - \dot{x}_m(t - T)$ increases and the system acts very unstable.

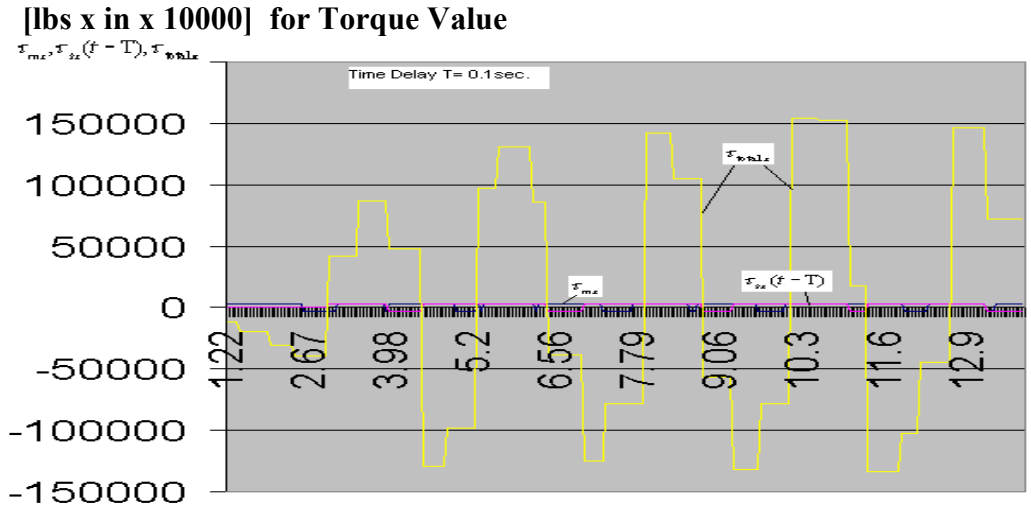


Fig. 37 The pulsation of the total torque has a wide interval but these values are corrected torque limits $T_MAX = \pm 3000$ and outside of this tunnel we have “force reflection tunnel” where the torque is very big and the operator may not be able to control this process. The system is very unstable. Despite a small time delay $T=0.1$ sec this system is very unstable because we use the big damper with “-” minus sign in component $B_e [x_m(t) - x_s(t-T)]$ in equation (32). The coefficient $k_F = 1$ but the big negative value of damper gives a negative torque on the master and slave also.

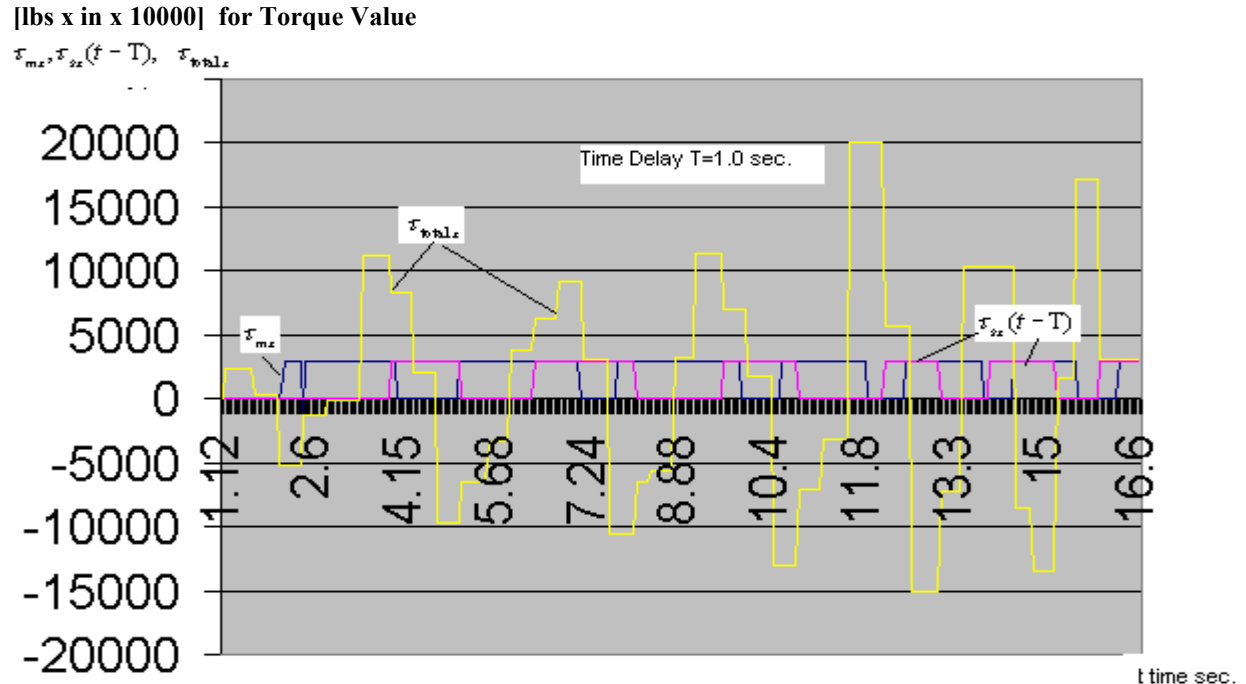


Fig. 38 The pulsation of the total torque has a wide interval but these values are corrected by the torque limits $T_MAX = \pm 3000$ and outside of this tunnel we have a dead area” where the torque is very big and the operator is not able to control this process. The system is very unstable. (Time delay $T=1.0$ sec., $k_F = 0.00001$).

The velocity value in [rad/sec]

$$\dot{x}_m(t), \dot{x}_m(t-T), \text{ total} \Rightarrow [\dot{x}_m(t) + \dot{x}_m(t-T)]$$

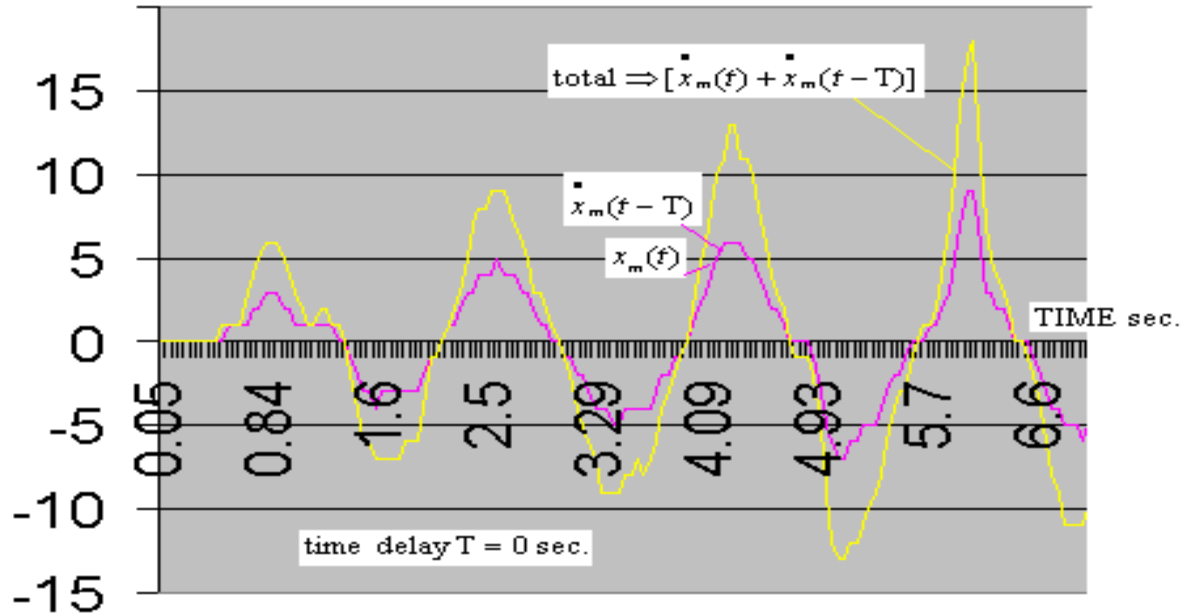


Fig. 39 The curve of total velocity $\dot{x}_m(t) + \dot{x}_m(t-T)$ for the positive feedback from the slave (yellow color line) without a Time delay $T = 0$.

The velocity value in [rad/sec]

$$\dot{x}_m(t), \dot{x}_m(t-T), \text{ total} \Rightarrow [\dot{x}_m(t) + \dot{x}_m(t-T)]$$

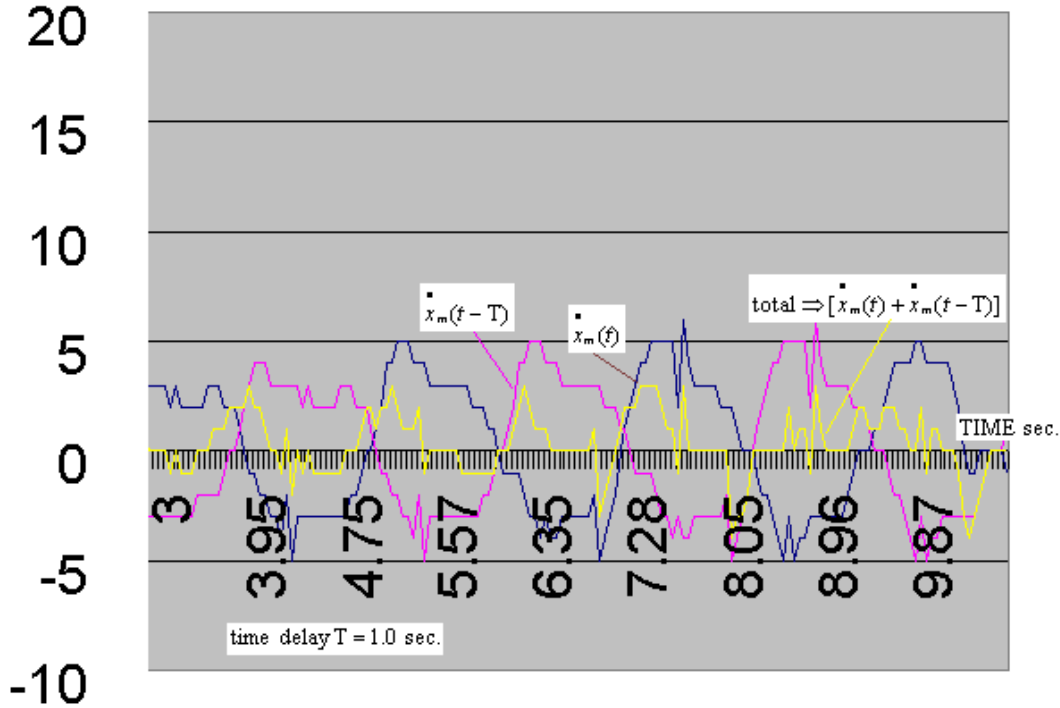


Fig. 40 The curve of total velocity $\dot{x}_m(t) + \dot{x}_m(t-T)$ for positive feedback from slave (yellow color line) without Time delay $T = 1$ sec. The system can be stable.

WAVE BASED TELEOPERATION

G. Niemeyer has pioneered a study in time delay teleoperation with wave variables. He has proposed the basic teleoperation for time delay where as at both sites the information, i.e. the velocity $\dot{x}(t)$ and force $F(t)$, are transformed into a wave domain via an appropriate transformation [Niemeyer 1996] see Fig. 41.

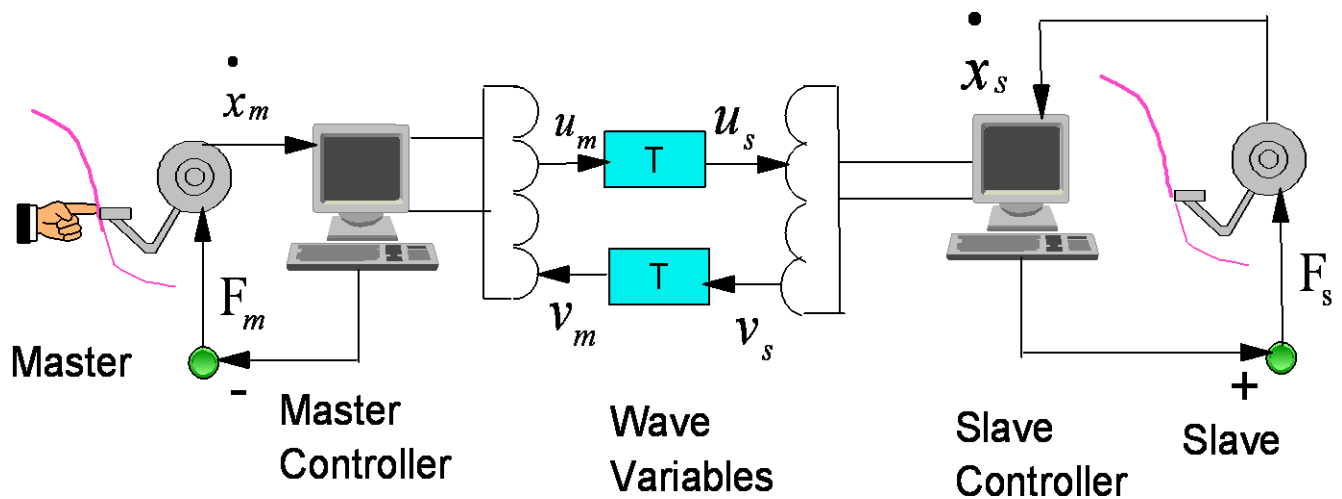


Fig. 41 The remote teleoperation with wave variables based on Niemeyer’s work [Niemeyer 1996].

Niemeyer has proposed a method to stabilize a bilateral teleoperation system with a constant time delay using the wave variables technique. In particular, he suggests to utilize “... left to right as the major direction of the flow of power, so that the right moving wave is transmitted from the master to the slave and vice versa” (see Fig. 41).

The equations describing the transmissions appear as follows:

$$\begin{aligned}
 u_s(t) &= u_m(t - \tau) \\
 v_m(t) &= v_s(t - \tau)
 \end{aligned}
 \tag{36}$$

where $u_m(t-\tau)$, represent the wave variables for the master $u_m(t-\tau)$ and $u_s(t)$ for the slave (from left to right) with a positive direction of power flow. Also $v_m(t)$, $v_s(t-\tau)$ wave variables for the master and slave, respectively, (from right to left) with a negative direction of stream power. While, based on this transformation, the input waves are specified as

$$u_m(t) = \frac{b\dot{x}_m(t) + \dot{p}_m(t)}{\sqrt{2b}} \quad (37)$$

$$v_s(t) = \frac{b\dot{x}_s(t) - \dot{p}_s(t)}{\sqrt{2b}} \quad (38)$$

Next, Niemeyer provides both possible output equations on both sides [Nieweyer 1996]:

$$F_m(t) = b\dot{x}_m(t) - \sqrt{2b}u_m(t) \quad (39)$$

$$F_s(t) = -b\dot{x}_s(t) + \sqrt{2b}v_s(t) \quad (40)$$

It is noted that Niemeyer does not specify nor restrict whether the force or velocity are considered as inputs or outputs at either side [Niemeyer 1996].

Extending these works we believe that there is a need to create suitable control strategies for robots and position tracking for a complex feature-based visual model in the query using a formal graphic specifications and equations. We can write the following equations for the PHANToM robot control with time delay:

On environment

$$F_{sx} = M_e [\ddot{x}_m(t) - \ddot{x}_s(t)] + B_e [\dot{x}_m(t) - \dot{x}_s(t)] + K_e [x_m(t) - x_s(t)] \quad (41)$$

we select \mathbf{b} as the input wave impedance which allows us to write the following equations based on possible output equation (38):

$$v_s(t) = \frac{b\dot{x}_m(t) - F_s(t)}{\sqrt{2b}} = \frac{b\dot{x}_m(t) - F_e(t)}{\sqrt{2b}} =$$

$$= \frac{b \dot{x}_m(t) - M_e[\ddot{x}_m(t) - \ddot{x}(t-\tau)] - \beta_e[\dot{x}(t) - \dot{x}(t-\tau)] - K_e[x(t) - x(t-\tau)]}{\sqrt{2b}}$$

or

$$v_s(t) = \frac{b \dot{x}_m(t) - F_s(t)}{\sqrt{2b}} = \frac{b \dot{x}_m(t) - F_e(t)}{\sqrt{2b}} = \frac{b \dot{x}_m(t) - M_e[\ddot{x}_m(t) - \ddot{x}(t-\tau)] - \beta_e[\dot{x}(t) - \dot{x}(t-\tau)] - K_e[x(t) - x(t-\tau)]}{\sqrt{2b}} \quad (42)$$

But we want to get the wave variables control strategies in equations (36):

$$v_m(t) = v_s(t-\tau) \quad (43)$$

Therefore we have, after the transformations of the equation (42),:

$$v_m(t) = \sqrt{\frac{b}{2}} \dot{x}_m(t-\tau) - \frac{M_e}{\sqrt{2b}} [\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] - \frac{B_e}{\sqrt{2b}} [\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] - \frac{K_e}{\sqrt{2b}} [x_m(t-\tau) - x_s(t-\tau)]$$

and $F_m(t)$ reflected to operator:

$$F_m(t) = \dot{x}_m(t) - \sqrt{2b} v_m(t)$$

or after transformations :

$$F_m(t) = b \dot{x}_m(t) - \sqrt{2b} \sqrt{\frac{b}{2}} \dot{x}_m(t-\tau) + M_e[\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] + \beta_e[\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] + \zeta_e[(x_m(t-\tau) - x_s(t-\tau))]$$

and

$$F_m(t) = b[\dot{x}_m(t) - \dot{x}_m(t-\tau)] + M_e[\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] + \beta_e[\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] + \zeta_e[(x_m(t-\tau) - x_s(t-\tau))]$$

In general, as it is shown above, we can let the following constants in this equations

represent optimal impedance values: $k_{bbe} b \neq k_{me} M_e \neq k_{mm} M_m \neq k_{ms} M_s$, $k_{bbe} b$

$\neq k_{be} B_e \neq k_{bm} B_m \neq k_{bs} B_s$ and $k_{bbe} b \neq k_{ke} K_e \neq k_{km} K_m \neq k_{ks} K_s$.

Therefore we have the reflected forces on the master robot for the 3 axes of coordinates:

$$F_{mx}(t) = b[\dot{x}_m(t) - \dot{x}_m(t - \tau)] + A_e[\dot{x}_m(t - \tau) - \dot{x}_s(t - \tau)] + \beta_e[\dot{x}_m(t - \tau) - \dot{x}_s(t - \tau)] + \zeta_e[(x_m(t - \tau) - x_s(t - \tau))]$$

$$F_{my}(t) = b[\dot{y}_m(t) - \dot{y}_m(t - \tau)] + A_e[\dot{y}_m(t - \tau) - \dot{y}_s(t - \tau)] + \beta_e[\dot{y}_m(t - \tau) - \dot{y}_s(t - \tau)] + \zeta_e[(y_m(t - \tau) - y_s(t - \tau))]$$

$$F_{mz}(t) = b[\dot{z}_m(t) - \dot{z}_m(t - \tau)] + A_e[\dot{z}_m(t - \tau) - \dot{z}_s(t - \tau)] + \beta_e[\dot{z}_m(t - \tau) - \dot{z}_s(t - \tau)] + \zeta_e[(z_m(t - \tau) - z_s(t - \tau))]$$

As it is shown in the work [Niemeyer 1996], wave communications will automatically take the role of either impedance or admittance as required by the rest of the system.

Experimental Validation of Increasing Stability Using Wave Variables

The developments with the wave variables are verified by modeling experiments. Consider Fig. 42

[lbs x 10000] for force value

$F_{mx}(t), F_{sx}(t-T), u_m(t), u_m(t-T), \tau_{total}$

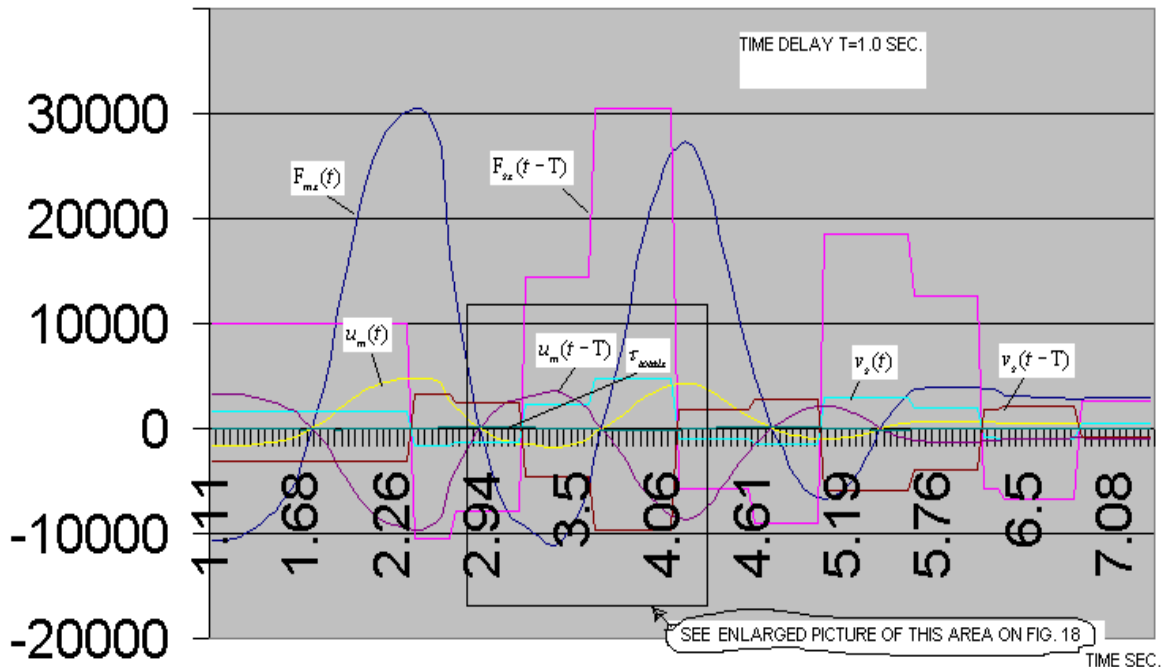


Fig. 42 The graphic real-time representation based on visual modeling for F_{mx} - master force with the $u_m(t)$ input wave variable and delayed wave variable $u_m(t - \tau)$. Also shown is the slave with the $F_{sx}(t - \tau)$ reflected force, the $v_s(t)$ input wave variable and the $v_s(t - \tau)$ delayed wave variable for $T=1.0$ sec. time delay. The system is very stable and has very a small total torque on the master - τ_{total} (see enlarged picture on Fig. 43 for this variable).

which show the results of modeling the forces and wave variables as on the master as on the slave. For clarity only the visual interpretation on the chart of the data for the x axis is shown here. However, similar results were obtained absolutely for all x, y, z axes in our experiments. The delay was set at $T=1.0$ sec. Above running experiments in this work, clearly shows that this time delay results in an unstable system for all cases without using the wave variables. But as it is shown in Fig. 42, the system, using the wave variables, allows us to create a very stable system with small magnitude for τ_{total} total torque on the master. For clarity Fig. 43 allows us to display a more enlarged image of area with little values of the τ_{total} total torque on Fig. 42.

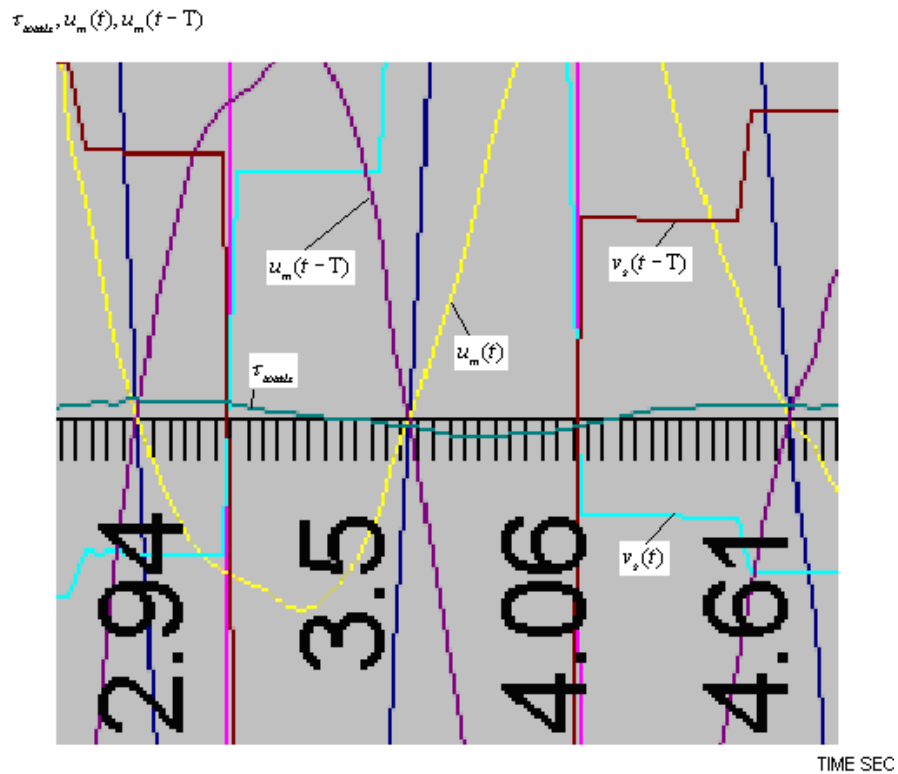


Fig. 43 The enlarged area for Fig. 42 where we clearly see that the τ_{total} total torque on the master has very little value and system is very stable for $T=1.0$ time delay for the case with using the wave variables.

This torque technique is represented with a larger scale 100 times greater. Fig. 43 demonstrates that the wave variables $u_{mx}(t - \cdot)$ and $v_{sx}(t - \cdot)$, after time delay make an

effect, are pushing in opposition to the forces corresponded to operator and slave. It allows the stabilization of the system because in equation (32) the components are:

$$\begin{aligned}
 M_e[\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] &\approx \tau \\
 B_e[\dot{x}_m(t-\tau) - \dot{x}_s(t-\tau)] &\approx \tau \\
 K_e[(x_m(t-\tau) - x_s(t-\tau))] &\approx \tau
 \end{aligned}
 \tag{44}$$

The instability of the system in this case may be only caused by the component $-b[\dot{x}_m(t) - \dot{x}_m(t-\tau)]$. However, this system is almost stable. As an example Fig. 44 and Fig. 45 indicate that the system is stable for a time delay $T = 1$ sec. (Fig. 44) as for $T = 0.2$ sec. (In Fig. 45 we note again that the scale factor 100 is used for these charts to demonstrate the total torque with greater clarity). In this case the system can be instable when the component $-b[\dot{x}_m(t) - \dot{x}_m(t-\tau)]$ is large in adding to the other components such as the uncertainties, contact with absolutely rigid restrictions, errors in structure or the control input law of control system, and large drifting that will be considered below in this work.

Experimental Results

- 1) We have displayed that the system with wave variables is very stable with time delay $T=0$ and with some time delays $T > 0$ (it can be large time delay also) because the wave variables can compensate for the instability using the reflected forces via damping. It can be realized without a P.D. controller using an optimal impedance “b” only. Unstable systems with large time delay can be stabilized using the wave variables.

2) We experimented a great deal on combinations of parameters using the wave controllers based on computer modeling in Real-Time system for the PHANToM master and slave in a virtual 3D environment with time delay and without delay for non-linear control laws . The possible solutions and conditions of experiments will be utilized into a Knowledge Database.

3) Results of the experiments allow the design of the self-organizing control system based on IPVTS and a heuristic search algorithm for Real –Time control. The system will define needed parameters and control them. The main goal of this system is to create intelligent control in robotics based on a virtual reality system with time-delay and using the wave variable method.

[lbs x 10000] for force value

$u_m(t), u_m(t-T), v_s(t-T), v_s(t), \tau_{total}$

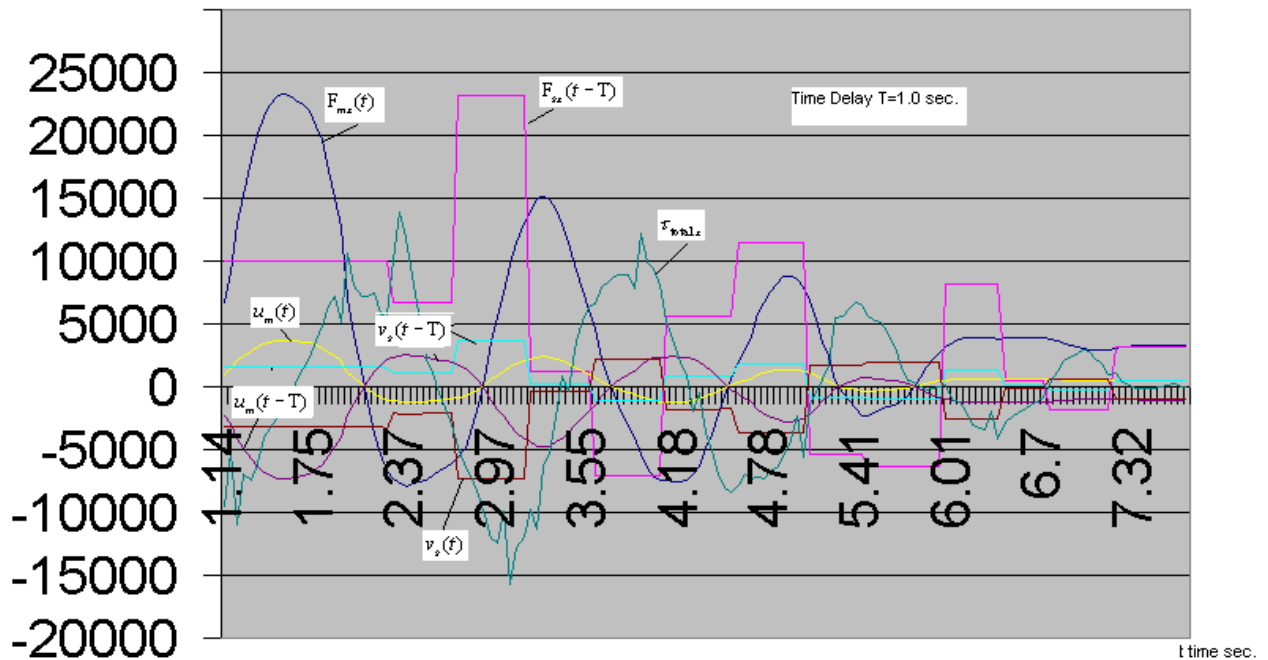


Fig. 44 The modeled curves for the $u_m(t)$ input wave variable , the delayed wave variable $u_m(t- \tau)$ on the master and the $v_s(t)$ input wave variable , the $v_s(t- \tau)$ delayed wave variable on the slave for $T=1.0$ sec. time delay are demonstrated here. The system is very stable and has very small total torque on master

- τ_{stab} . For clarity the scale of total torque on master - τ_{stab} has been increased in 100 times real time (In reality this torque is less scaled by 1/100).

Future work

It is important to research Wave-Based Teleportation for the PHANToM with integrated IPVTS for different variable parameters modified over wide limits:

- a) wave variables b) single variable combinations of both velocity and force data c) for large delays with methods including wave impedance matching, d) wave filtering and wave controllers e) wave reflection g) research the Drifting Problem and the three Feedback Paths Problem.

We will change these combinations of parameters using the wave controllers based on computer modeling in a real-time system with and without delay. The possible solutions and conditions of experiments will be to utilize into a Knowledge Database. The goal of these experiments is to compare the stability of the control system with the wave variables using the different types of sensory information, for large delays with methods including wave impedance matching, the wave filtering, and wave controllers.

It allows us to use the advantages of the intelligent m-dimensional parametric control based on IPVTS.

- Run the experiment based on IPVTS together with the PHANToM with a predictor incorporated inside the wave junction.

Run the experiments for target detection using IPVTS together with the PHANToM based on the different types of sensory information and when the cues work in conjunction with the intelligent m-dimensional parametric control. It is

[lbs x 10000] for force value
 [in x 10000] for position value

$u_m(t), u_m(t-T), v_s(t-T), v_s(t), \tau_{\text{whl}}$

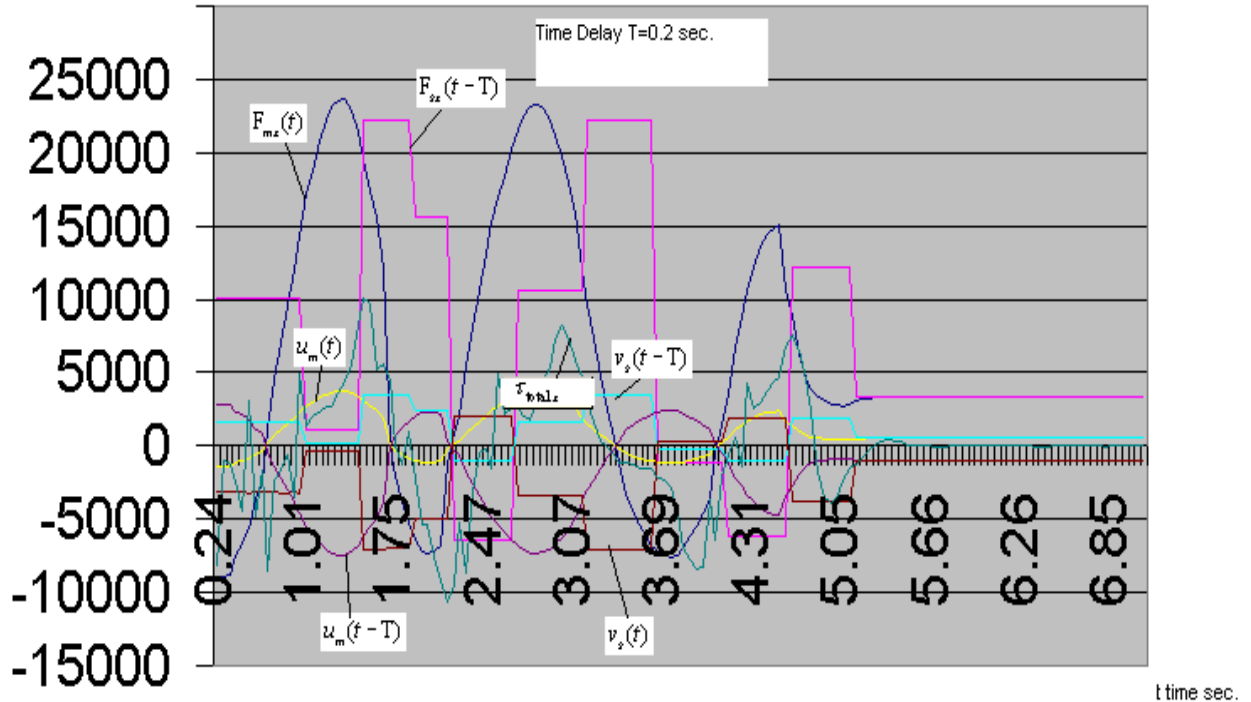


Fig. 45 The wave variables for more little time delay $T=0.2$ sec. As it is shown the system is very stable in this case also. The modeled curves for the $u_m(t)$ input wave variable, the delayed wave variable $u_m(t - \cdot)$ on master and a the $v_s(t)$ input wave variable, the $v_s(t - \cdot)$ delayed wave variable on the slave

- necessary to analyze the errors that might occur in the control process. It is desired to use the advantage of the intelligent m-dimensional parametric control based on IPVTS.
- Use Transfer Functions $H(s) = V(s)/U(s)$ to further illustrate the Visual Model to better understand the meaning and behavior of wave variables and systems as criteria for control.

- Create the optimization system in IPVTS for the design of classic elements in wave space for basic controllers such as the PHANToM Robot, Haptic Stick, 3D-Audio Computer Controller and Computer Based MEMS controllers.

MODELING OF DYNAMIC AND UNCERTAINTIES BASED ON TRANSFER FUNCTION

Suppose we specify the dynamics in the environment such that

$$G(s) = \frac{\dot{\tau}_e(s)}{F_e(s)} \quad (45)$$

next we have

$$\dot{x}_e(s) = \tau_e(s) * G(s) \quad (46)$$

where $G(s)$ some stable transfer function. Then the equation (37) will incorporate the reflected wave:

$$\sqrt{2b}v_s(s) = \tau_e \dot{x}(s) - \tau_e(s) \quad (47)$$

$$\sqrt{2b}v_s(s) = \tau_e [F_e(s) * G(s)] - \tau_e(s) \quad (48)$$

or after the transformations

$$\sqrt{2b}v_s(s) = b_e * F_e(s) * [G(s) - 1] \quad (49)$$

Now we can consider

$$v_m(t) = v_s(t - \tau)$$

and at the master robot :

$$F_m(t) = \tau \dot{x}_m(t) - \sqrt{2b} * v_m(t) \quad (50)$$

We wish to design $F_m(t)$ at the master:

$$F_m(t) = b \dot{x}_m(t) - \sqrt{2b} * v_s(t - \tau) \quad (51)$$

then choose

$$F_m(t) = b \dot{x}_m(t) - \sqrt{2b} * b_e * F_e(t - \tau) * [G(s) - 1] \quad (52)$$

therefore we have the $F_m(t)$ force on the master robot that can be controlled, with desired characteristics and the desired transfer functions $G(s)$ using additional elements (an example damper), which are inserted into the P.D. PHANToM controller .

We can control the desired $F_m(t)$ on the master robot by varying the transfer function $G(s)$ using the three dimensional model with the x,y,z axes :

$$\begin{aligned} F_{mx}(t) &= b \dot{x}_m(t) - \sqrt{2b} * b_e * F_{ex}(t - \tau) * [G_x(s) - 1] \\ F_{my}(t) &= b \dot{x}_{my}(t) - \sqrt{2b} * b_e * F_{ey}(t - \tau) * [G_y(s) - 1] \\ F_{mz}(t) &= b \dot{x}_{mz}(t) - \sqrt{2b} * b_e * F_{ez}(t - \tau) * [G_z(s) - 1] \end{aligned} \quad (53)$$

It is very important for optimal design to create the best match of the impedance between the P.D. controller and the wave transmission. We propose, using a layout as shown in Niemeyer's work, (see Fig. 41) for our visual model of the PHANToM robot.

We may add a noise or uncertainties into the wave variable on slave robot $v_s(t)$. Suppose that the signal $v(t)$ is the one we wish to track. We want $x_e(t)$ to follow the target in space $v(t)$. The tracking error in this case is desired:

$$x_{error} = v(t) - \dot{x}_e(t) \quad (54)$$

We further use three methods for getting of signal $v(t)$ on our model:

- As a wave with $v(t) = \Lambda * \sin \omega$;

- As random signal $v(t)$, this signal is generated by the moving mouse along the x, y (z signal is generated) in time as control events in model;
- As one random signal $v(t)$.

We would like x_{error} to be asymptotically stable such that

$$G(s) = \frac{\dot{x}_e(s)}{F_e(s)} = \text{which implies some stable transfer function (i.e. we have a}$$

stable impedance in the environment).

We select $v_s(s)$ in the environment to consist of two terms:

$$\sqrt{2b}v_s(s) = \dot{x}_e(s) - \dot{x}_e(s) = \text{tracking error} + \text{impedance interaction} + \text{noise error} + \text{master}$$

Hence

$$\sqrt{2b}v_s(s) = \dot{x}_e(s) - \dot{x}_e(s) = x_{error}[e^{-\zeta}] + \eta(t) + v(t) + \beta_n(t) \quad (55)$$

$$\text{Hence } b_e \dot{x}_e(s) - \dot{x}_e(s) = \eta(t) + v(t) + \beta_n(t)$$

$$\text{Hence } F_m(t) = \dot{x}_m(t) - \sqrt{2b} [v_s(t-\tau)]$$

$$F_m(t) = \dot{x}_m(t) - \sqrt{2b} [\gamma(t-\tau)]$$

$$F_m(t) = \dot{x}_m(t) - \sqrt{2b} [\gamma(t-\tau) + \beta_n(t-\tau) + \eta(t)] \quad (56)$$

$$\text{where } \gamma(t-\tau) = [v_s(t-\tau)]$$

IMPEDANCE MATCHED TELEOPERATOR

In this case, described by Niemeyer, the wave communications acts as a true admittance. It observes forces on both sides and updates the desired motions as needed. Both master and slave are controlled via an impedance matched P.D. controller.

SLAVE AND MASTER CONTACT WITH ENVIRONMENT

Preliminary research with the PHANToM has shown that the time delay in the condition of contact causes a very unstable process. However, to continue this research is needed more detail with the wave variable method.

FUTURE RESEARCH

Use Wave-Based Teleportation for the PHANToM with integrated IPVTS for different variable parameters changing in wide limits:

- a) wave variables
- b) single variable combinations of both velocity and force data
- c) for large delays with methods including wave impedance matching,
- d) wave filtering and wave controllers
- e) wave reflection
- g) research the Drifting Problem and three Feedback Paths Problem.

We will change these combinations of parameters using the wave controllers based on computer modeling in a real-time system with and without time delays. The possible solutions and conditions of experiments will be utilized into a Knowledge Database. The goal of these experiments is to compare the stability of the control system with the wave variables using the different types of sensory information, for large delays and with methods including wave impedance matching, the wave filtering, and wave controllers.

It allows us to use the advantage of the intelligent m-dimensional parametric control based on IPVTS.

- Run the experiment based on IPVTS together with the PHANToM with a predictor incorporated inside the wave junction.
- Run the experiments for target detection using IPVTS with the PHANToM based on the different types of sensory information and in conjunction with the cues which work together with the intelligent m-dimensional parametric control. Analyze the errors that might occur in the control process. It is desired to use the advantage of the intelligent m-dimensional parametric control based on IPVTS.
- Use Transfer Functions $H(s) = V(s)/U(s)$ to further illustrate in a Visual Model the meaning and behavior of wave variables and systems as criteria for control.
- Create the optimization system in IPVTS for the design Classic Elements in Wave Space for basic controllers such as PHANToM Robot, Haptic Stick, 3D-Audio Computer Controller and Computer Based MEMS Controller.

REFERENCES

- P. Buttolo, P. Braathen and B. Hannaford, "Sliding Control of Force Reflecting Teleoperation: Preliminary Studies," *PRESENCE*, Vol. 3, pp. 158–172, 1994.
- Dori, D. and M. Weiss . A Scheme for 3D Object Reconstruction from Dimensioned Orthographic Views, *Engineering Applications of Artificial Intelligence*, Vol. 9, N 1, 53-64.
- J. R. Flanagan and R. S. Johansson, Hand Movements, *Encyclopedia of the Human Brain*, Vol. 2, Elsevier Science (USA). 2002, pp.1-16,
- J. Floyd. Haptic Interaction with Three-Dimensional Bitmapped Virtual Environments. *M. Eng Thesis*, MIT, Department of Electrical Engineering and Computer Science, May 1999.
- J. D. Foley, A. Van Dam, Feiner, and Hughes. *Computer Graphics: Principles and Practice*. Addison Wesley, 2 edition, 1996, 620 p.

R. Fosner , OpenGL Programming for Windows 95 and Windows NT , Addison-Wesley, 1997, 288 p.

Installation & Technical Manual, "PHANToM Haptic Interface, SensAble Devices Inc." Cambridge, MA, Updated version, July 1995

Lin, P.D. and I. J.Tsai. The Machining and On-line Measurement of Spatial Cams On Four-Axis Machine Tools, *International Journal Machine Tools & Manufacture*. **Vol. 36, N 1.**

T. Massie and K. Salisbury, "*The PHANToM Haptic Interface: A Device for Probing Virtual Objects*," *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, Chicago, IL, November 1994.

Niemeyer and Jean-Jacques E. Slotine, Stable Adaptive Teleoperation, *IEEE Journal of Oceanic Engineering*, Vol.16, No.1, pp. 152-162, 1991.

G. Niemeyer Using Wave Variables in Time Delayed Force Reflecting Teleoperation, Thesis of the PhD dissertation in Mechanical Engineering and Information Science. MIT, Department of Aeronautics and Astronautics, 1996

D. W. Repperger, M. W. Haas, B. J. Brickman, L. J. Hettinger, L. Lu, and M. M. Roe, "Design of A Haptic Stick Interface as A Pilot's Assistant in A High Turbulence Task Environment," *Perceptual and Motor Skills*, vol. 85, 1997, pp. 1139-1154.

D. W. Repperger, C. A. Phillips, and T. L. Chelette, "Performance Study Involving A Force Reflecting Joystick For Spastic Individuals Performing Two Types of Tracking Tasks," *Perceptual and Motor Skills*, December, 1995, No. 81, pp. 561-562.

D. W. Repperger, C. A. Phillips, and T. L. Chelette, "Study of Spatially Induced 'Virtual Force' With An Information Theoretic Investigation of Human Performance," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 10, October, 1995, pp. 1392-1404

D. W. Repperger, "Using an Intelligent Machine to Modify or Adapt Human Behavior," Book Chapter, pp. 289-313, in *Intelligent Machines- Myths and Realities*, editor, Clarence W. de Silva, CRC Press, 2000.

D. W. Repperger, "Inverse Kinematic Control Algorithms with a Reduced Coriolis Component for Use in Motion Simulators," *Transactions of ASME, Journal of Dynamic Systems, Measurement , and Control*, Vol. 117, October, 1995, pp. 570-577.

Scientists 'touch' via Internet, Technology could allow people to feel others over Web, CNN.com, Oct. 29, 2002, London-Boston,
<http://www.cnn.com/2002/TECH/internet/10/29/tech.handshake.reut/index.html>

N. Tikhonov Intelligent Parametric Visual Thinking System (IPVTS) as a Paradigm for Control Strategies in Robotics . NRC Research Proposal, NRC/AFRL, 2002