INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Université d'Ottawa • University of Ottawa

PERMISSION DE REPRODUIRE ET DE DISTRIBUER LA THÈSE

PERMISSION TO REPRODUCE AND DISTRIBUTE THE THESIS

NOM DE L'AUTEUR / NAME OF AUTHOR:	BASIC, Goran
ADRESSE POSTALE / MAILING ADDRESS:	3-372 BLAKE BOULEVARD VANIER ON K1L6L2
GRADE / DEGREE:	**ANNÉE D'OBTENTION / YEAR GRANTED**
M.A.Sc. (Mechanical Engineering)	2003
TITRE DE LA THÈSE / TITLE OF THESIS: "HARDWARE-IN-THE-LOOP SIMULATION OF MECHANICAL LOADS FOR MECHATRONICS SYSTEM DESIGN"	

L'auteur permet, par la présente, la consultation et le prêt de cette thèse en conformité avec les règlements établis par le bibliothécaire en chef de l'Université d'Ottawa. L'auteur autorise aussi l'Université d'Ottawa, ses successeurs et cessionnaires, à reproduire cet exemplaire par photograhie ou photocopie pour fins de prêt ou de vente au prix coûtant aux bibliothèques ou aux chercheurs qui en feront la demande.

Les droits de publication par tout autre moyen et pour vente au public demeureront la propriété de l'auteur de la thèse sous réserve des règlements de l'Université d'Ottawa en matière de publication de thèses.

The author hereby permits the consultation and the lending of this thesis pursuant to the regulations established by the Chief Librarian of the University of Ottawa. The author also authorizes the University of Ottawa, its successors and assignees, to make reproductions of this copy by photographic means or by photocopying and to lend or sell such reproductions at cost to libraries and to scholars requesting them.

The right to publish the thesis by other means and to sell it to the public is reserved to the author, subject to the regulations of the University of Ottawa governing the publication of theses.

N.B. LE MASCULIN COMPREND ÉGALEMENT LE FÉMININ

DEC. 16 2002 — Goran Basic

DATE (AUTEUR) SIGNATURE (AUTHOR)

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

BASIC, Goran

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Mechanical Engineering)

GRADE - DEGREE

Mechanical Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Hardware-in-the-Loop Simulation of Mechanical Loads For Mechatronics System Design

Dan-Sorin Necsulescu

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

M. Liang

J.Z. Sasiadek

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

HARDWARE-IN-THE-LOOP SIMULATION OF MECHANICAL LOADS FOR MECHATRONICS SYSTEM DESIGN

Goran Basic

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE

In Mechanical Engineering

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering
University of Ottawa
Ottawa, Canada

Abstract

Current research efforts in Hardware-In-The-Loop (HIL) simulations are directed toward testing Electronic Control Units, simulated digitally, in a physical experimental setup. This thesis presents different approach of using HIL simulations. Active and passive mechanical loads can be simulated physically on direct drive motors, under computer control. The work in thesis is based on effort-flow concept, which allows components of the experimental setup to be replaced as needed by physical or digital model. The only requirement that has to be satisfied is that elements of the setup retain their inputs and outputs, in the form of effort and flow pairs.

Based on this theory, the new experimental setup was built, a generic HIL setup containing two DC motors, which are connected by shaft. One of the motors is used to actuate the system, while another motor represents the physical simulator. Based on the sets of derived formulas, physical simulator is able to simulate active and passive loads.

Three different experimental levels are presented in the thesis. The open loop, current control and torque control experiments.

The experimental results prove the concept in whole and show that theory can be applied in real world applications. The focus in this research is on simulation of nonlinear loads, whose models are presented by sets of nonlinear differential equations. Digital simulations require solutions of those equations and it is demanding job. The method of physical simulations presented in this thesis shows simpler way of simulating complex loads.

Acknowledgment

I would like to express my deepest appreciation to all those who helped me to complete this research.

I would like to thank, in particular, my supervisor Dr. Dan Necsulescu, whose expertise, guidance, reviews, criticism and suggestions helped me throughout my research.

Thanks to Bumsoo Kim, for introducing me to dSPACE system and transferring his knowledge about this system to me.

I would like to thank, as well, Ljubenko Porubovic and all other technicians from our shop who produced the experimental setup.

Special thank to Doug and Sylvia Cutler for encouraging me to pursue my dreams and for supporting me throughout this research.

For all their encouragement and support, I would like to thank my friends: Jill, Bruce, Brennan and Sarah Cassidy.

In particular, I would like to thank my wife Renata for her continuing support, encouragement and understanding throughout the whole research.

Sincerely

Goran Basic

Table of Contents

List of Figures

Appendix A

Appendix B

Appendix C

Appendix D

Appendix G

List of Tables

List of Symbols

B	coefficient of viscous friction
B_{fl}	nonlinear coefficient of viscous friction of a fan
B_l	coefficient of viscous friction of a load
B_{l1}	coefficient of viscous friction of a load 1
B_{lsim}	coefficient of viscous friction of a simulated load
B_m	coefficient of viscous friction of a DC motor
B_{pit}	coefficient of viscous friction of a DC motor (Pittman 9232)
F	force
F_1	force 1
F_{dry}	dry friction
F_{dryl}	dry friction of a load
F_{drypit}	dry friction (Pittman 9232)
F_{drysim}	simulated dry friction
f	frequency
G	shearing modulus of elasticity
i	current
I_{cl}	current limit
J	inertia
J_{eq}	equivalent inertia
J_l	inertia of a load

J_{l1}	inertia of a load 1
J_{lsim}	inertia of a simulated load
J_m	inertia of a DC motor
J_{pit}	inertia of a DC motor (Pittman 9232)
K	motor constant
K_{amp}	amplifier constant (transfer function)
K_{pit}	motor constant (Pittman 9232)
K_s	shaft stiffness
K_t	tachometer constant
L	motor inductivity
L_s	length of a shaft
M	mass
M_1	mass 1
M_{sim}	simulated mass
N_i	number of gear teeth
R	motor resistance
R_{cl}	current limit resistance
R_f	amplifier gain resistance
R_m	amplifier gain resistance
R_s	sensing resistance
t	time
T	torque
T_C	Coulomb friction
t_d	actuator torque signal
T_{dry}	dry friction moment
T_l	load torque
t_l	load torque signal

T_{l1}	load 1 torque
T_{lsim}	simulated torque
T_m	motor torque
T_s	stiction
U	voltage
v	velocity
x	displacement
X	input magnitude
θ	angular position
φ	angular position
φ_l	angular position of a load
ω	angular velocity
ω_l	angular velocity of load
ω_{lpf}	low pass filter cutoff frequency

1. INTRODUCTION

In today's world of fast developing technologies, the field of mechatronics although relatively young is becoming very important. Every-day life is becoming more demanding and the products that we are using are becoming more complex. Cars are not only mechanical devices as they were at the beginning of the last century. More complex mechatronics products replaced simple car components. There are cameras and signal devices all around us to help us manage flow of people and goods. Process of production improved since robots took over the work, which was used to done by people. Welding, assembling, cutting, and controlling quality of products are responsibility of machines. The space explorations are continuing. Wars are not fought with the same weapons used hundreds years ago. Machines are banking for us, warning us about danger, entertaining us. The benefits of new technologies are remarkable. Not only that those products are more complex, their designing process is more demanding as well. Today engineers from different fields have to work together to be able to produce a competitive product.

1.1 Motivation for the research and the research goal

The mechatronics is the field that intersects four basic fields: mechanics, electronics, computers and control. Each of these disciplines represents big challenge by itself. Integration of all of them together is technological art. Expressions like rapid prototyping, cost effective simulations and modular design are more and more part of our everyday conversations. A Hardware-In-The-Loop (HIL) simulations and testing are part of mechatronics developed with the mission to design products rapidly and with cost effective technology.

The Hardware-In-The-Loop methodology is well known for years and has been used in different fields. Moreover, every field has specific requirements, which have to be

followed with every design. Knowing the basics of the HIL, we may conclude that some general rules apply. Effort-flow concepts, modularity, use of hardware parts or software simulations interchangeably are just some of those common points. So far, the Hardware-In-The-Loop was used for different purposes. Actuators, control units and loads were designed as real hardware or simulated using software. The motivation behind this thesis was to try to generalize the HIL approach and to prove that interchangeability between physical components and software-simulated components can be implemented.

The goal was to design and implement a very generic Hardware-In-The-Loop experimental setup with the purpose to simulate linear and nonlinear loads using not only software simulation but also real hardware (DC motor), which would be supported by additional signals generated by a computer. Theoretically, building as well as modeling of such a setup is possible by applying the concepts of effort-flow, modularity and interchangeability. The system should be as generic as possible and should allow monitoring different characteristics of a simulated and a real load, including velocity, torque and position variables.

2. LITERATURE REVIEW

2.1. HIL simulations

Recent developments in microelectronics and computer technology allowed the development of several disciplines including robotics, intelligent structures and mechatronics. The Hardware-In-The-Loop (HIL) simulations are one of the examples where such technological development shows attractive results. Classical computer simulations were used to simulate complete systems using mathematical models of system components, but the results were not always completely satisfactory. Mathematical models are, by default, created using approximation and linearization, sometimes neglecting important characteristics of the system. The HIL simulations use real hardware instead of some mathematically simulated models and thus preserve all necessary characteristics of those components.

The advantages of simulating systems using HIL experimental setups are significant. Engineers can get characteristics of parts of the system or their behavior even before those parts are physically made. Tuning the parameters of simulation can make the process of design faster and less costly. Systems may consist of only a few real parts, while the rest could be based on software simulation. Usually, the parts that are of major interest are physically present. Software simulated parts could be parametrically altered. Simulations can take place in laboratory conditions even for processes that are in reality designed to operate outdoors. External conditions can be simulated by software.
HIL simulations can reduce time dependence. The same results can be obtained in a short time for the process that can take days or months if tested in natural conditions.

Repeatability of the HIL simulations is high. For some systems, creating the same environmental conditions is a crucial part of the simulation. Sometimes, using classical testing methods is very hard, or even impossible, trying to repeat the same initial conditions.

HIL simulations can be repeated as many times as needed, with the same initial conditions. The model parameters could be changed as well as the model itself.

Using HIL simulations, the development time decreases while quality, reliability and safety increases.

HIL simulations are already important development method for different engineering fields. Automotive industry, military, robotics and space programs are just some of HIL users.

2.2 Robotics

Liu and Wright (2000) developed a new joint control system that automatically integrates design, testing and HIL simulation using Simulink/RTW. The user develops a model in Simulink using plant dynamics [35]. Then model runs in Simulink off line and corrections are made accordingly. Once results are satisfactory C code is generated using Real-time Workshop. Next follows the HIL simulation, which helps further tuning of parameters until they are ready for use in production systems.

Temeltas *et al.* (2001) proposed a new HIL simulation system developed to serve the design and control of robot manipulators for any given configuration [31]. The system consists of two direct drive motors – one as the joint actuator and other as the load simulator. The load simulator serves the purpose of generating the actual torque components affecting a particular joint. A simple PID controller is used to achieve a basic tracking performance for the operation of the load simulator. The results show that HIL outputs for position, friction, inertia and acceleration variables of robot links. The HIL experimental setup presented in these thesis differs from the one proposed by Temeltas in several points. First, the system presented here is clearly based on the effort-flow model, which is the rigorous fundamental for HIL. Such a theoretical base was not mentioned in Temeltas paper. Second, the feedback used in this thesis is torque feedback as opposed to

position feedback used in the paper. In addition, the paper mentions both position and torque feedback controllers, which cannot be achieved with a feedback with 1 DOF system. Moreover, the purpose of this thesis is to create a new generic HIL setup, while the system proposed in the paper has very specific purpose and is used for a specific robotics application.

2.3 Automotive Industry

Sailer and Essers (1994) developed a 3- dimensional, nonlinear simulator for the dynamical behavior of trucks. One of the elements of that simulator was an anti-lock braking system integrated as a Hardware-In-The-Loop device for the development of electronic control units (ECU) [26]. Simulations with such a HIL system were highly facilitated by changing the parameters of the template files. This way, a different truck models and a different driving cycles were possible to test. The simulator showed high quality results and was used in industrial applications.

Oueslati *et al.* (1995) worked on limited state active suspensions for articulated heavy vehicles [46]. Control forces, generated by active force generators, were produced using a control scheme based on feedback from road excitation and vehicle response variables. This approach is very similar to the HIL simulation of a vehicle on the simulated road.

Hanselmann (1996) introduced a CACSD Toolset for the HIL simulation testing and integration. Electronic Control Units (ECU) used in vehicles control different components of the vehicle [3]. This toolset allows the use of the component of interest based on a real hardware while other vehicle parts models can be executed on real-time computer systems. The development of the toolset relies on Matlab/Simulink for both prototyping and HIL simulations. The main advantages are: the plant model can be reused for the testing various ECU, all ECU software changes can be easily checked and no extra learning curve is required for the user. The toolset can be used as a tool for engineers to improve efficiency and quality in product development.

Isermann *et al.* (1998) worked on HIL simulations for the design and testing of Engine-control systems [5]. They used two approaches. One was with a transputer and another with a DSP. Tests were conducted on two types of engines – 6 and 8 cylinder diesel

engines. The HIL test bench consisted of a real-time computer system, sensor interface, actuator interface, real injection pumps and the real control unit. The goal was to develop new algorithms and to investigate effects of faults in sensors and actuators and the engine itself. The tests showed comparable results from the HIL simulation with regard to measurements on real diesel engines.

S.Y.Yang *et al.* (1999) developed a simulation platform for the development of an Anti-lock Breaking System (ABS) used in cars [11]. For that purpose, they created an 18 DOF vehicle model and used real hydraulic brake assembly. HIL simulation used hydraulic circuits to include physical parts in the HIL setup. Road surface conditions were created artificially. The results showed a very high degree of similarity, limiting the need for old-fashioned way of the road testing.

Beaumont *et al.* (1998) proposed new algorithms of software development for Electronic Control Units (ECU) used in vehicles [8]. These algorithms were tested on the target hardware in combination with HIL vehicle simulator. Lab-Car was used to develop algorithms, which were tested in two different modes. The first mode enabled the development and testing of ECU before a production of a compatible ECU is available. The second mode allows real ECU to be connected through I/O drivers to the simulator.

ECU development by Reuss and Flamig-Vetter (1998) was made with the purpose of lowering time and cost of the development process [7]. The goal of HIL simulations was to make simulation models and real components work together under realistic operating conditions. The only real part of the HIL setup was the actuator. All other parts were computer simulated. One of the subsystems used to conduct simulation was a diesel injection system with a piezoactuator. The results of the HIL simulations showed that the piezoactuator was able to drive efficiently an injection valve.

Lamberg and Waltermann (2000) showed another approach for developing an Electronic Control Units (ECU) used in vehicles [4]. Their HIL simulation included actual ECU connected to the real time simulation models of the vehicle. This time, ECU did not have only analog and digital interface but it communicated with the sensors and actuators via protocols. During the HIL simulation, software and hardware models simulated the behavior of the vehicle. The real vehicle components were connected via their electrical interfaces to a simulator that reproduced the behavior of real-time environment.

Experiments performed on the HIL system can be reproduced, in this case, precisely and as often as required.

dSPACE Company developed dSPACE Vehicle Dynamics Simulator as commercial product developed on the basis of HIL simulations. Tests can be executed in a virtual test scenario, instead on the road. Real parts of the vehicle communicate with the simulator through I/O processing functions and other functions are simulated within the simulator. The simulator has several digital or analog inputs and outputs, designated for sensors and actuator used in vehicle.

2.4. Military applications

Goldsmith *et al.* (1996) presented the Hardware-In-The-Loop facility of the Air Force Development Test Centre designed to test guided munitions performance [43]. The major HIL components included an infrared (IR) scene generator, IR projector, five axes flight motion simulator (FMS), 6 DOF missile flight simulators and an optomechanical interface to mount the projector onto the FMS.

Sieglinger *et al.*(1996) worked on HIL simulation using direct signal injection (DSI) [44]. This technique was developed for test and evaluation of guided missile seekers. The seeker can be simulated by a projector or, as it is in this case, by a DSI. The sensor hardware was not used. Scene data was provided directly to the seeker signal processor where sensor measurements were computed and simulated. Sensor effects included blurring, sampling, detector response characteristics and noise.

Sanders (1996) presented new techniques for creating the signature models necessary for the infrared texture map generation [45]. The method is part of real-time modeling for HIL to achieve frame rates necessary for missile stimulation. The new modeling method PRISM for model generation of threat vehicles showed greater simulation flexibility and removed the need for expensive data collection.

Huber and Courtney (1997) presented an infrared seeker HIL simulator as a part of Dynamic Infrared Missile Evaluator (DIME) [9]. The HIL testing was performed to verify accuracy of the seeker model. The results were compared to the model prediction. The seeker is a radiometer, which produces voltage and is calibrated by a blackbody and

a collimator. The calibrated seeker produces preamplified output, which is used in HIL testing instead of having to rely on external device to measure the value for each scenario.

DeCesaris and Millner (1997) presented a three-tiered approach to HIL testing [10]. Testing was used to fully characterize system's operational characteristics. The three tiers are the element levels, which include sub-element level testing of critical components, the weapon system level and the family of the systems (FoS) level. The objective of testing each element was to evaluate the performance of elements in a simulated environment. The objective of Weapon system level was to evaluate data exchange between the elements of the weapon system in a simulated operational environment. The purpose of FoS level was to assess interoperability of weapon systems.

Tucker and Zabel (1998) showed the use of a HIL test facility for the purpose of adding physics-based synthetic smoke to prerecorded 2-dimensional infrared imagery [33]. Two computational processes were discussed. The first process centers around the application that has already been implemented and which adds physics-based synthetic smoke to prerecorded 2-dimensional infrared imagery. The second computation process focuses on real time pixel intensity correction. The HIL simulation setup has a missile seeker, which is set inside of simulated environment and operates just as it would during an actual flight. The simulated environment is changing computer-generated images on the screen and the response of the system is obtained.

2.5. Other HIL simulations

P. Schafer (1992) presented HIL simulations of multi-body system models with transputers [13]. The multi-body system was modeled as a system of bodies connected by spring-damper muscles or as actively controlled elements with holonomic and scrclonomic constraints. Transputers are special microprocessors with distributed memories that allow parallel processing, in real time and make available peripheral processes for control devices.

Ptak and Foundy (1998) worked on integration of a flight hardware with real-time simulations to increase attitude control systems (ACS) accuracy by providing greater test

coverage through end-to-end testing in a realistic test environment [12]. They used real-time software architecture RT-Linux. The environment included RT-Linux computers and Matrix-X control system design automation tools for the real-time simulation and the HIL testing.

Lucas *et al.* (2000) introduced remote simulation to evaluate real-time traffic control strategies [34]. One of the simulation methods used was a HIL simulation with the purpose to show the migration of the real-time traffic-adaptive control logic onto the traffic controller hardware itself. The simulations were tested on several different types of networks and were found to perform in the same manner as its local counterpart.

Ueno and Okada (2000) worked on a control of motor bearings and they proposed an experimental setup consisting of bearing stator and rotor held on two radial ball bearings [32]. A DC motor generator was used as a torque generator. The concept of simulating the load by a motor falls into HIL simulation category.

3. THEORETICAL FRAMEWORK

3.1 Effort-flow concept

Mechatronic systems, in general, are mixed systems consisting of mechanical, electrical and computer components. Mechanical components usually represent a system or a part in which one type of energy (mechanical or electrical) is converted into mechanical work and where a main process takes place. Electrical components are used as actuators (motors) which convert electrical energy into mechanical work (rotation). Another use for electrical components would be as transducers, which convert the mechanical energy transferred from the plant into electrical signal. Finally, electrical components are used as signal conditioning devices and drivers. As signal conditioning devices, those components are electrical circuits that are placed between transducers and a computer with the purpose of changing the signal amplitude for easier conversion of the signal inside of the computer. Electrical components, used as drivers, are modulating power supply needed to drive the actuators in accordance to the command signals coming from computer and that way generating power signal. Computer components, Analog to Digital Converters, Digital to Analog Converters are used to receive conditioned signal and convert it into computer readable code. Such a code is the base for calculation done by software, which prepares the control signals to be sent back to the system.
As seen, all components of the mechatronic system have one common characteristic and that is the change of the energy type from one to another [1].

Each particular component from such system has input, output and state defined by state variables. The interface between components can be signal or power transmission variables, depending on the place and the role of the particular component.

Input and output of the system are signal or power transmission variables depending on what kind of transformation takes place inside the component.

It is very important to notice that the communication between components of the system is based on cut variables, input or output variables. To preserve the law of the energy conservation, the power balance has to be respected at the connections of the objects [1]. That means that if we presume that there is no power loss during the transformation inside of the component, input and output cuts correspond to the same amount of the power.

For variety of mechatronics components, the power is calculated in a different ways. For mechanical components, the power is the product of a force and a velocity or a moment and an angular velocity depending if a mechanical system is moving linearly or rotating. The power is calculated for electrical systems by multiplying a voltage and a current and for hydraulic systems by multiplying a pressure and a flow rate. In general, the power is the result of product of two different variables, called effort and flow variables, at the connection of the components (either on input or output side) as a result of an effort-flow cut.

Power is transferable and various components can be linked together by knowing the information about their effort-flow cuts. For us, the important information is on input and output side of the component, while what is happening inside of the component is less important. That concept is well known in computer science as Object-Oriented Approach [1].

Each component of a mechatronics system can be seen as an object. Objects communicate by message passing. In the case of mechatronics, object messages are values of effort-flow variables on input and output side. The objects inside consist of private data variables and methods, in the case of computer objects, while mechatronics objects have transfer functions with the same purpose of transferring input variables into output ones.

Mechatronic systems are, using this analogy, a set of mechatronic objects with known effort-flow cuts. Further analysis shows that some other object (component), which has the same effort-flow cuts, on both, input and output sides can replace each of these original objects (components).

The Hardware-In-The-Loop (HIL) simulation uses the concept of replacing components with the same effort-flow cut as the base for the method. Each component is modeled as a module with effort-flow variables used for its input and output side. The modules are assembled together into a greater system model. Since every module has its own model with recognizable inputs and outputs, it is easy to replace it with similar modules and again obtain a model for the whole system. This way, not only the physical system has modular characteristics, but also models are easily interchangeable. More important to notice, HIL simulations are different from old fashion ways of testing since the experimental setup does not have to have all real physical components available. Computer models could simulate some components while others are real. Having the model of the system written in such a way that it is an array of different modules, it is graphically easy to choose modules to be simulated by computer and those that will be presented by a real hardware, as well as the connection between them.

3.2. General thesis model

The main goal of this thesis is to show that effort-flow cuts, and principles of module interchangeability used in HIL simulation, can be employed for a rapid testing of different types of loads that are not available in physical form at the present time.
It is simpler to say that computer simulation of unavailable parts could be a solution, but knowing that computer simulations are based on mathematical models usually generated using approximation and linearization and neglecting sometimes very important characteristics of the component, a more realistic load simulation has to be considered. The basic idea behind the proposed experimental setup is the use of two DC motors; one of them representing an actuator or a device, which is driving the load, while the second DC motor represents an active load. An active load means that, in principle, the load DC motor can physically simulate any load for which a precise analytical model is available. In another words, the simulated load should produce the same output, e.g. the same velocity characteristic, or the same torque characteristic as in the case of a real load. This is possible, since the effort-flow variables for real and simulated load remain the same.

The load that will be considered for the proof of this concept is an inertial-viscous load. In addition, some of loads will be linear while the others will be nonlinear.

3.3 Modeling

Physical simulation using the experimental setup requires much deeper analysis than simple modeling from previous examples, which were presented only as illustrations of the concept. As mentioned before, the experimental setup was built as a modular testbed with two places for motors. Each motor place has two holders with two different sizes for holes, which means that different kinds of motors could be mounted on the setup base plate. The shaft that connects motors can be of different length depending on which holders are used for the motor. Since one of the motors represents the active load, it is readily replaceable with real loads. Looking that way, there are two different setups to be modeled. One consists of two motors, the actuator and the load, connected by a shaft and the second consists of the same actuator-motor and any real load to be connected to the actuator by the same shaft. Since these models are going to be tested together, the modeling will be done in parallel. The difference between the setups is only in the model of the load. Following the effort-flow principles that should be an easy task, in which we have to consider different inside modules, while the input output variables should remain the same. Figure 3.5 shows the general scheme of the setups that will be modeled in parallel.

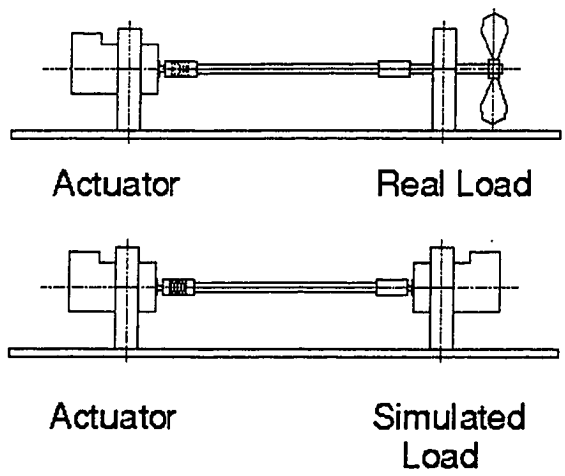


Figure 3.1 Real and simulated load setups

Before the modeling starts, it is important to describe the system to be modeled. The desirable characteristic of these systems is their linearity. A system is called linear if the principle of superposition applies [19]. If the system is linear then its input-output relation is linear as well. Chapter 4 contains experimental verification that the system which is about to be modeled is linear. Of course, no system is precisely linear; we can only approximate its behavior using mathematical models. Every system that shows linearly dependant output for any input, within certain error interval, can be modeled as a linear system. DC motors, which are part of this experimental setup, contain some nonlinear behaviors, which cannot be neglected in the model. A dry friction is a significant form of nonlinearity.

In general, DC motor friction can be modeled with three components [36], viscous friction, Coulomb friction and stiction. The last two can be combined into one called dry friction. Figure 3.2 shows the form of those friction forms as a function of the velocity.

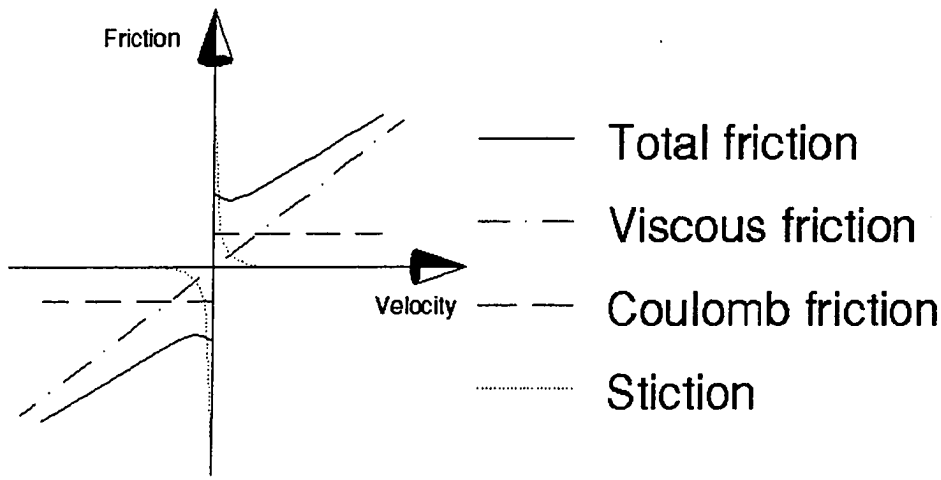


Figure 3.2 Friction model for a DC motor

3.3.1 Model of the experimental setup with linear inertial-viscous load

To be able to physically model properly using the experimental setup it is important to model each particular module of the setup using effort-flow concept. For a compact presentation, matrix form will be used. In general, we have to model the actuator, the shaft and the load taking in consideration their inputs and outputs. The actuator, in our case, a permanent magnet DC motor (Pittman 9232) [54], will not change at all. The shaft, which connects the actuator and the load, could be changed, and in this case, there are two shafts that are part of the setup; one is simple stiff linear shaft and another is built with a spring in the middle, which brings non-linearity to the system. The load is the part of the setup that will be changed most often during the experimental procedure. Different kinds of the loads will be considered, including linear and nonlinear loads as well as real and simulated loads. Simulated loads will be generated using an active load, a second permanent magnet DC motor, controlled by a PC based model such that it produces the required reaction torque.

The figure 3.3 shows a model with a linear inertial-viscous load.

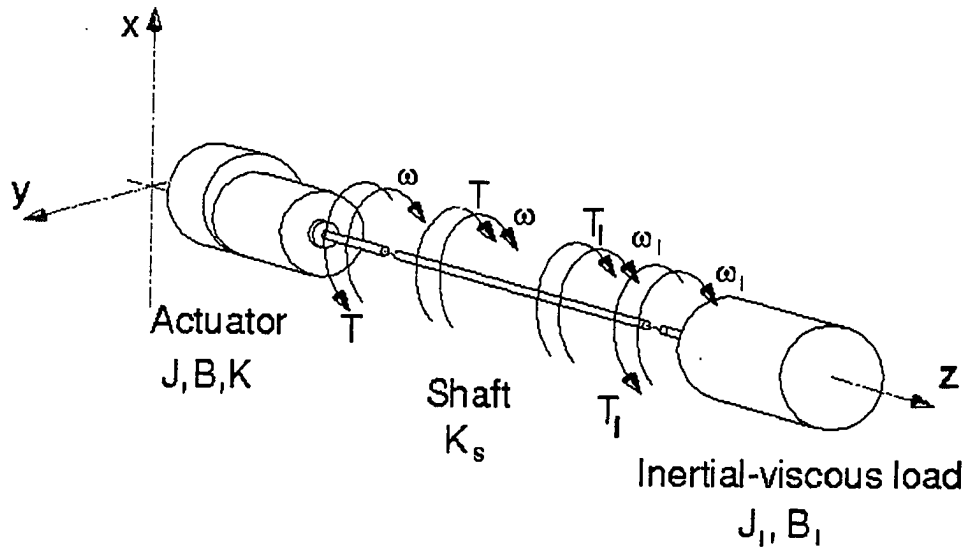


Figure 3.3 The model with a linear inertial-viscous load

In general, a PM DC motor can be described by this set of equations [1]

$$\begin{aligned} K*i &= J*\frac{d\omega}{dt} + B*\omega + T \\ U &= L*\frac{di}{dt} + R*i + K*\omega \end{aligned} \tag{3.1}$$

After applying Laplace transform for zero initial conditions:

$$\begin{aligned} K*i(s) &= (J*s+B)*\omega(s) + T(s) \\ U &= (L*s+R)*i(s) + K*\omega(s) \end{aligned} \tag{3.2}$$

A PM DC motor has two cuts, one mechanical and the other electrical. For the mechanical cut, a power flow is determined by a torque T and a velocity ω, while for the electrical cut a power flow is characterized by a voltage U and a current i. By manipulating set of equations 3.2 and using matrix notation, we can get the model of the actuator in the matrix form.

$$\begin{bmatrix} T(s) \\ \omega(s) \end{bmatrix} = \begin{bmatrix} -(J*s+B)/K & K+(J*s+B)(L*s+R)/K \\ 1/K & -(L*s+R)/K \end{bmatrix} \begin{bmatrix} U(s) \\ i(s) \end{bmatrix} \tag{3.3}$$

The shaft model is easy to obtain from the free body diagram

$$T(s) = -T_l(s) = \frac{K_s}{s} * (\omega(s) - \omega_l(s)) \tag{3.4}$$

or using matrix notation

$$\begin{bmatrix} T_l(s) \\ \omega_l(s) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -s/K_s & 1 \end{bmatrix} \begin{bmatrix} T(s) \\ \omega(s) \end{bmatrix} \tag{3.5}$$

The actuator and the shaft models, as presented, are ready to be used in final composition for each particular setup. The load is the critical part of this derivation. Before starting modeling the load, it is important to mention that the feedback from the system will be obtained at the actuator side of the setup. Looking at the model of the shaft it is easy to notice that due to the flexible shafts, the velocities of the load and the actuator are not the same. An equation is required to describe the dependence between these two velocities.

Let us presume that the load is of the form

$$T_l(s) = (J_l * s + B_l) * \omega_l(s) \tag{3.6}$$

which is the basic form of any linear inertial-viscous load. Knowing that

$$T_l(s) = -\frac{K_s}{s} * (\omega(s) - \omega_l(s))$$

we can derive the following equation

$$\omega_l(s) = \frac{K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) \tag{3.7}$$

This equation 3.7 will be used every time when the form of the load is as described in equation 3.6 to have all dynamics relations function of one velocity.

To obtain the load model we need first to choose the type of the load to model. First, let us consider a simple linear inertial-viscous load, which can be described by equation 3.6

$T_l(s) = (J_l * s + B_l) * \omega_l(s)$.

Using the matrix notation, the following model of the load is obtained.

$$\begin{bmatrix} 0 \\ \omega_l(s) \end{bmatrix} = \begin{bmatrix} 1/J_l * s + B_l & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_l(s) \\ \omega_l(s) \end{bmatrix} \tag{3.8}$$

If we consider the equation 3.7 and substitute it in the 3.8, this represents the final model of the inertial-viscous load.

$$\begin{bmatrix} 0 \\ \omega(s) \end{bmatrix} = \begin{bmatrix} 1/J_l * s + B_l & 1 \\ 0 & J_l * s^2 + B_l * s + K_s / K_s \end{bmatrix} \begin{bmatrix} T_l(s) \\ \omega_l(s) \end{bmatrix} \tag{3.9}$$

All modules of the general model are derived and we can compose the complete model for the inertial-viscous load.

$$\begin{bmatrix} 0 \\ \omega(s) \end{bmatrix} = \begin{bmatrix} 1/J_l * s + B_l & 1 \\ 0 & J_l * s^2 + B_l * s + K_s / K_s \end{bmatrix} \begin{bmatrix} -1 & 0 \\ -s/K_s & 1 \end{bmatrix} \begin{bmatrix} -(J*s+B)/K & K + (J*s+B)(L*s+R)/K \\ 1/K & -(L*s+R)/K \end{bmatrix} \begin{bmatrix} U(s) \\ i(s) \end{bmatrix} \tag{3.10}$$

This series model was built as an array of modules and has two inputs U and i and one output ω. By modeling each module, we can obtain, from the matrix model, two scalar equations, for the overall system. The system can have a voltage control or a current control of the output velocity ω.

3.3.2 Model of experimental setup with non-linear inertial-viscous load

Compared to the model with the linear load, the model of the setup with nonlinear load differs only in the model of the load module. Models of the actuator and the shaft will remain the same. Figure 3.4 shows the model with nonlinear inertial-viscous load.

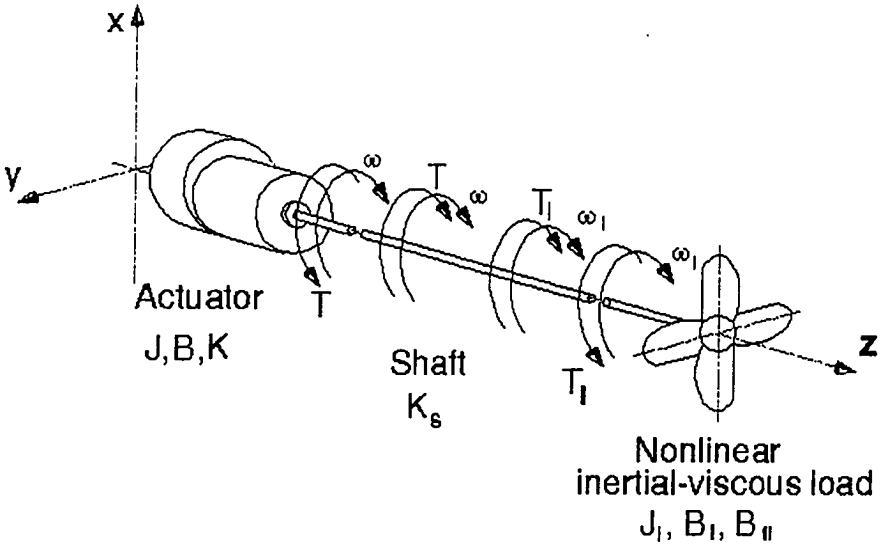


Figure 3.4 The model with a nonlinear inertial-viscous load.

A fan is a typical nonlinear inertial-viscous load. It can be modeled as [55]:

$$T_l = J_l * \frac{d\omega_l}{dt} + B_l * \omega_l + B_{fl} * \omega_l^2 \qquad (3.11)$$

The last element in the equation is a nonlinear term, which is common for fans and pumps and describes the (blowing) pumping effect. To be able to apply the Laplace transformation, it is necessary to linearize the model.

Every stable system reaches steady state of operation after transient response. The operating point in the steady state is often chosen as the reference for the linearization. If the deviations of the variables around the operating point are small, it is possible to use the Taylor series to linearize the model. For the load described in 3.11 let us presume that steady state operating conditions are reached for $\omega = \omega_{op}$.The torque, for the steady state can be calculated as

$$T_l(\omega_{op}) = T_{lop} = B_l * \omega_{lop} + B_{fl} * \omega_{lop}^2 \qquad (3.12)$$

since $\frac{d\omega_l}{dt} = 0$ for steady state.

The linearization of the equation 3.12 can be made either for the case of an operating point at zero steady state velocity or for the case when the steady state velocity of the system is different from zero. In our case, the velocity of the system can not be predicted and it will vary in the ranges bigger than linearization would allow. To be able to model the system with nonlinear load we will use encapsulation method, well known in software engineering. As long as we maintain the effort-flow input output variables, it is possible to continue with the derivation.

A nonlinear load will be assumed attached to the shaft and the corresponding cut for the system. At that point the effort-flow variables of the system, from the shaft side, are T_l and ω_l. On the side of the nonlinear load, at the same cut, we should maintain the same variables to allow connectivity with the rest of the system.

The model of the nonlinear load will be represented with a block containing the set of nonlinear equations. The solution of the set of equations is derived for the model of a linear load for the cut variable. In this case they are ω, velocity of the system and the torque T. The input variables of the block will be T_l and ω_l, which are outputs for the shaft. Inside the block, there will be variables important only for that module, invisible

for the outside system. Those variables are known as private variables in object oriented modeling. They are inertias and viscosities of the nonlinear load module.

The block representing the nonlinear load model, with effort flow inputs and outputs, contains following nonlinear differential equations

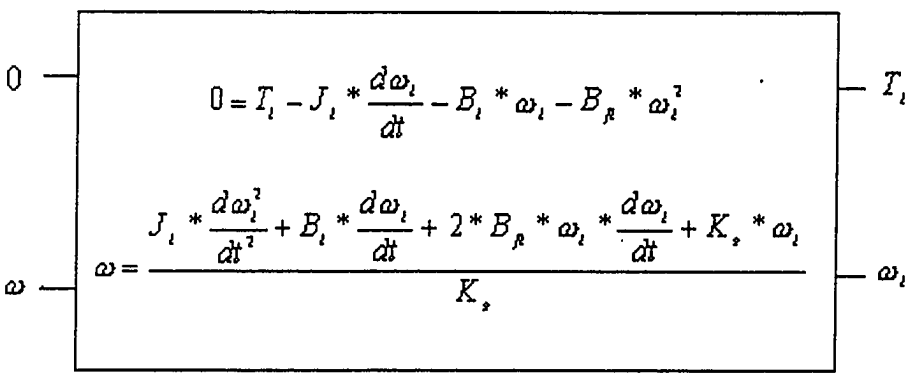
$$0 = T_l - J_l * \frac{d\omega_l}{dt} - B_l * \omega_l - B_{fl} * \omega_l^2 \tag{3.13}$$

and

$$\omega = \frac{J_l * \frac{d\omega_l^2}{dt^2} + B_l * \frac{d\omega_l}{dt} + 2 * B_{fl} * \omega_l * \frac{d\omega_l}{dt} + K_s * \omega_l}{K_s} \tag{3.14}$$

The equation 3.14 is modified equation 3.7 in time domain, with added member for fan viscosity B_{fl} and represents the relationship between load and actuators angular velocities. J_l, B_l are private variables of the system and they are not recognizable by any other module.

The module block can be presented as follows



(3.15)

Knowing that the equation set 3.15 maintains the basic effort-flow connectivity, it could be considered as any other module in the system. HIL systems are based on interchangeability of the modules and as long as effort flow cuts are compatible, the system is valid.

When combined with the actuator and the shaft modules, the complete model for fan real load is represented by:

$$\begin{bmatrix} 0 \\ \omega(s) \end{bmatrix} = [Nonlinear \quad block] \begin{bmatrix} -1 & 0 \\ -s/K_s & 1 \end{bmatrix} \tag{3.16}$$

$$\begin{bmatrix} -(J*s+B)/K & K+(J*s+B)(L*s+R)/K \\ 1/K & -(L*s+R)/K \end{bmatrix} \begin{bmatrix} U(s) \\ i(s) \end{bmatrix}$$

The equation 3.16 represents the model of nonlinear load connected to the actuator by shaft. One of the major advantages coming from the HIL method presented in this thesis is the possibility of physically simulating nonlinear loads, making it much simpler than using digital simulations. While digital simulations require mathematical solvers for sets of nonlinear differential equations to be able to simulate loads, the HIL method is simpler and requires only simple model, which is easy to implement. The following paragraphs show the derivation of those models.

3.3.3 Model of experimental setup with physically simulated load

The figure 3.5 shows the model for the experimental setup with the physical load simulator.

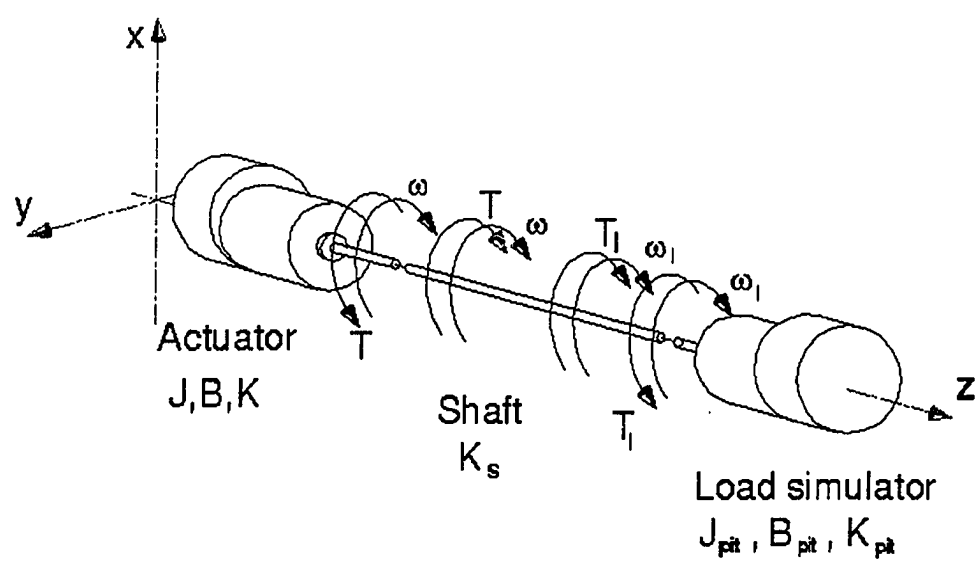


Figure 3.5 The model of the setup with a physical load simulator

The real load, linear and nonlinear, can be simulated by using an active load. Simulating the load means that under any input given by a control system and conveyed by the actuator, the simulated load has to generate the same output as if it was the original real load. The experimental setup designed for thesis has a load simulator using a PM DC motor, in particular the Pittman 9232. The motor itself can act as active or passive load depending on current input. If there is no exogenous torque command sent to the motor, then motor acts as linear passive inertial-viscous load with nonlinear characteristics at zero velocity state. If an exogenous torque command is sent to the motor, then motor becomes active load with additional controllable torque.

The value of the endogenous controllable (simulating) torque is of interest here and will be derived for two cases, one for linear inertial-viscous load and second for nonlinear inertial-viscous load

The figure 3.6 shows the equilibrium conditions for the cut between the shaft and the load simulator when the simulator is simulating a linear inertial-viscous load.

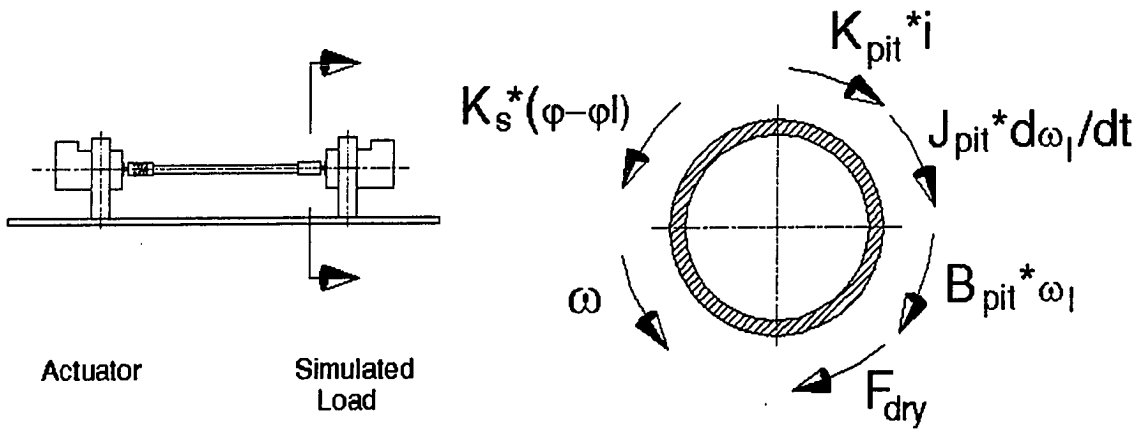


Figure 3.6 Free body diagrams for the simulation of a linear load

Using D'Alembert principle, we can obtain the following equation for the cut.

$$K_s * (\varphi - \varphi_l) = J_{pit} * \frac{d\omega_l}{dt} + B_{pit} * \omega_l + F_{drypit} + K_{pit} * i \qquad (3.17)$$

This equation shows the moments that are opposing the moment conveyed from the actuator over the length of the shaft. J_{pit} and B_{pit} are rotor inertia and coefficient of viscous friction, respectively. Each of them could be written in a different way considering that the simulation of a different load is to be achieved with a Pittman motor as a load simulator. There are two possibilities. The first one is that the real load parameters J_l and B_l are larger than the load of the simulator acting as passive load, and the second is that the real load is smaller. We will consider the first case for the analysis, while the second option would differ from the first one due to the opposite sign. In the first case:

$$J_{pit} = J_l - J_{sim} \tag{3.18}$$

$$B_{pit} = B_l - B_{sim} \tag{3.19}$$

where J_{sim} and B_{sim} represent an inertia and a viscosity added to the Pittman motor inertia and viscosity to make the simulator load behave as the real load.

$K_s * (\varphi - \varphi_l)$ is a moment generated as a result of difference in the angular rotation on opposite ends of the shaft, which is the load to the actuator and driving moment for the active load. F_{drypit} is the dry friction torque of the load simulator. Again, it can be written in a different way by trying to include real load and simulation difference.

$$F_{drypit} = F_{dryl} - F_{drysim} \tag{3.20}$$

$K_{pit} * i$ is the simulation moment and represents the value that has to be added (subtracted) to the load simulator to achieve output that corresponds to the output of the real load. In another words, this is the moment generated by the control signal from computer making the motor active load. We can rewrite equation 3.17 using the identities 3.18, 3.19 and 3.20. By applying Laplace transform, we obtain

$$\frac{K_s}{s} * (\omega(s) - \omega_l(s)) = [(J_l - J_{sim}) * s + (B_l - B_{sim})] * \omega_l(s) + F_{dryl} - F_{drysim} + K_{pit} * i(s) \tag{3.21}$$

After rearranging,

$$K_{pit} * i = \frac{K_s}{s} * \omega(s) - [(J_l - J_{sim}) * s + (B_l - B_{sim}) + \frac{K_s}{s}] * \omega_l(s) - F_{dryl} + F_{drysim}$$

or

$$K_{pit} * i = \frac{K_s}{s} * \omega(s) - (J_l * s + B_l + \frac{K_s}{s}) * \omega_l(s) + (J_{sim} * s + B_{sim}) * \omega_l(s) - F_{dryl} + F_{drysim}$$

Using equation 3.7 for the relationship between velocities and substituting it into the equation, we can write:

$$K_{pit} * i = \frac{K_s}{s} * \omega(s) - (J_l * s + B_l + \frac{K_s}{s}) * \frac{K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) +$$

$$(J_{sim} * s + B_{sim}) * \frac{K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) - F_{dryl} + F_{drysim}$$

It can be seen that the first two factors on the right hand side of the equation cancel each other.

$$K_{pit} * i(s) = \frac{(J_{sim} * s + B_{sim}) * K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) - F_{dry} \tag{3.22}$$

This equation represents the way of calculating moment that can be used to simulate any linear inertial-viscous load with PM DC motor load simulator. F_{dry} component in the equation represents the difference between dry friction of load and simulator. If the real load does not have a dry friction component and if we are not simulating another motor, then this factor is equal to the dry friction of the load simulator.

To obtain the load module model using matrix notation we have to use the equation 3.17 and say that $K_s * (\varphi - \varphi_l)$ actually represents the moment output from the shaft module $T_l(s)$.

$$K_s * (\varphi - \varphi_l) = T_l(s) \tag{3.23}$$

In this case, the equation 3.17 changes to

$$T_l(s) = J_{pit} * \frac{d\omega_l}{dt} + B_{pit} * \omega_l + F_{drypit} + K_{pit} * i \tag{3.24}$$

Using the Laplace transform and substituting the value for $K_{pit} * i$ obtained in 3.22 we can write

$$T_l(s) = (J_{pit} * s + B_{pit}) * \omega_l(s) + F_{drysim} + \frac{(J_{sim} * s + B_{sim}) * K_s}{J_{pit} * s^2 + B_{pit} * s + K_s} * \omega(s)$$

and using 3.7 for the velocity relationship, final torque on the far end of the shaft is:

$$T_l(s) = \frac{(J_{pit} * s + B_{pit}) * K_s}{J_{pit} * s^2 + B_{pit} * s + K_s} * \omega(s) + F_{drysim} + \frac{(J_{sim} * s + B_{sim}) * K_s}{J_{pit} * s^2 + B_{pit} * s + K_s} * \omega(s)$$

or rearranged

$$T_l(s) = \frac{[(J_{pit} + J_{sim}) * s + (B_{pit} + B_{sim})] * K_s}{J_{pit} * s^2 + B_{pit} * s + K_s} * \omega(s) + F_{drysim} \tag{3.25}$$

Using equations 3.25 and 3.7, we can write matrix model for the simulator when it is simulating a linear load.

$$\begin{bmatrix} 0 \\ \omega(s) \end{bmatrix} = \begin{bmatrix} \dfrac{J_{pit} * s^2 + B_{pit} * s + K_s}{[(J_{pit} + J_{sim}) * s + (B_{pit} + B_{sim})] * K_s} & \dfrac{F_{drysim} * (J_{pit} * s^2 + B_{pit} * s + K_s)}{\omega_l(s) * [(J_{pit} + J_{sim}) * s + (B_{pit} + B_{sim})] * K_s} \\ 0 & J_{pit} * s^2 + B_{pit} * s + K_s / K_s \end{bmatrix} \begin{bmatrix} T_l(s) \\ \omega_l(s) \end{bmatrix}$$

or

$$\begin{bmatrix} 0 \\ \omega(s) \end{bmatrix} = \frac{J_{pit} * s^2 + B_{pit} * s + K_s}{K_s} * \begin{bmatrix} \dfrac{1}{(J_{pit} + J_{sim}) * s + (B_{pit} + B_{sim})} & \dfrac{F_{drysim}}{\omega_l(s) * [(J_{pit} + J_{sim}) * s + (B_{pit} + B_{sim})]} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_l(s) \\ \omega_l(s) \end{bmatrix} \tag{3.26}$$

At this point, we can write complete model for the Hardware-In-The-Loop simulation method of unknown linear load that is using two PM DC motors.

$$\begin{bmatrix} 0 \\ \omega(s) \end{bmatrix} = \frac{J_{pit} * s^2 + B_{pit} * s + K_s}{K_s} * \begin{bmatrix} \dfrac{1}{(J_{pit} + J_{sim}) * s + (B_{pit} + B_{sim})} & \dfrac{F_{drysim}}{\omega_l(s) * [(J_{pit} + J_{sim}) * s + (B_{pit} + B_{sim})} \\ 0 & 1 \end{bmatrix} \tag{3.27}$$

$$\begin{bmatrix} -1 & 0 \\ -s/K_s & 1 \end{bmatrix} \begin{bmatrix} -(J * s + B)/K & K + (J * s + B)(L * s + R)/K \\ 1/K & -(L * s + R)/K \end{bmatrix} \begin{bmatrix} U(s) \\ i(s) \end{bmatrix}$$

The second case is for a nonlinear inertial-viscous load. Figure 3.6 can be used again, since the same load simulator will be used to simulate a nonlinear load. The difference will occur in the description of particular factors of the equation.

The equation 3.17 can be used again to start the derivation. We are interested in finding the value of the additional moment, which would be used by the load simulator in order to simulate unknown nonlinear load.

$$K_s * (\varphi - \varphi_l) = J_{pit} * \frac{d\omega_l}{dt} + B_{pit} * \omega_l + F_{drypit} + K_{pit} * i$$

To be consistent with previous derivation, real load parameters will be considered larger than the load of the load simulator when used as passive load. In the case of the nonlinear load, we can write:

$$J_{pit} = J_l - J_{sim} \tag{3.28}$$

$$B_{pit} = B_l - B_{sim} \tag{3.29}$$

$$B_{fpit} = B_{fl} - B_{fsim} = 0 \tag{3.29a}$$

$$F_{drypit} = F_{dryl} - F_{drysim} \tag{3.29b}$$

The relation 3.29a represents a virtually inserted formula member. It really does not exist physically in the simulator, but the nonlinear load has it. It is needed here for pure mathematical derivation purposes. Since it does not exist in the simulator free body cut it is equalized to zero.

The other relations are the same as in the previous derivation for linear load. The non-linearity in the load does not allow the use of the Laplace transform, so the rest of the derivation will be continued in the time domain, using differential equations.

$$K_s * (\int \omega * dt - \int \omega_l * dt) = (J_l - J_{sim}) * \frac{d\omega_l}{dt} + (B_l - B_{sim}) * \omega_l + (B_{fl} - B_{fsim}) * \omega_l^2 + F_{dryl} - F_{drysim} + K_{pit} * i(s) \tag{3.30}$$

The following is rearranging of the equation 3.30

$$K_{pit} * i = K_s * \int \omega * dt - (J_l - J_{sim}) * \frac{d\omega_l}{dt} - (B_l - B_{sim}) * \omega_l - (B_{fl} - B_{fsim}) * \omega_l^2 - K_s * \int \omega_l * dt - F_{dryl} + F_{drysim}$$

$$K_{pit}*i=K_s*\int\omega*dt-(J_l*\frac{d\omega_l}{dt}+B_l*\omega_l+B_{fl}*\omega_l^2+K_s\int\omega_l*dt)+(J_{sim}*\frac{d\omega_l}{dt}+B_{sim}*\omega_l+B_{fsim}*\omega_l^2)-F_{dryl}+F_{drysu}$$

(3.30a)

The equation 3.14 is a relation between velocities on the different ends of the shaft for a nonlinear load and it can be used in the next step. Similarly to the derivation of equations for linear load we can presume that the first two factors on the right hand side will cancel each other. In that case, we can write

$$K_s * \int\omega * dt = J_l * \frac{d\omega_l}{dt} + B_l * \omega_l + B_{fl} * \omega_l^2 + K_s * \int\omega_l * dt$$

If we take the derivative of the previous expression, we can write

$$K_s * \omega = J_l * \frac{d^2\omega}{dt^2} + B_l * \frac{d\omega_l}{dt} + B_{fl} * \frac{d\omega_l^2}{dt} + K_s * \omega_l \qquad (3.30b)$$

Now if we consider that

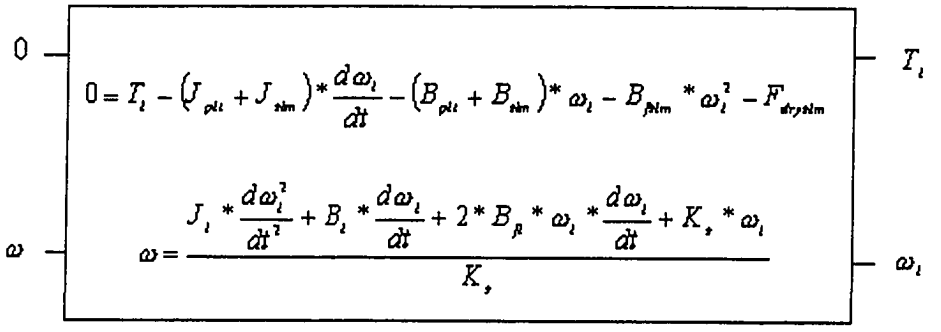
$$B_{fl} * \frac{d\omega_l^2}{dt} = 2 * B_{fl} * \omega_l * \frac{d\omega_l}{dt}$$

and if we substitute the equation 3.14 into left hand side of 3.30b we can see that equation holds.

After cancellation of the first two factors in the right hand side of the equation 3.30a and using 3.14, we will obtain two differential equations for the following block containing the equation

$$K_{pit} * i = J_{sim} * \frac{d\omega_l}{dt} + B_{sim} * \omega_l + B_{fsim} * \omega_l^2 - F_{dry} \qquad (3.31)$$

It represents the moment needed to add to the load simulator to be able to simulate nonlinear load using PM DC motor as a load simulator. Following the similar procedure as for the linear load, we can obtain model of the load simulator module written in the form of the encapsulated block. The block is nonlinear and contains the set of nonlinear differential equations, maintaining the effort flow connectivity with the rest of the system. In that case, the torque on the end of the shaft connected to the load would be:



(3.32)

Using the block equation 3.32, which represents an nonlinear encapsulated block with effort-flow cut connections, and by incorporating that module into the general model for the HIL we can write

$$\begin{bmatrix} 0 \\ \omega(s) \end{bmatrix} = [Nonlinear \quad Block]$$
$$\begin{bmatrix} -1 & 0 \\ -s/K_s & 1 \end{bmatrix} \begin{bmatrix} -(J*s+B)/K & K+(J*s+B)(L*s+R)/K \\ 1/K & -(L*s+R)/K \end{bmatrix} \begin{bmatrix} U(s) \\ i(s) \end{bmatrix} \tag{3.33}$$

Equation 3.33 represents the complete model for the Hardware-In-The-Loop simulation of unknown nonlinear load that is using two PM DC motors.

LOAD TYPE	CASE	SIMULATING TORQUE
LINEAR	L real > L sim.	$\frac{(J_{sim} * s + B_{sim}) * K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) - F_{dry}$
	L real = L sim.	0
	L real < L sim.	$-\frac{(J_{sim} * s + B_{sim}) * K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) + F_{dry}$
NONLINEAR	L real > L sim.	$J_{sim} * \frac{d\omega_l}{dt} + B_l * \omega_l + B_{fsim} * \omega_l^2 - F_{dry}$ $\omega = \frac{J_l * \frac{d\omega_l^2}{dt^2} + B_l * \frac{d\omega_l}{dt} + 2 * B_{fl} * \omega_l * \frac{d\omega_l}{dt} + K_s * \omega_l}{K_s}$
	L real = L sim.	$B_{fsim} * \omega_l^2$ $\omega = \frac{J_l * \frac{d\omega_l^2}{dt^2} + B_l * \frac{d\omega_l}{dt} + 2 * B_{fl} * \omega_l * \frac{d\omega_l}{dt} + K_s * \omega_l}{K_s}$
	L real < L sim.	$-J_{sim} * \frac{d\omega_l}{dt} - B_l * \omega_l - B_{fsim} * \omega_l^2 + F_{dry}$ $\omega = \frac{J_l * \frac{d\omega_l^2}{dt^2} + B_l * \frac{d\omega_l}{dt} + 2 * B_{fl} * \omega_l * \frac{d\omega_l}{dt} + K_s * \omega_l}{K_s}$

Table 3.1 Simulation moments

Models for HIL simulations using two PM DC motors, for two basic linear and nonlinear cases, are obtained and their corresponding simulation moments. Table 3.1 summarizes simulation moments for each particular case.

While for a linear case simulation moments are presented as moments solved straight forward for the reference system's velocity, the nonlinear case contains pairs of equations from the encapsulated block. Either way, the equations from the table are ready for the software implementation.

3.4 Block diagrams

The Hardware-In-The-Loop system has been modeled for three cases. The first case was for real linear inertial-viscous load, the second for nonlinear inertial-viscous load and the third for load simulator. All these systems can be presented in a block diagram form, which clearly shows modules and signal flow.

3.4.1 Linear inertial-viscous load

The following figure 3.7 shows block diagram for the system with linear inertial-viscous load.

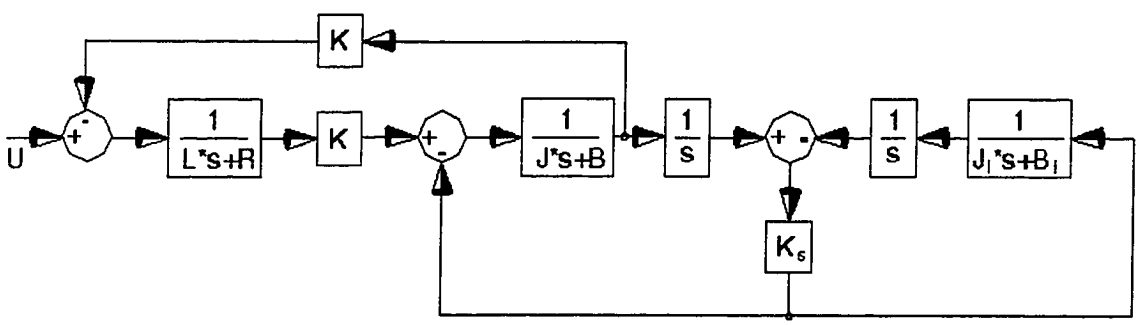


Figure 3.7 Block diagram of the system with linear inertial-viscous load

3.4.2 Nonlinear inertial-viscous load

The figure 3.8 shows block diagram for the system with nonlinear inertial-viscous load.

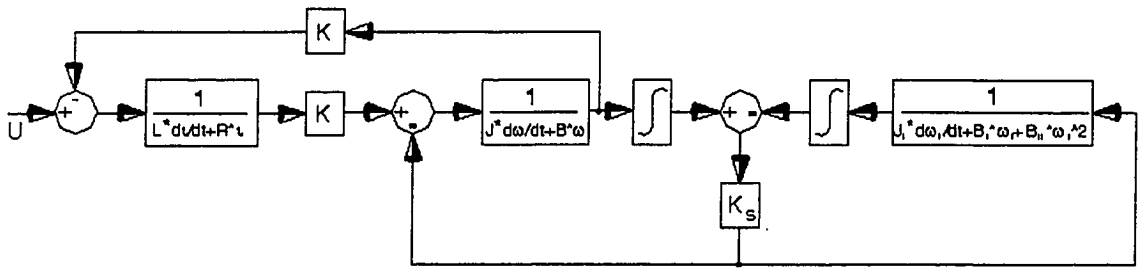


Figure 3.8 Block diagram of the system with nonlinear inertia-viscous load

3.4.3 Physical load simulator

Figure 3.9 shows the block diagram of the system for the load simulator there are two inputs to the system. One is to the actuator, while second one is to the active load. The sign +/- is because simulator can simulate loads that are smaller or larger of the simulator's passive dynamics.

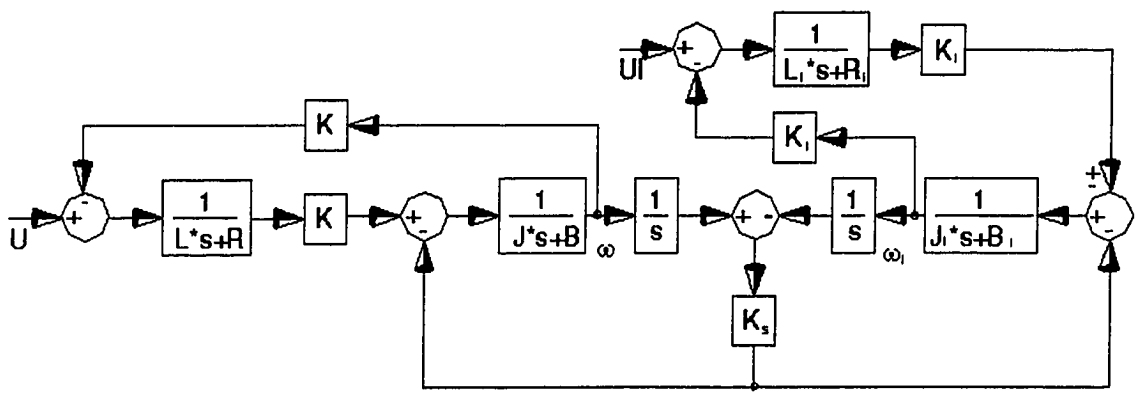


Figure 3.9 Block diagram of the system with physical load simulator

4. DESCRIPTION OF THE EXPERIMENTAL SETUP

4.1. Introduction

The experimental setup used for this research was based on the main idea to create a very new experimental setup with a capability to perform Hardware-In-The-Loop (HIL) simulations. Since HIL is a specific method and requires specific features, during the process of design, several guiding requirements were considered. They were:

-modularity of design;
-easy and fast exchange of modules for quick experimental resetting;
-use of already available parts from the lab;
-generic design, easy to modify, useful for the future research;
-real time control and monitoring system;
-adaptable to different control systems;
-simple design for signal conditioning and protection circuits;
-low cost.

4.2. History of the development

The very first idea of the experimental setup was based on two motors connected with a shaft. During that period of research, the components that were used were changed to suit improvement of a design, but the idea and concept stayed the same from the first day. One of the motors was designated as an actuator while the second motor was used as

a physical load simulator. The load simulator had to be easily removable since the nature of the experiments required frequent change between real loads and physical load simulator. The actuator motor was the only part considered not to be changed, even though the holders for the actuator allow the use of a different motor. The following is the review of the initial choice of parts considered for the setup and the reasons behind rejecting some of them as the part of the setup.

4.2.1 Main setup

As mentioned before the idea was to create the setup consists of two motors connected with a shaft. At the beginning of the process, we had two different motors available in the lab and they were first candidates for the setup.

NSK direct drive motors were considered as a good choice at the beginning, since they have very high torque, they could be controlled easily and precisely and they are brushless motors [51]. The problem was to create a modular setup using these motors. Due to the bearings limitations, the motors had to be positioned with central axis looking upwards; the only way to connect two of them with the shaft was to build a significantly large metal construction to lift one of the motors above another. First, NSK motors are heavy and lifting and securing one of them in a high position is a challenge. Such construction would have a significant cost and would require more labor hours than we considered allowable. Besides that, the NSK motors are very sensitive to the axial force and in this configuration would have difficulty in accommodating gravity forces.

The second pair of motors considered for the setup was Parker AC motors [52]. These motors are brushless as well and have an already available driver, System7. The problem with these motors was that they did not perform very well in previous experiments and their documentation is insufficient. In this case, the driver was practically useless and we had to consider some other options. Later, these motors were used as the real passive load since they had no brushes and gears.

The last pair of motors considered was geared PM DC motors Pittman 14202 [54]. This was the best choice and was considered for the setup prototype. The only problem that was encountered was that geared motors can not be connected to each other

since gears create backdrivebility problems. Only one motor has to be backdriveable, the motor used as the load simulator. The drive motor was chosen the geared Pittman 14202. For the load simulator, it was decided to purchase another small non-geared motor, Pittman 9232, with approximately 15 % value of the peak torque found in the actuator motor Pittman 14202. At that point, the decision was to have load simulator with approximately 10-20 % of the actuator power.

That finalized the choice of the motors for the first version of the experimental setup. Both motors were small and the base plate together with motor holders was built to suit their sizes. Between motors, there was gap of approximately 150 mm, left for shaft. Two versions of shafts were designed. The first one was a simple full shaft built form aluminum, while the second shaft, also from aluminum has a spring attached in the middle part of the shaft. The purpose of the spring was to introduce nonlinearities in the model. The experiments showed however, that the Pittman 9232 is too light a load for the Pittman 14202. The geared Pittman 14202 did not allow the Pittman 9232 to create any kind of disturbance, and results obtained showed that with or without the Pittman 9232, the characteristics were the same. That problem was explained by using simple lumped inertia calculation. The gear ratio in the Pittman 14202 is 1/75.1 and the equivalent inertia

$$J_{eq} = J_1 + \frac{N_1}{N_2} * J_2 .$$

is practically the same as J_1, since gear ratio makes second inertia almost negligible. It was decided to replace the actuator motor with another Pittman 9232. That change was the last and the second and final version of the setup was created. With an ungeared Pittman 9232 on both sides, the setup has the actuator and the load of the same torque value, which gives wide operating range to the setup. Since there are no gears, motors can freely rotate in both directions.

4.2.2. Control system

Another important segment of the setup is its control system. The main consideration for this HIL experimental setup was to use a system that can provide real time control. In the lab, we had two systems worth considering. The first one was the

dSPACE system [39] which was already proven as very reliable real-time control system and, the second LabVIEW [56] software under Windows operating system. Beside those, some other systems were also reviewed. For example, The Mathworks Inc. released a new real-time operating system called XPC target [53], which according to the company's tutorial was an excellent and easy to setup and use system. The main idea was to use two PC computers. One of them was supposed to be the host and another was designated as the target computer. The target computer would have run under XPC system and once booted it would perform only the determined tasks downloaded in the form of the compiled code from the host computer. The host computer would be the communication line between operator and target computer. The experimental setup would be connected to the target and the compiled program, in this case written in Matlab/Simulink or C, would be uploaded from the host to it. The XPC target system was just released and the lack of suitable drivers made it unusable for the current HIL system.

Windows is known not to have the characteristics of a real-time OS due to the interrupts. At some level Windows was still useful. With LabView on the top, it could perform simulations, and those results could be later compared with real-time results obtained from the dSPACE.

The dSPACE system was finally taken as the best choice because of its performance and reliability.

4.3 Experimental setup

Figure 4.1 shows the generic layout of the final design of the Hardware-In-The-Loop experimental setup. There are three main parts of the system; main testbed, control system and signal conditioning and isolation circuitry.

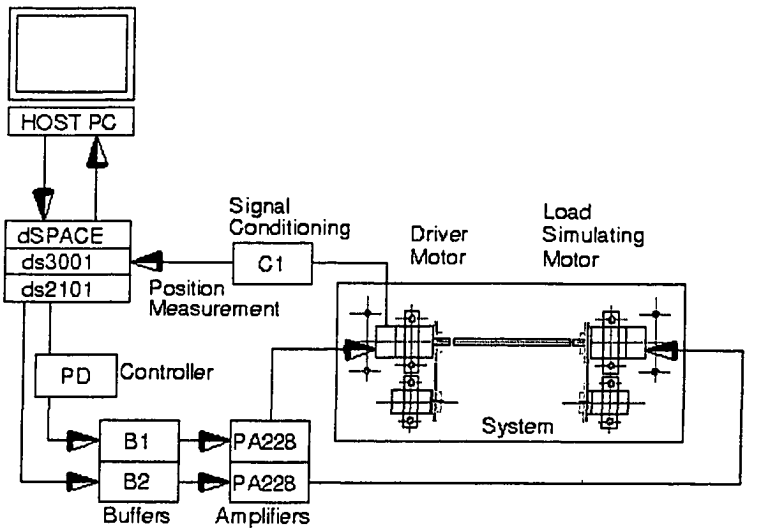


Figure 4.1 The system's layout

4.3.1 Main testbed

The main tesbed consists of two motors, a shaft, two tachometers, two pulleys, motor holders and the base plate. Motors are designed as the actuator and the load simulator, and shaft connects them. Both motors are Pittman 9232 and their characteristics are given in Appendix B. They are connected to the main plate with holders, which are made in two sizes to accommodate two different sizes of motors. Holders are made from aluminum and replacement of the motor is made easy by unscrewing two screws on the top of the holder. Each holder is fastened to the main plate with two screws from the bottom. At any moment, they can be removed if not needed. Shafts are made from aluminum. The purpose of shafts is to connect the motors and to introduce the nonlinearity, if the shaft with spring is attached to it. Another use of the shaft is to carry a disk considered as a passive inertial load. The stiffness of the simple shaft can be calculated by using the following formula [27]:

$$K_s = \frac{G * J}{L}$$

where G is shearing elasticity modulus – for aluminum $G = 26.2 * 10^{-9}$ $[N/m]$, J is shaft inertia calculated as $J = 1/2 * \pi * r^4$, and L is shaft length

This results in $K_s = 59.25 \quad [Nm]$.

Couplers connect the shaft and the motors. One coupler is flexible, while another is rigid. The base plate was made from 12.5 mm thick aluminum plate.

4.3.1.1 Sensors

Two types of sensors are used for the main testbed to generate feedback for the control system, tachometers and optical encoders.

There are two tachometers placed on smaller holders parallel to each motor. The tachometers are the same and the constant is $K_t = 20.8 \quad [V / krpm]$.In the case of using them, as a feedback from the motors, there is a pair of pulleys ready to be assembled with the purpose of transfering motor rotation to the tachometer. Pulleys are supplied as standard pairs and their ratio is r = 7.9. To avoid a slippage between pulleys and the belt, a timing belt was used.

The motors are equipped with optical encoders, used as main feedback during the research work. Both encoders have 500 pulses per revolution.

The Figure 4.2 shows the main tesbed layout.

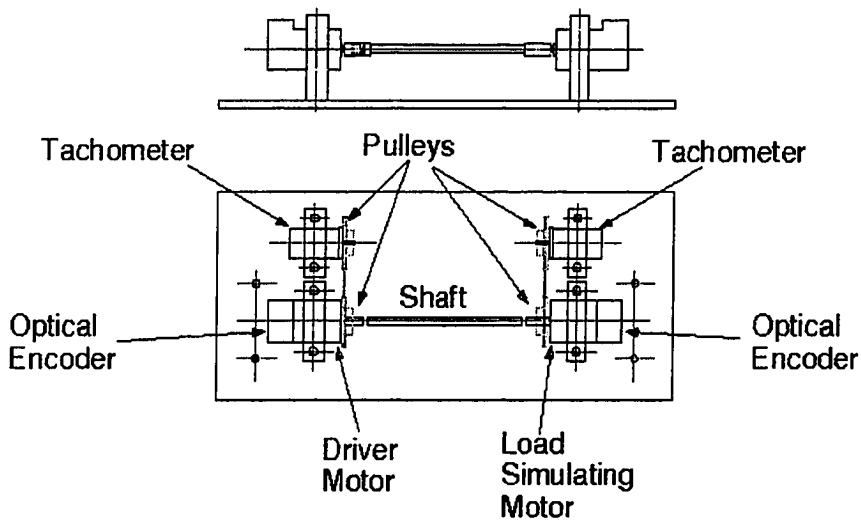


Figure 4.2 The main testbed

4.3.2 Control and monitoring system

The real-time computer system monitors and controls the main testbed. It consists of the dSPACE real-time operating system based on Digital Signal Processor (DSP) that has several boards attached to it, system's software and user built software [39].

4.3.2.1 Hardware

The hardware part of the dSPACE system includes processor, D/A converter, A/D converter and encoder boards [38]. The whole system runs on TMS320C30 processor chip. The dSPACE has no user interface and is not accessible directly. It has to be connected to a host PC computer, which has 486DX33 processor. The host computer is used to send commands to the dSPACE through expansion bus connection and to build user software using Turbo C developing environment.

The following boards are part of the dSPACE system [38]:

DS1002

This board is main board of the system. The TMS320C30, Digital Signal Processor from Texas Instruments chip is installed on it. The processor cycle is 60 nano seconds giving 16.7 millions of instruction per second.

DS2002

There are two DS2002 boards in the system. They are designed for high-speed variable digital control, and have two separate 16-bit A/D converters with 16 multiplexed inputs each. The conversion time for 16-bit is 5 μs.

DS2101

This board is D/A converter with five parallel 12-bit voltage outputs. The settling time to 0.01% is 3 μs. There are two boards in the system and each of them was used in research. One was driving the actuator motor and second was driving the load simulator.

DS3001

Two DS3001 boards are found in the system. They have five parallel 24-bit encoders each and can accept differential (RS442) or single ended (TTL) encoder inputs.

In this research, one board was used to accept and process the encoder signal from the actuator motor.

Further information about the boards is available in dSPACE DSP-CITpro manuals [39] [40].

4.3.2.2. Software

We can divide software into two categories: system and user software. System software came with the hardware and its purpose is to setup connection between the hardware and user's input as well as to permit the user to send direct command to the system using the host computer. There are three main commands that user can send directly to the system.

SED30

This command starts setup software, which configures all boards and generates a setup file. Also, by using this software, the user can configure the monitor display which will be used during tracing the signals from the system.

MON30

Command used to start monitoring process. Software contains user interface and by choosing user can upload software to dSPACE processor, run it or reinitialize boards. It is possible to run user program and quit the software, which permits the user to start tracing signals.

TRACE30

This software is used to trace signal from DS1002 board. It contains user interface to help user to setup appearance for signal graphs. There are two files needed to run Trace30.TRC is memory map file, which memorizes all signal values, while .MAP is a linker map file, which links signals with board addresses.

User software is written in C programming language. There is a possibility to convert software written in the Simulink to a C code. Programs are compiled using standard Turbo C compiler. The compiled code file .OBJ can be uploaded to DS1002 processor board by using MON30 command.

4.3.3. Amplifiers

A signal sent from D/A converter board DS2101 is used to drive the testbed DC motors. Each motor has its own signal line, which is passing through a power amplifier. The amplifiers are the same for both motors, PA228 Linear Power Amplifiers [57]. Power supply used to drive amplifiers is obtained from two-12 V batteries in series giving 24 V. The PA228 amplifiers were used for the preliminary analysis and the first part of experiments with the open loop configuration.

The second amplifier used in this research was the power operational amplifier PA12 from APEX [49]. The purpose of changing amplifiers was pure technical given that, one of PA228 was defective and it was not possible to find a new one any more. The amplifier PA12 was used for the second part of experiments when the voltage controlled current source circuit was introduced.

Appendix B lists technical characteristics of PA228 and PA12.

4.3.4. Signal conditioning and buffering

The feedback signal is sent from the sensor to the dSPACE system through signal conditioning boards. The purpose of those boards is to amplify and rectify a row signal from the sensor.

In the case of using an optical encoder, the generated signal is a train of pulses. Since pulses, from the encoder, are not shaped as perfect rectangular, a Schmitt trigger is used for the rectification. Another important issue is the level of the digital signal. If the signal is to be used with TTL compatible devices, it is important to keep the voltage of the signal above 2.8 V, for the digital "one" and below 0.8 V, for digital "zero". C1 from the figure 4.1 denotes the signal conditioning board for the optical encoder digital signal.

If a tachometer is used as a feedback device, than the signal generated is an analog voltage. It is important to condition that signal to fall in the range 0-10 V to be compatible with the system's A/D board. For that purpose, another board with an amplifier was assembled.

The signal sent from D/A board to the amplifier is buffered by the voltage follower buffering boards, with the purpose to protect dSPACE boards from drawing too much current. B1 and B2 from the figure 5.1 denote buffering boards used in the system.

Similar voltage follower boards were made for protection of the DAQ PAD board in case of using Windows and Lab View control system.

For more technical information about signal conditioning and buffering boards see Appendix B.

4.3.5. Digital filters

The signal received from sensors (optical encoders), was filtered from noise by a second order Butterworth digital filter.

The general form of a Butterworth filter is given as [58]:

$$H(z) = \frac{y(z)}{x(z)} = \frac{\sum b_i * z^{-i}}{1 + \sum a_j * z^{-j}}$$

The equivalent discrete-time relationship is:

$$y(n) = b(1) * x(n) + b(2) * x(n-1) + b(3) * x(n-2) - a(2) * y(n-1) - a(3) * y(n-2)$$

This form was used in the C programs used to run experiments.

The following table 4.1 shows the coefficients that can be used for different filter cutoff frequencies [58]. All coefficients are obtained for the sampling frequency of 1000 Hz. All the programs designed for the control purposes and written in C programming language for this thesis used sampling time of 0.001 second. The sampling frequency of 1000 Hz suits that fact. During this research, the cutoff frequencies of 1 Hz and 10 Hz were also used to filter the signal.

Coefficient	1 Hz	5 Hz	10 Hz	20 Hz	50 Hz
a(2)	-1.9911	-1.9556	-1.9112	-1.8227	-1.5610
a(3)	0.9912	0.9565	0.9150	0.8372	0.6414
b(1)	0.00000983	0.0002414	0.0009	0.0036	0.0201
b(2)	0.00001965	0.0004827	0.0019	0.0072	0.0402
b(3)	0.00000983	0.0002414	0.0009	0.0036	0.0201

Table 4.1 Coefficients of the second order Butterworth filter (for 1000 Hz sampling frequency)

4.4. Linearity of the system

A system is called linear if the principle of superposition applies [19]. In order to build a linear lumped-parameter model and analyze it, a system has to be linear. If the system is linear then its input-output relationship is also linear. Since Hardware-In-The-Loop modeling depends largely on the linearity of systems components, it was important to check linearity of the motor(s) used.

The following experiment was proposed to check the linearity of the DC motor Pittman 9232. The main idea of the experiment was to send signal from dSPACE to the Pittman DC motor. That signal will during the steady state response, produce a certain velocity. If the motor is linear, than a change in signal would produce a proportional change in velocity. Several different constant voltage inputs were sent to the DC motor to check its linearity. Motor was connected to a tachometer by set of pulleys and output signal from tachometer was monitored in LabView program specially created for this purpose. Here we presume that the tachometer and amplifiers are linear meaning that tachometer is without eccentricity and that noise level of both sensors is minimal. Only signal acquired during steady-state response was considered for analysis.

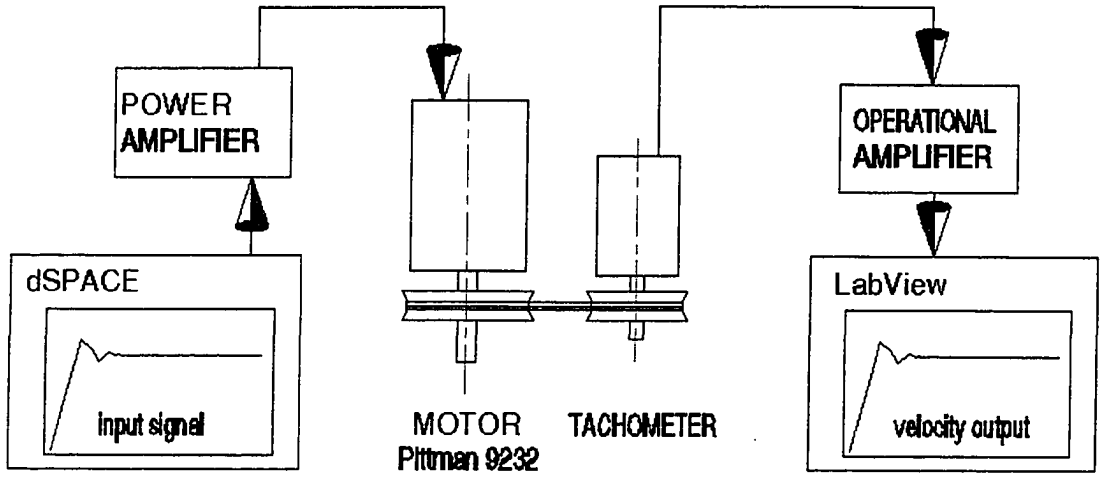


Figure 4.3 Linearity of the system

The tachometer constant is 20.8 $\left[\frac{V}{krpm}\right]$ and the set of pulleys used to transfer signal from motor to tachometer has ratio of 7.9. A timing belt, which prevents error due to sliding of the belt on the pulley, was used.

To make sure that a signal from tachometer is in the range of 0 – 10 V an amplifier is used which amplifies original signal 4.3 times. Table 4.2 shows recorded data.

Input signal [V]	Output signal [V]	Calculated velocity [rad/s]
0.4	0.85	2.329
0.5	0.88	2.412
0.6	0.92	2.522
0.7	0.96	2.631
0.8	0.976	2.675

Table 4.2 Experimental data

The following Figure 4.4.shows a linearity of experimentally obtained data

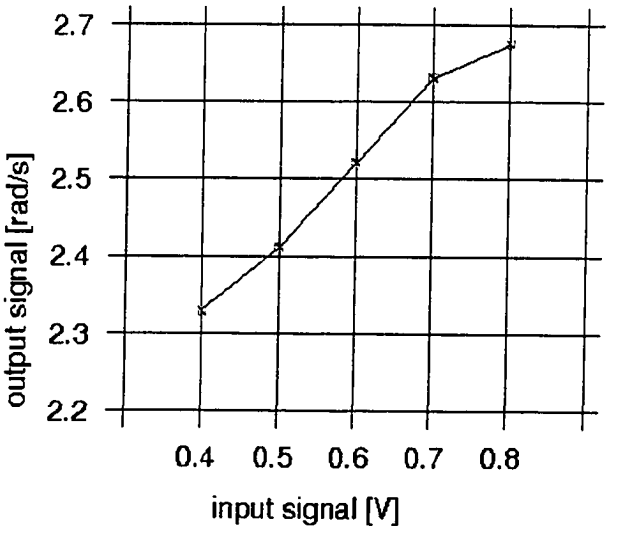


Figure 4.4 Linearity of the DC motor Pittman 9232 – experimental results

A straight line can approximate experimentally obtained data, and this indicates that the system is linear. This was necessary to prove before we proceed with other experiments that require motor linearity.

4.5. Breaking current

The minimum current needed to start the Pittman 9232 was obtained experimentally by sending incremental signals from the dSPACE. Three different sources of friction are acting on a DC motor. Viscous friction depends on the velocity of the motor and does not affect the motor when it is not moving. Two other types of friction, Coulomb friction and stiction affect a motor when it is not moving. They can be combined and together are called dry friction of the motor. The breaking current is current that "breaks" dry friction, providing motor with enough torque to overcome the stiction.

Experimental study permitted to determine that the breaking current is 0.86 [A]. Minimum voltage sent from the dSPACE was 0.43 [V]and since power amplifier transfer function is 2 [A/V], the required breaking current was obtained.

To find the value of the dry friction of the Pittman 9232 motor, the breaking current is multiplied with the motor constant that can be found in the motor manufacturer's data.

$T_{dry} = 0.86 * 0.0156 = 0.0134 \quad [Nm]$.

We can find Coulomb friction in the manufacturer's data as $T_c = 3.5 * 10^{-3} \quad [Nm]$. Knowing that, stiction is $T_s = T_{dry} - T_c$, or $T_s = 0.009916 \quad [Nm]$.

4.6. Loads

Besides the load simulator, which is part of the experimental setup, several other loads were used in the research. Hardware-In-The-Loop simulations require two types of loads, real load and simulated load, where the second is acquired with the load simulator. Real loads used in this research are Parker AC motor, a fan and a disk. Parker motor was used as passive linear inertial-viscous load. That was accomplished by not providing electric power to the motor. The motor's rotor was turning freely inside of the stator using the moment from the actuator. This way the rotor inertia and viscous friction were considered as part of the passive load.

A fan is a nonlinear viscous-inertial load. It was used to try to simulate nonlinear loads with HIL simulations. A disk is a linear load and it was attached to the shaft connecting two motors. While the disk was on the shaft, the load simulator was acting as a passive load so total load was linear and its value was added values of PM DC motor's rotor and disk itself.

4.7. Safety Issues

Safety of the Hardware-In-The-Loop experimental setup has three basic levels:

a) Mechanical level

b) Electrical level

c) Digital level

We should assume that the motors are rotating in the same direction and that can be ensured by proper wiring of the motors. Before motors are connected with the shaft, they should be tested with a test signal to ensure the direction of rotation for each motor.

On the mechanical level, it is necessary to pay attention to the connections between mechanical elements used in the setup. If the load applied on either one of the motors is too large or if the motors are trying to rotate in the opposite direction, it is important to introduce such devices as clutches or couplings to prevent motors from burning. The HIL experimental setup uses couplings as connections between motors and shaft(s). In the future, it is recommended to use clutches instead, for at least one of the couplings.

On the electrical level, the most important issue is opening of the circuit in the critical moment. That protection is known as over current protection. If the motors are drawing too much current from the source or if the load motor draws more current than actuator motor the whole process must stop, to prevent burning either one of the motors. The implementation of that safety level requires two types of devices. First is simpler, and includes fusses at all critical places in the circuitry.

Those fuses must obey certain rules and should be appropriately chosen. It is important for the fuse not to react too fast since the current drawn when a motor is starting is very large. Fuses must be able to handle currents up to 10 times of the rated current for the motor.

Figure 4.5.shows the fuse characteristic [14].

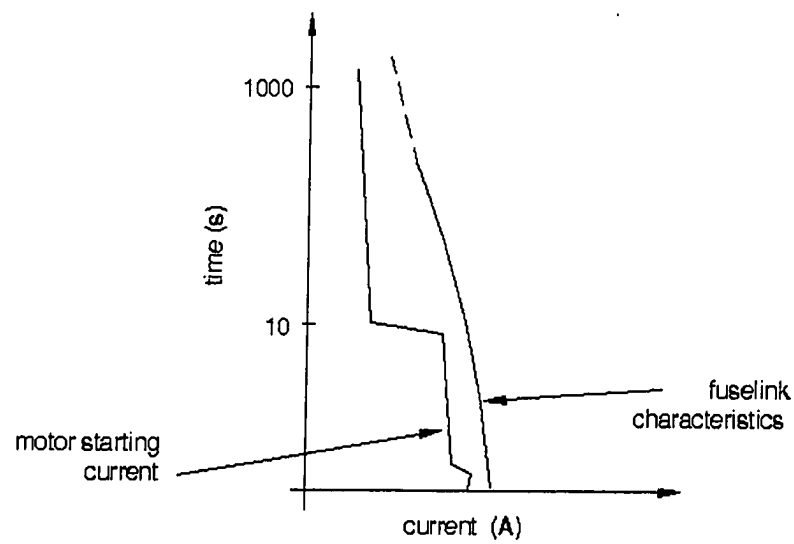


Figure 4.5 Safety characteristics of a fuse [14]

The second device that could be used for the electrical level of safety is a comparator. The comparator must be able to receive signals from the both motors and those signals

should be compared and then processed. If the signal from the load motor is higher than the one from the actuator motor, the process must stop by disconnecting motors from the current source. In that case, the motor would be overloaded since a higher current at load motor side would produce greater moment, and the actuator motor would be exposed to the higher load than it was designed for. This refers to the overload protection of the motor.

Other devices that could be used to protect the setup on the electrical level are circuit breakers, low and high voltage contactors, solid-state contactors, and programmable controllers.

The third and the last level of the safety is at the digital level. The software used in a HIL simulation should be implemented in such way that signals driving the actuator and load motors should stay within certain limits. If the current to drive those motors is too high, the signals should be stopped such that the motors will not be overloaded. We can call this type of device software comparator.

5. EXPERIMENTAL CONFIGURATIONS

5.1 Introduction

The chapter 3 gave theoretical bases for the development of the experimental setup and for the thesis in general. It prepared particular models for different kinds of loads and the load simulator itself. Moreover, the model of the whole system for the Hardware-In-The-Loop (HIL) simulation was developed and presented as well as methods for simulating torques in the case when the load simulator is used. The effort-flow concept presented and used in previous chapter is the basis on which HIL simulations stand. Modularity, which is emphasized by the concept, permits interchangeability of real and simulated components in the HIL simulation.

Development of the experimental setup was based on the theory presented in Chapter 3. During the research, the experimental setup was steadily improved and according to that, we can talk about three phases of the HIL experimental setup development and three sets of the results obtained accordingly.

The first two phases were actually implemented and results obtained and will be presented in Chapter 6, while third one is recommended for future research.

The phases are:

1. Open loop experimental setup with a position measurement used to monitor velocity characteristics of the load (no control feedback was used) – HIL1
2. Closed loop experimental setup with a current (torque) feedback used to control the torque characteristics of the load simulating motor and position measurement used only to monitor velocity characteristics of the load – HIL2

3. Closed loop experimental setup with a torque feedback, using an actual torque sensor, and a position measurement to monitor velocity characteristics of the load – HIL3

Phase 2 can be implemented in two different ways, digital or analog, depending on the control system ability to handle analog feedback signal. The computer used for this thesis had problem with A/D converter, so analog version of current feedback was implemented for the phase 2.

5.2 Phase 1 – Open loop (HIL1)

In the first phase (HIL1), the experimental setup is in an open loop configuration, i.e. a feedback was not implemented. The purpose of the first phase was, in whole, to monitor velocity characteristics of the real and simulated loads and to prove, experimentally, the validity of the sets of formulas derived for the purpose to simulate loads. Figure 5.1 shows the concept of the HIL1 development stage presenting main components of the experimental setup. It clearly shows that no other feedbacks were used. Position measurement is used to monitor velocity profile.

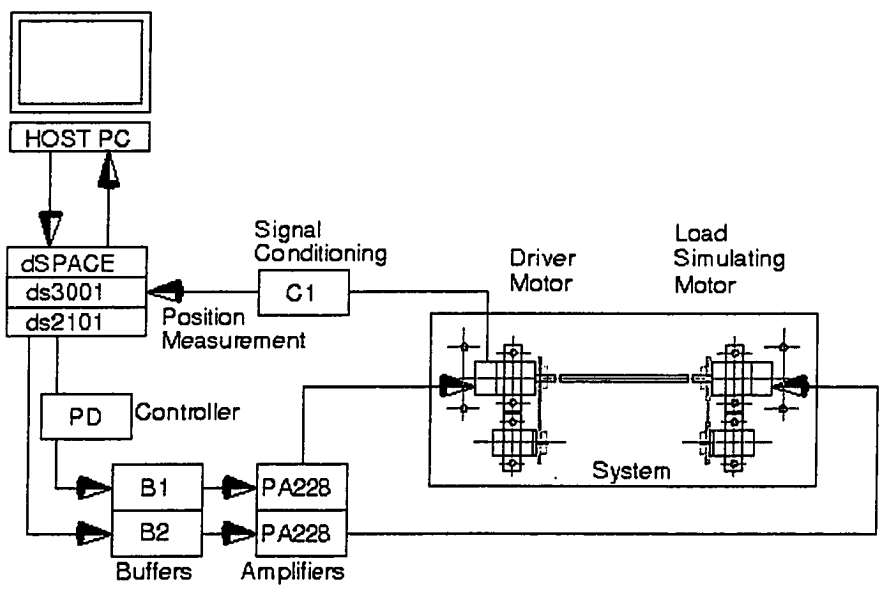


Figure 5.1 HIL1 concept

The following block diagram, shown in Figure 5.2, corresponds to the same HIL1 concept. System is clearly open loop, without any control feedbacks. Optical encoder obtains the only response of the system and from that velocity characteristic was derived. Velocity derived from position feedback was used to calculate input for load simulator using formulas founded in Chapter 3.

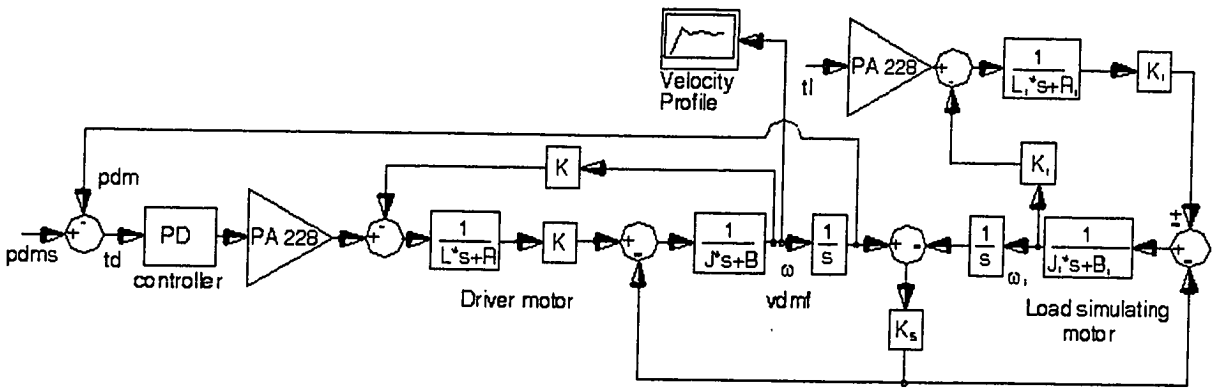


Figure 5.2 Block diagram for the open loop HIL1 concept

5.3. Phase 2 –Closed loop with current feedback control (HIL2)

The motion control (position and speed control) requires indirect control of the motor torque [42].This is achieved by measuring armature current feedback. Closed loop of the experimental setup using current feedback can be achieved in two ways. One is to measure a voltage over a small resistor in the motor supply circuit [29] and feed back that signal to a computer through analog line to A/D converter. A converted signal can be used to obtain a current error and that error can be sent back as signal to the setup. Another way is to use the same resistor to measure voltage drop and than condition that signal and send it to analog voltage differentiator to obtain analog voltage error [41]. In both cases, measured and conditioned physical variable is voltage, which in fact represents current. The experimental setup is designed as torque-controlled setup and

current is directly connected to the torque, which means that controlling current is same as controlling torque.

5.3.1 Digital implementation of current control

Figure 5.3 shows design concept of experimental setup with digitally implemented current feedback. The dSPACE provides two A/D converters DS2002 and they can be used to feed analog signal, measured as a voltage drop over a small resistor, to computer. The small resistor used to measure voltage drop has to be small enough not to affect adversely motor dynamics. If the chosen resistor is high in value then the motor may not receive enough power to turn or perform assigned action. On the other hand, too small of a resistor may not produce enough voltage measured across and signal might not be sufficient for further usage. Another concern is the power that resistors can handle. Since DC motors used in the experimental setup can receive up to 6.22 A, such a high current can easily destroy a small resistor. For that purpose, a calculation has to be performed to see what power dissipation is needed for the resistor.

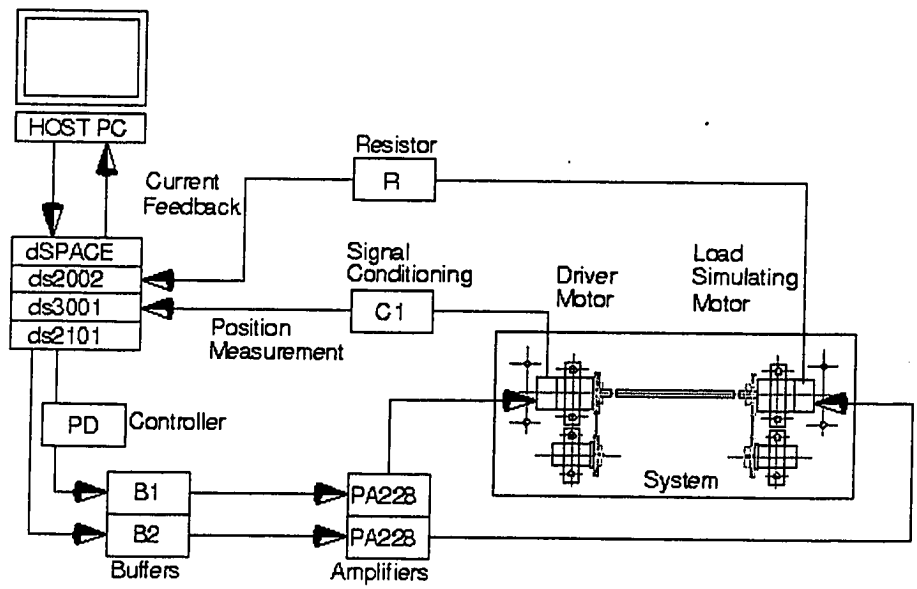


Figure 5.3 HIL2 concept – digital implementation

When the analog signal is received and converted using DS2002, it needs conditioning, which can be implemented inside of the control software. The program

requires the part of the code, which would subtract measured signal from the reference signal and that way create the error signal that can be sent back as the input to the setup. Figure 5.4 shows block diagram, which corresponds to HIL2 concept. The voltage drop measured over a small resistor has to be divided by the value of the resistor to obtain instant value of the current. Input voltage U represents the desired current and the summation point creates a current error by subtracting two currents – measured and reference. Resulting error is fed back to the power amplifier as input. A position feedback is still part of the setup with the purpose to monitor velocity characteristics of the load and supply velocity as variable necessary to calculate torque for simulated load.

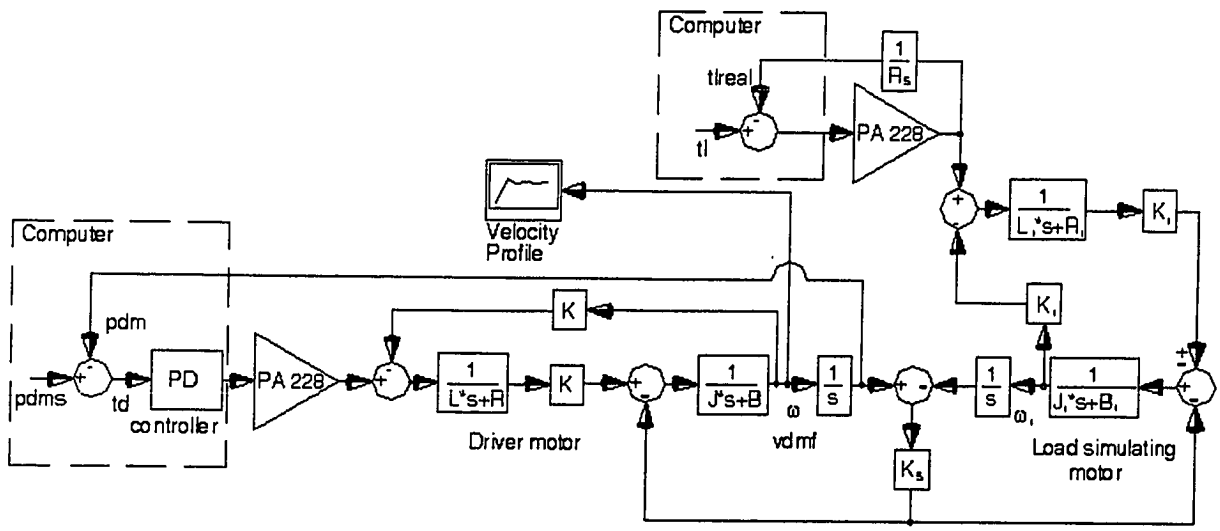


Figure 5.4 Block diagram of the digital implementation for the closed loop HIL2 concept

5.3.2 Analog implementation of current control

The difference between analog and digital implementation is that the signal sent from the computer in digitally implemented control is the current error, since summation process is part of the software, while for analog implementation computer is sending only reference current and summation is performed in circuits outside of computer. The signal obtained by measuring the voltage drop has to be conditioned. One of the stages in signal conditioning is to obtain the value of the current from measured voltage and that can be achieved by dividing the voltage with the value of the small resistor. The summation is

performed in the circuit, which is taking input from the computer as reference (desired) current, and actual conditioned value of the current. The error signal is sent back to the power amplifier and system.

Figure 5.5 shows the HIL2 concept with analog implementation of the current feedback.

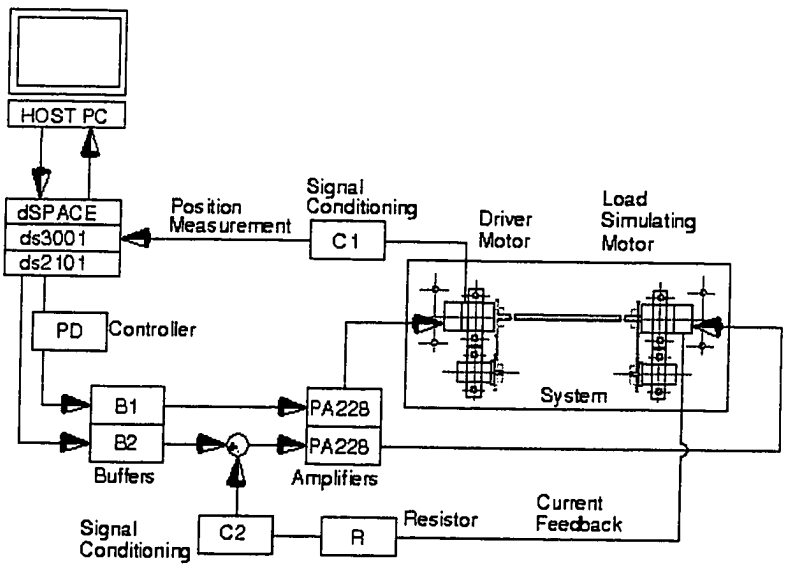


Figure 5.5 HIL2 concept – analog implementation

Figure 5.6 presents the block diagram of the system for analog implementation of current feedback.

As it can be seen from the block diagram, the feedback processing takes place outside of computer and the computer only supplies the reference point, in this case the desired current. Signal conditioning and summation are implemented in circuit hardware. R_s represents the value of the small resistor , while K_{sc} is the gain that replaces a lumped value for all other conditioning values. Details about analog implementation are given in paragraph 5.3.2.1

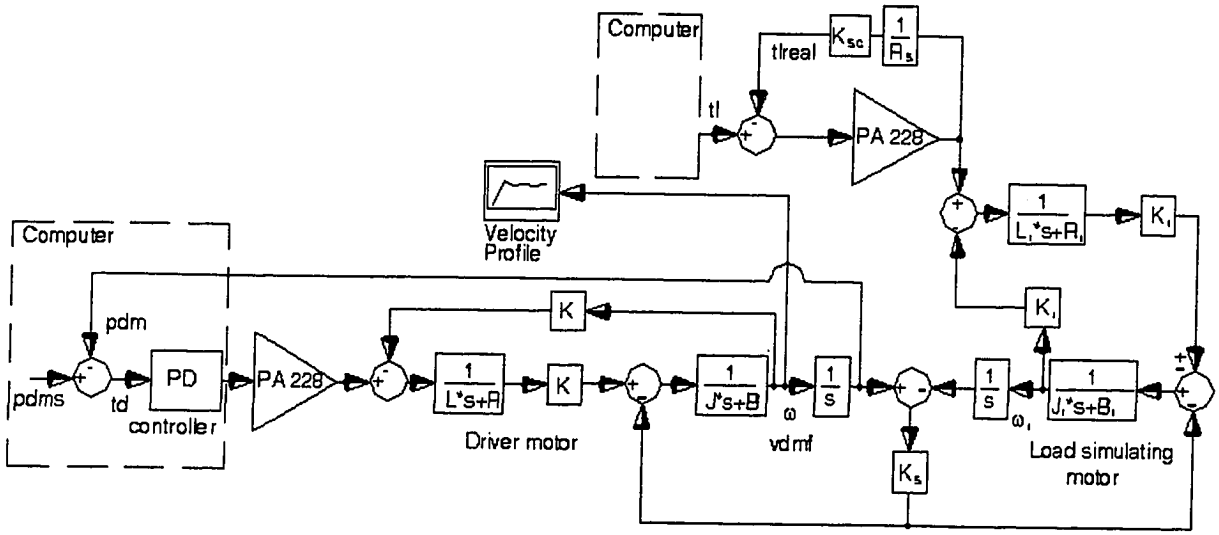


Figure 5.6 Block diagram of the analog implementation for the closed loop HIL2 concept

5.3.2.1 Analog implementation of voltage controlled current source

The circuit implemented for the voltage control of current source is built around a power operational amplifier [49]. The sensor part of the circuit is a small resistor, which is connected to the ground and whose purpose is to create a small measurable voltage drop in the circuit. This voltage drop is proportional to the current that flows in the output loop of the amplifier. That current can be expressed by the following formula, using voltage drop on the small resistor.

$$i = \frac{V}{R_s}$$

Figure 5.7 shows circuit used as analog implementation of voltage controlled current source.

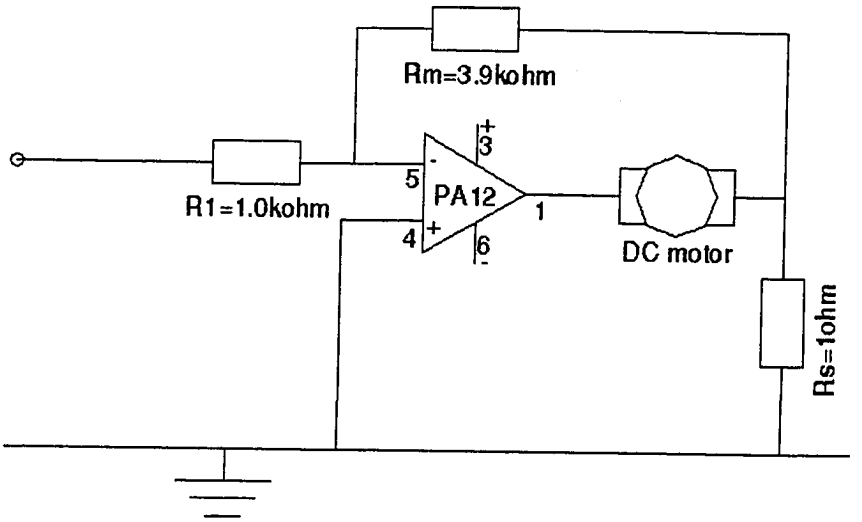


Figure 5.7 Analog implementation of voltage controlled current source

The resistor, which produces a measurable voltage drop, was designed as follows. To produce a voltage drop, which can generate a significant signal and in the same time, not to interfere with power needed for motor to perform required tasks, a resistor with $R_s = 1 \quad [ohm]$ was chosen. Such a small value of the resistor is vulnerable to high currents unless resistor has high dissipation power. The resistor that was used has 10 W of dissipation power, which permits current to be up to $i = 3.16 \quad [A]$. Highest values of the current obtained using test signal were ~1.5 [A], which gives enough space for resistor to work properly. The resistor is placed in amplifier's output circuit with motor. In this case, the motor is a floating load.

The power amplifier used in this application is PA12 from APEX [49]. The amplifier is set using inverting configuration. According to the recommendation from manufacturer, the smallest gain needed for amplifier to work properly is 3. In this case, the amplification is set to be 3.9. The first resistor is set to be $R_m = 3900 \quad [ohm]$ and the second $R_m = 1000 \quad [ohm]$.

The transfer function of the amplifier with current feedback can be written as

$$\frac{i_o}{V_{in}} = -\frac{R_f}{R_i} * \frac{1}{R_s} = 3.9 \quad \left[\frac{A}{V}\right]$$

The formula is indicating that the current obtained in the output circuit is 3.9 A for an input voltage of 1 V.

5.4. Phase 3 - Closed loop with torque feedback control (HIL3)

The following is the presentation of the third phase of the HIL experimental setup. This is only a recommendation and the real implementation depends on the future research plans.

Figure 5.8 shows the basic concept of a generic HIL design with torque sensor feedback. The difference between this and previous concept is that torque sensor actually measures the torque at the shaft and this way the real value of the torque is obtained. In the HIL2 phase, the current of the DC motor was measured and that value was converted to torque, knowing the direct relationship between torque and current. HIL3 phase is more advanced since the motor torque is not the same as the load torque measured on the shaft. Here measurements of the torque and control are more accurate.

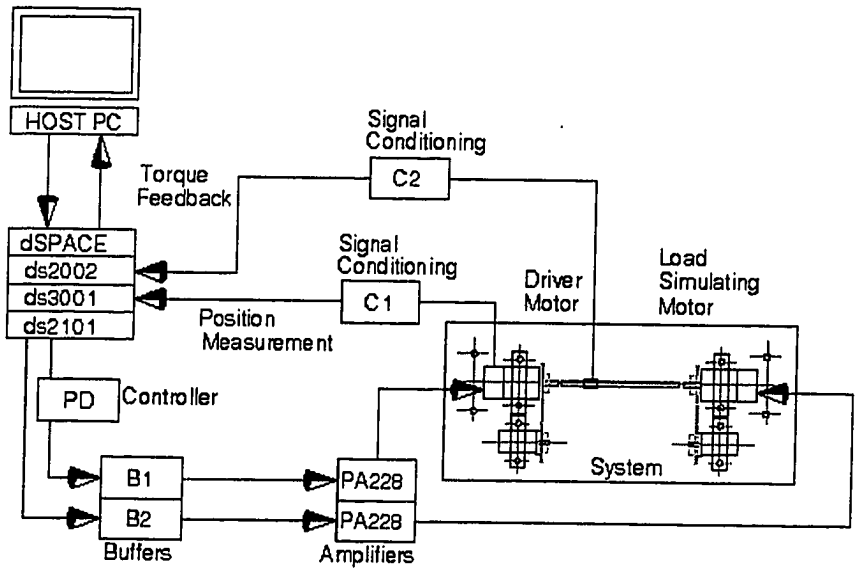


Figure 5.8 HIL3 concept

Corresponding Block Diagram is shown on the Figure 5.9.

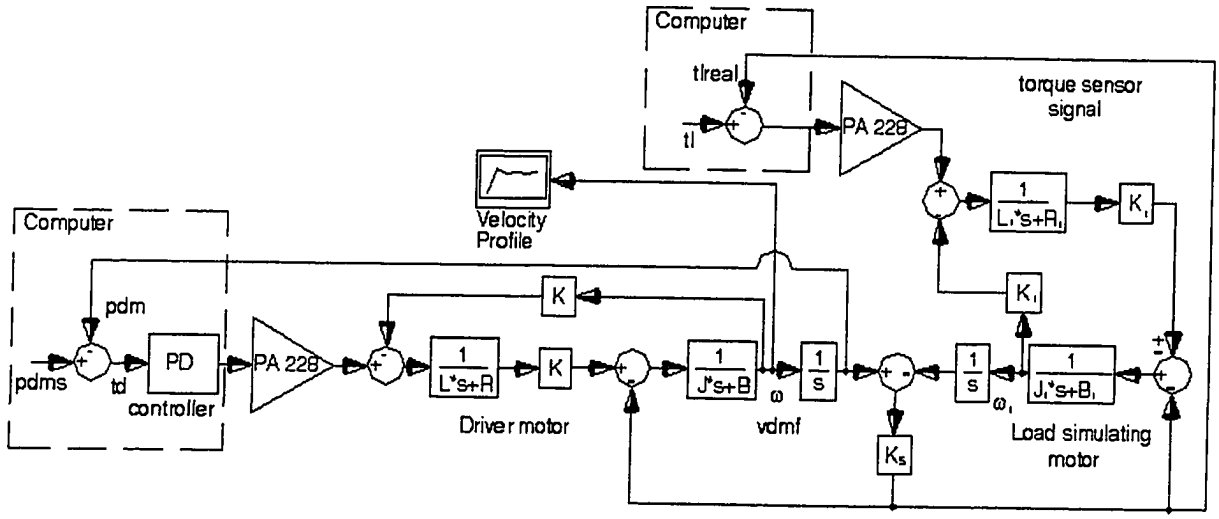


Figure 5.9 Block diagram for the closed loop HIL3 concept

6. EXPERIMENTAL RESULTS

6.1. Introduction

Experiments with the Hardware-In-The-Loop setup were conducted in three levels. The first level was designed to confirm that derived formulas based on theoretical background could be implemented in real experimental situation. In addition, it was necessary to analyze the behavior of the load simulator for a series of changes in load variables, which we used to generate simulated loads.

The second level of experiments was designed according to first design concept described in Chapter 5. An open loop HIL setup was used to monitor the behavior of the velocity profile for a variety of loads. The load simulator was not controlled by any feedback signal. Real loads used for this stage were linear, inertial, rotational loads represented by the rotor of the Parker DC motor passively connected to the actuator by a shaft and by a nonlinear fan. The load simulator, represented by the DC motor Pittman 9232, under two different sets of inputs simulated these loads. Inputs were designed as variety of positions, which were followed by driver motor. A simple PD controller was used to accomplish this task. The first input was a linear increase of position, which resulted in step velocity, while the second input was a combination of ramp and cosine function varied with three different frequencies. The second position input resulted in sinusoidal velocity characteristics.

The third level of experiments corresponds to the second design concept described in Chapter 5. This time a current feedback control was introduced in the setup to control the torque applied to the load simulator. The driver motor side remained the same using the

PD controller to control position assigned to the system. Again, the same real loads were used and later their characteristics were physically simulated.

For all experiments, the same experimental setup was used. The difference for the last level was that an Apex PA12 amplifier replaced a PA228 amplifier. That was done because one of PA228 was defected and the original amplifier was not available on the market any more.

6.2 Input output considerations and display form

The main purposes of the software were to generate an input signal, receive velocity characteristics of the system and according to the given load parameters create signal for physical load simulator.

The input from Figure 5.2, for first level of experiments, was a ramp and for the rest of experiments was a step plus sinusoids. The Trace30 software traced signals and the intervals of 10 points were used to record points. This means that each 1.5 seconds recorded on the record sheet, in reality covers 15 seconds.

First type of input was a linear increase of position, with the purpose to see the response of the system trying to accomplish step velocity characteristic. The equation used for this input set point was

$$U = P_{dms} = \begin{cases} 0.04 * t & for \quad t \le 10\,\text{sec} \\ 0.4 & for \quad t > 10\,\text{sec} \end{cases} \tag{6.1}$$

The input is a ramp, for the first 10 seconds of the experiment, and afterward a constant. Figure 6.1 shows the set point for the input pdms over the experimental time that is denoted in Figure 5.2.

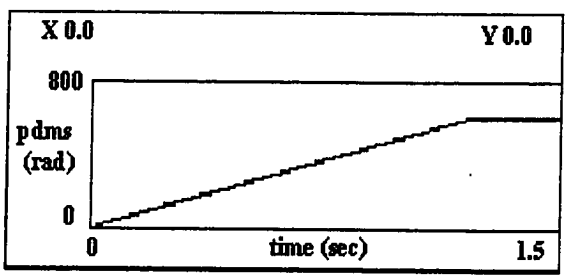


Figure 6.1 Ramp input

Second input type was a combination of ramp and cosine function. The basic formula used for this position set point was:

$$P_{dms} = 0.04 * t - 10 * \cos(\omega * t) \tag{6.2}$$

$$P_{dms} = 0.04 * t - \cos(\omega * t) \tag{6.3}$$

The equation 6.2 was used for the experiments with the frequency of 1 Hz, while equation 6.3 was used for all other higher frequencies. The reason for use of equation 6.2 was to create higher amplitude of the velocity signal, which would be useful for analysis. With amplitudes 10 times smaller and 0.159 Hz frequency, the output would look almost as straight line. For the frequencies of 0.796 and 1.592 Hz that was not necessary since output lines were reasonable.

The purpose of this signal was to create sinusoidal velocity characteristic, which would not cross the zero line to avoid pass through dry friction zone. The input frequency of the cosine part ω was changed with the purpose to see if the load simulation is possible at low as well as higher frequencies. Figure 6.2 shows the second input for various frequencies. Since the cosine part of the combinational input is small in value, it is not possible to notice that change on the Figure. Only calculated velocity can show that this input contains cosine part.

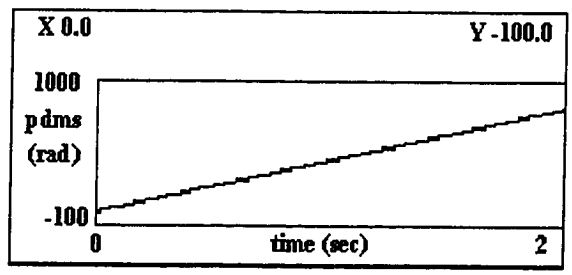


Figure 6.2 Combined input with 0.159, 0.796 or1.592 Hz frequency

The velocity characteristic was monitored in all simulations and was considered the main output. Besides that, velocity characteristic was used to calculate acceleration and jerk components needed for signal creation for load simulator.

The Trace30 software was used for real-time acquisition and display of data. The following was standard display schedule for the monitor. Six graphs were displayed at the time. The first row was reserved for the input and control signals. The left graph

displayed control signal to the actuator produced by PD controller and the right graph displayed signal sent to the load simulator. The second row was reserved for positions. The first one was set point position assigned to the system as an input, while the second was measured on the actuator's optical encoder. The third row contains calculated velocity and acceleration.

6.3 Preliminary analysis

The first experiment was designed to check the validity of the derived formulas for physical simulations of the loads. The implementation of such an experiment was based on the idea of trying to expose the physical simulator to different types of signals and to monitor the behavior of its output while the virtual inertia and viscosity of the simulator change. The results can be compared and the conclusion drawn knowing the basic responses for the loads with different inertias and viscosities. The experimental results can be interpreted qualitatively as follows. For example, if subjected to the same input, two different loads with different inertias will respond differently. The load with higher inertia will respond much more slowly because of the acceleration factor in the load $J\frac{d\omega}{dt}$, while a load with lower inertia will have a quicker response. On the other hand, two loads that differ in viscosity will behave differently again. The load with higher viscosity will show higher damping characteristics than the load with lower viscosity.

A physical simulator can simulate any load by changing the current characteristics, which are calculated based on the given inertia and viscosity. If we start experimenting with various inertia and viscosity values and say that that particular output is the base for the further analysis, we can change inertia and viscosity values and verify if the changes in the output respond to the input changes.

The input signal designed for this experiment includes several simple position control inputs, which follow each other. The input signal starts as a slowly increasing position input, after which it continues with a parabolic and another linearly increasing signal. The second linearly increasing signal has a different tangential constant, which produces a different velocity. The signal finishes with a decreasing parabola signal and

finally with a decreasing linear position (deceleration). The last two parts will force the system to turn in the opposite direction and finally stop. The input signal ends at zero after the deceleration part. The Figure 6.3 shows the input set point signal designed for preliminary analysis.

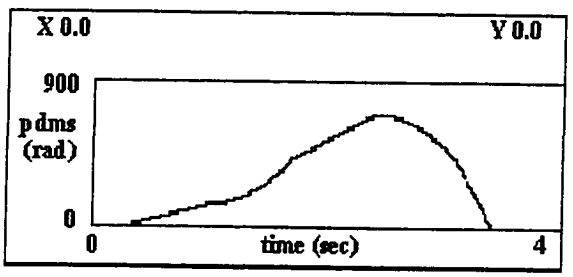


Figure 6.3 Input for preliminary analysis

The following conditions were assumed for the experiment: the simulated load is linear, the inertia and viscosity are higher for the real load than for the simulated load and a low-pass filter filters all frequencies higher than $\omega_{lpf} = 1 \left[\frac{rad}{s}\right]$. The following equation was used to implement the physical simulator (from Table 3.1):

$$T_{sim} = \frac{(J_{sim} * s + B_{sim}) * K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) - F_{dry}$$

where:

$J_{sim} = 0.0001 \quad [kgm]$

$B_{sim} = 0.0001 \quad \left[\frac{Nm}{rad/s}\right]$

$F_{dry} = 0.0000234 \quad [Nm]$

The rest of the values needed for the equation are known. J_l and B_l are inertia and viscosity of the physical simulator Pittman 9232 and K_s is the stiffness of the shaft that connects the actuator and the physical simulator. ω is the velocity of the actuator measured as a position on the optical encoder and calculated using the sampling period as a base. It is used to calculate the necessary additional torque and represents the output of

the system. The Figure 6.4 shows the output of the system obtained using Trace30 software.

The first row shows variables td and tl, from Figure 5.2, which represent the control signal sent to the driver and the calculated signal sent to the physical load simulator respectively. While td represents the signal that depends on the desired input and is produced by the PD controller, tl depends on the velocity output and is calculated in real time. The second row shows the desired and actual positions of the system presented by pdms and pdm variables, while last row is reserved for the velocity vdmf, denoted in addition to ω in Figure 5.2, and the acceleration admf.

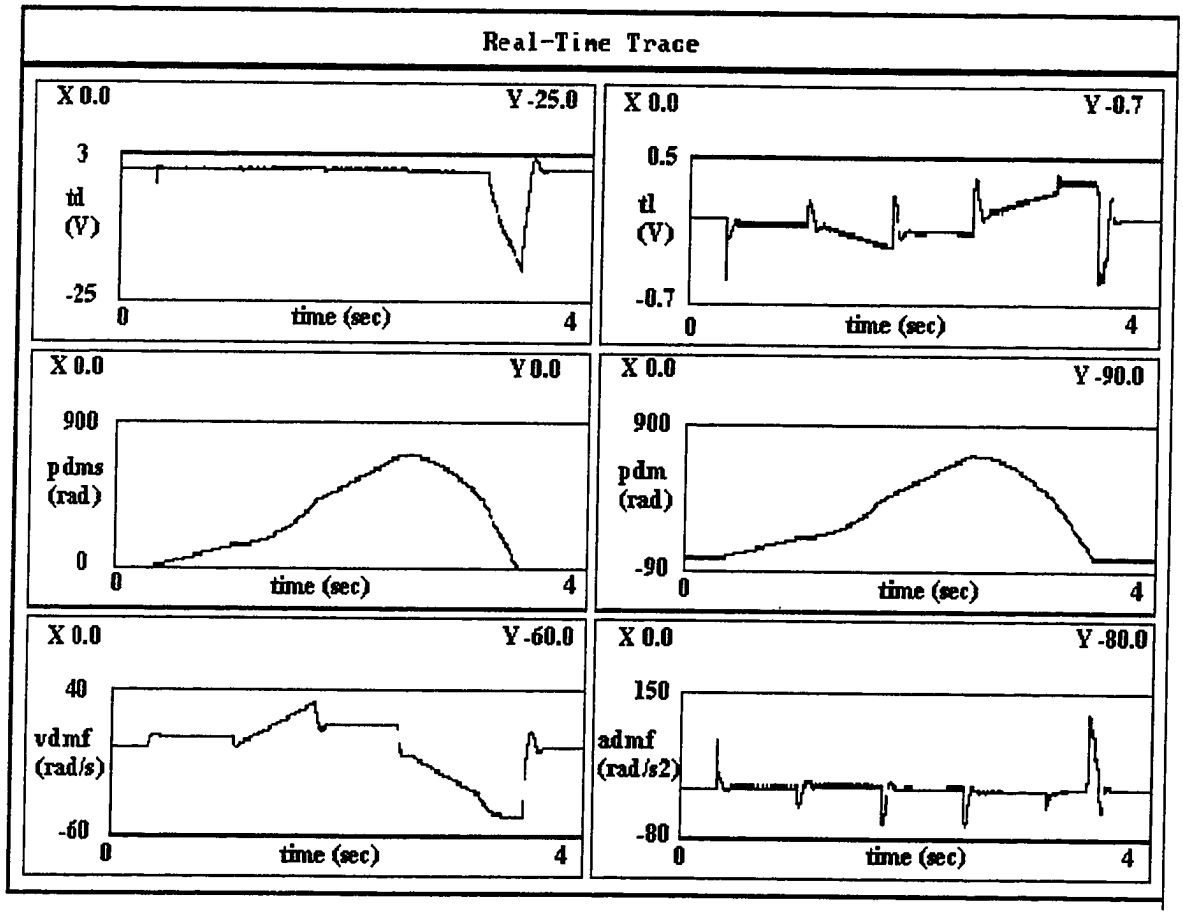


Figure 6.4 Initial output for a high inertial load

It can be observed that a velocity characteristic is a numerical derivative of the position. For the increasingly linear parts, the velocity is constant, while for parabolic sections it tends to increase or decrease in value. The acceleration, which is one of the

major contributors to the calculated torque, used as the signal for load simulator, has its local peaks every time when the position signal changes its section. The biggest peak can be noticed at the end of the cycle, when the system stops, and that is the point when the controller produces the highest negative signal to the actuator.

The next experiment was conducted with a different simulated inertia. The new values were:

$J_{sim} = 0.00001 \quad [kgm]$

$B_{sim} = 0.0001 \quad \left[\frac{Nm}{rad/s}\right]$

$F_{dry} = 0.0000234 \quad [Nm]$

We can see that the new inertia is 10 times smaller than the previous one. If we keep all other values the same and repeat the experiment the output shown in Figure 6.5 is obtained.

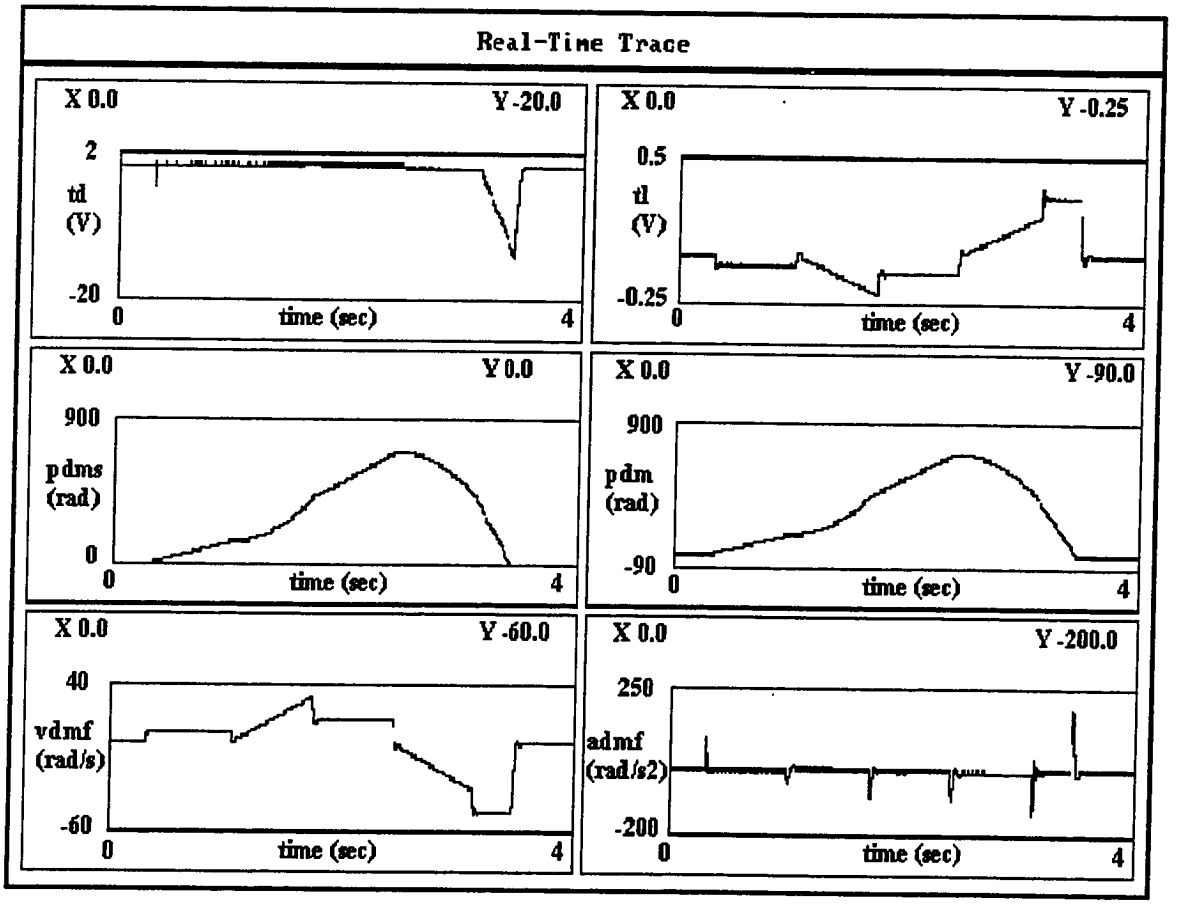


Figure 6.5 Output with reduced inertia

We can observe that with reduced inertia the shape of the actual position follows the desired signal. The velocity output has sharper rise and the overshoots are reduced throughout. That is due to a smaller inertia, which was large enough for the initial experiment to produce transitions with higher overshoots. The signal created for the load simulator has smaller peaks, which reflect smaller peaks in acceleration. The largest disturbance can be seen at the end of the experiment. The conclusion for this experiment is that the smaller value of the inertia allows the system to have higher maneuverability, which is visible in smaller acceleration peaks and smaller overshoots when the signal changes.

The next experiment was conducted with the inertia 10 times higher than the initial one. The following values were used:

$J_{sim} = 0.001 \quad [kgm]$

$B_{sim} = 0.0001 \quad \left[\frac{Nm}{rad/s}\right]$

$F_{dry} = 0.0000234 \quad [Nm]$

Without changing any other values or conditions, the following results shown in Figure 6.6 were obtained.

The higher inertia results in smoother transition between different input sections. The actual position is not exactly copy of the desired position due to the higher inertia. That is especially visible at the end of the cycle, when the PD controller has a problem to follow the given position task. The high inertia creates a position overshoot, forcing the actual position to go into negative values, which is later rectified by the controller. The velocity characteristics definitely do not follow the original velocity characteristics. Even though the actual position seems to be following the desired values, the velocity shows that this is not the case. We can see that the velocity is trying to maintain the same level without respect to the input signal. The high inertia can be compared to a car flywheel, which function is to maintain the constant speed in the car engine. Acceleration of the system is staying around zero with no clear peaks. Instead of peaks, acceleration is unstable when the velocity changes. The signal sent to the load simulator reflects that acceleration.

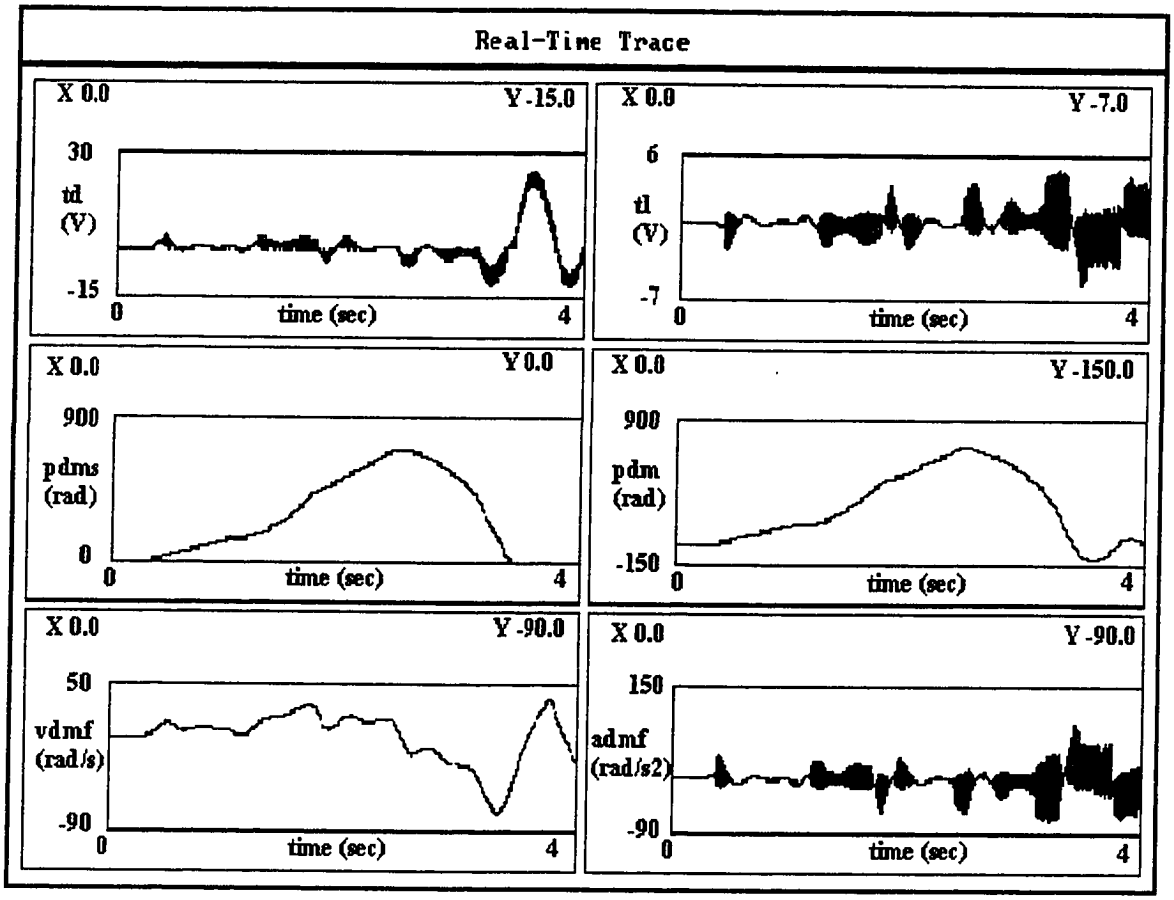


Figure 6.6 Output with increased inertia

The next two experiments were conducted to see the effect of viscosity changes. Those experiments can be described as monitoring a mixer mixing oils with three different viscosities. Initial experimental results can be seen as mixing oil of medium viscosity.

For the next experiment, we change initial viscosity to be 10 times smaller while other values remain the same as for the initial experiment. The following values were used:

$J_{sim} = 0.0001 \quad [kgm]$

$B_{sim} = 0.00001 \quad \left[\frac{Nm}{rad/s}\right]$

$F_{dry} = 0.0000234 \quad [Nm]$.

Figure 6.7 shows the results.

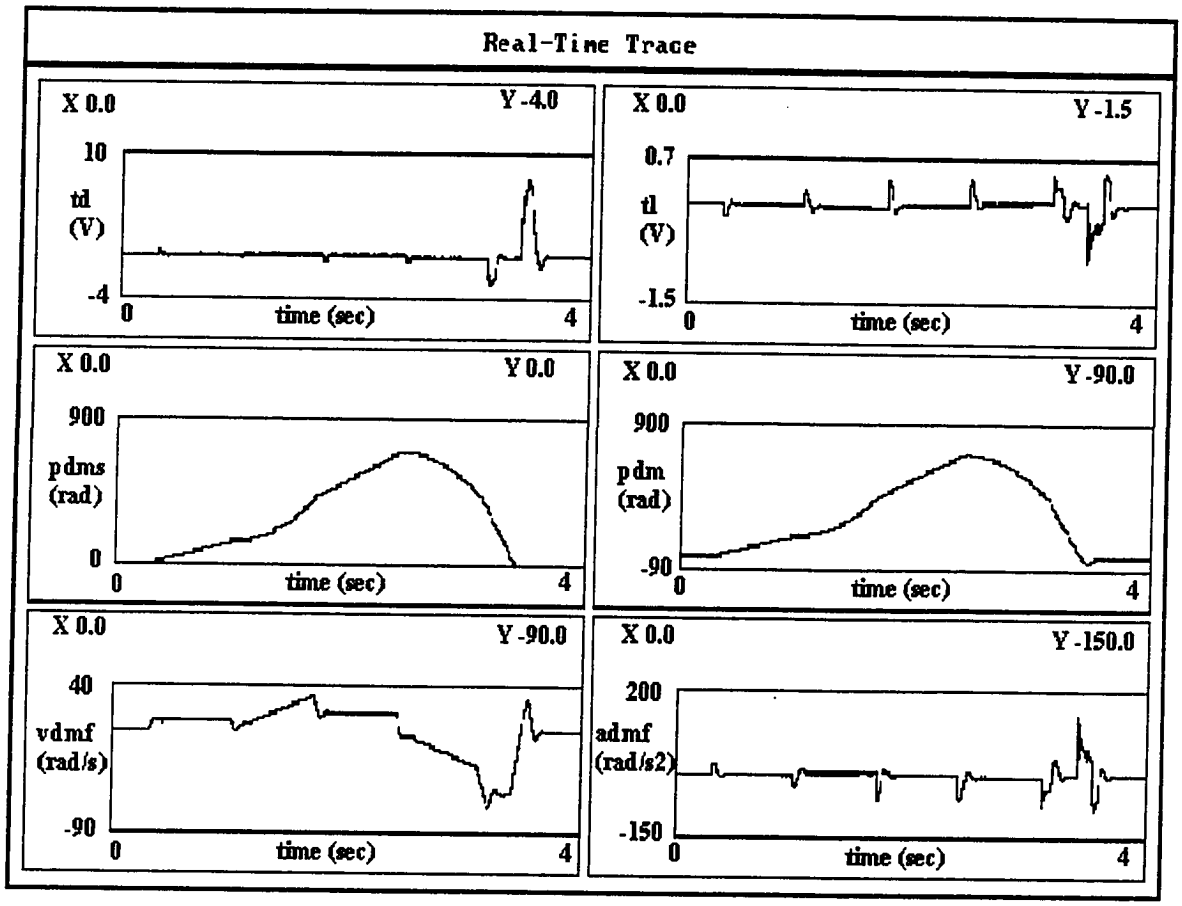


Figure 6.7 Output with reduced viscosity

Compared to the initial output, this load with a 10 times lower viscosity produces a very similar actual position, with a difference when the system stops. This system is harder to control since it goes more easily to negative position values. That is understandable if we imagine a mixer mixing water instead of heavy oil with a high viscosity. The velocity is almost the same compared to the velocity of original experiment. The largest difference occurs at the end when the system is about to stop. The acceleration shows a constant value throughout the experiment, with smaller peaks that appear when the input changes the section. The signal generated for the load simulator reflects the acceleration.

The last experiment for this preliminary analysis was conducted for the system with a higher viscosity than used for original experiment. This can be seen as a mixer mixing high viscosity oil. The values used in formula were the following:

$J_{sim} = 0.0001 \quad [kgm]$

$$B_{sim} = 0.001 \quad \left[\frac{Nm}{rad/s}\right]$$

$$F_{dry} = 0.0000234 \quad [Nm].$$

Using the same conditions, the following results, shown in Figure 6.8, were obtained.

The system response shows larger differences between position signals. The actual position has only three visible curves, two rising and the falling one. Due to the high viscosity, the PD controller is not able to create a signal sufficient to follow the desired position. The velocity of the system is significantly smaller and shows uniformity in three levels. The acceleration shows instability in the areas where the system creates biggest errors. The deceleration is slow and the actual system stops later than it was designed according to the input signal.

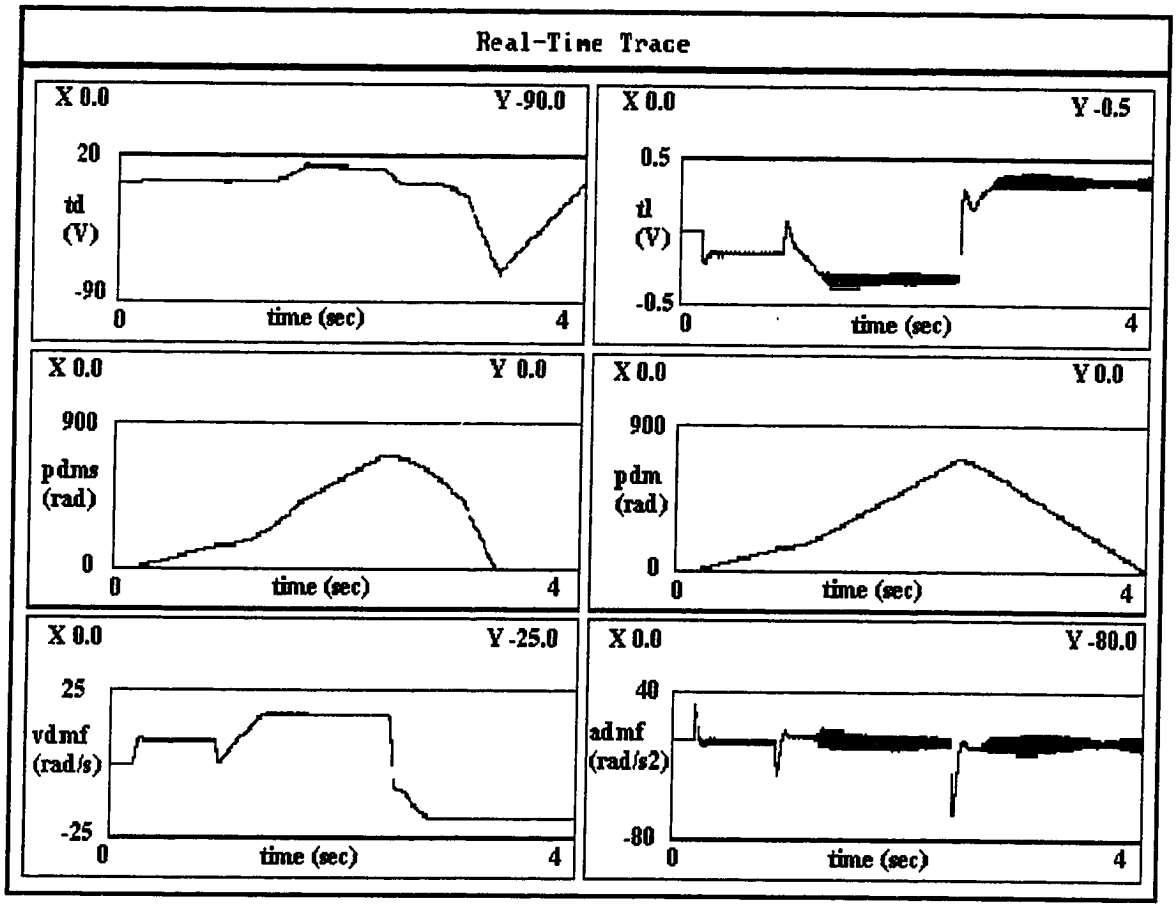


Figure 6.8 Output with increased viscosity

The results from Figures 6.4-6.8 show that the equation for linear load, derived in Chapter 3, can be used as an additional tool with any physical load simulator. The results show the expected behavior for various loads as their inertia and viscosity changes. Further modeling may be required for critical stages of the simulation but in general basic responses are obtained and they confirm theoretically stated relationships.

6.4. Open loop experiments

6.4.1. Linear load

The next set of experiments was designed to illustrate the performance of the HIL method in physically simulating real loads. In the beginning, the torque control feedback was not included in the control loop. The measured signal from encoder was used to calculate velocity and acceleration of the system for model based simulation of the load. This principle was described in detail in Chapter 5.

The values for the load were not arbitrary as they were taken from the real load. The aforementioned Parker motor was used to represent the linear real load. The rotor of the motor has all the characteristics of the inertial viscous load and if the motor has no electric power supplied, it can be considered as a passive load. Moreover, the motor's rotor already has bearings and is connected to the shaft, which can be considered a real load to the rest of the experimental setup.

Two different position inputs were used in the experiments and they are presented in the Figures 6.1 and 6.2. The output from the real load was recorded, for both inputs before physical simulations started. The formula used in the program to calculate an additional physical simulation torque for the physical simulator can be found in the Chapter 3, derived as:

$$\frac{(J_{sim} * s + B_{sim}) * K_s}{J_l * s^2 + B_l * s + K_s} * \omega(s) - F_{dry}$$

The Pittman 9232 physically simulated the Parker motor and the data used for simulation are as follows:

$J_{sim} = 0.00002609 \quad [kgm]$

$$B_{sim} = 0.00025 \quad \left[\frac{Nm}{rad/s}\right]$$

$$F_{dry} = 0.0000234 \quad [Nm].$$

These data were obtained knowing the real data for both motors, real load and physical simulator, by using equations 3.41 3.42 3.43.

6.4.1.1 Ramp input

The results of the experiment with the Parker real load using first ramp input are shown on the Figure 6.9. The second window, which records the additional torque named tl, shows that the Parker motor was not an active but rather a passive load, since no signal was sent to the motor. The velocity characteristic, which is considered as an output of the system, is shown in the first window of the third row and is named vdmf.

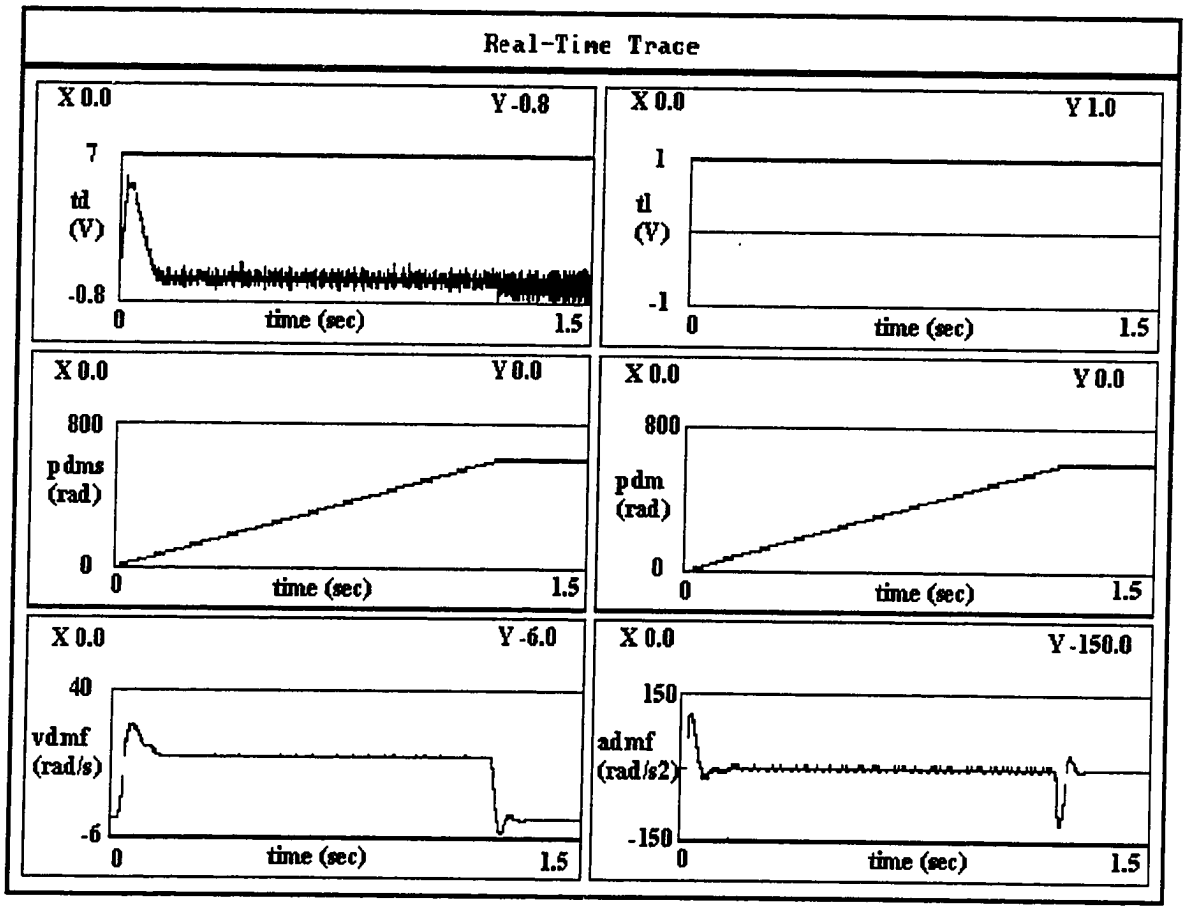


Figure 6.9 Parker motor used as a real passive linear load with ramp input

It shows steady value around 22 $\left[\frac{rad}{s}\right]$ after the transition. The system stops suddenly and that causes the velocity to drop below zero mark showing that the system has significant inertia. The cut-off frequency used to filter data was 1 Hz for the input frequencies of 0.159 Hz and 0.796 Hz, and 5 Hz for the input frequency of 1.592 Hz.

The acceleration, which is shown in the right window of the last row, has two peaks, one at the beginning and one at the end of the cycle. The first peak corresponds to the relatively high inertia of the rotor and responds to the step input at the beginning of the experiment. The second peak is the result of the opposite setting. The motor is running and then suddenly the signal drops to zero. The inertial load cannot stop immediately and it results in negative acceleration.

The physical simulation of the Parker rotor as a real load is shown in the Figure 6.10.

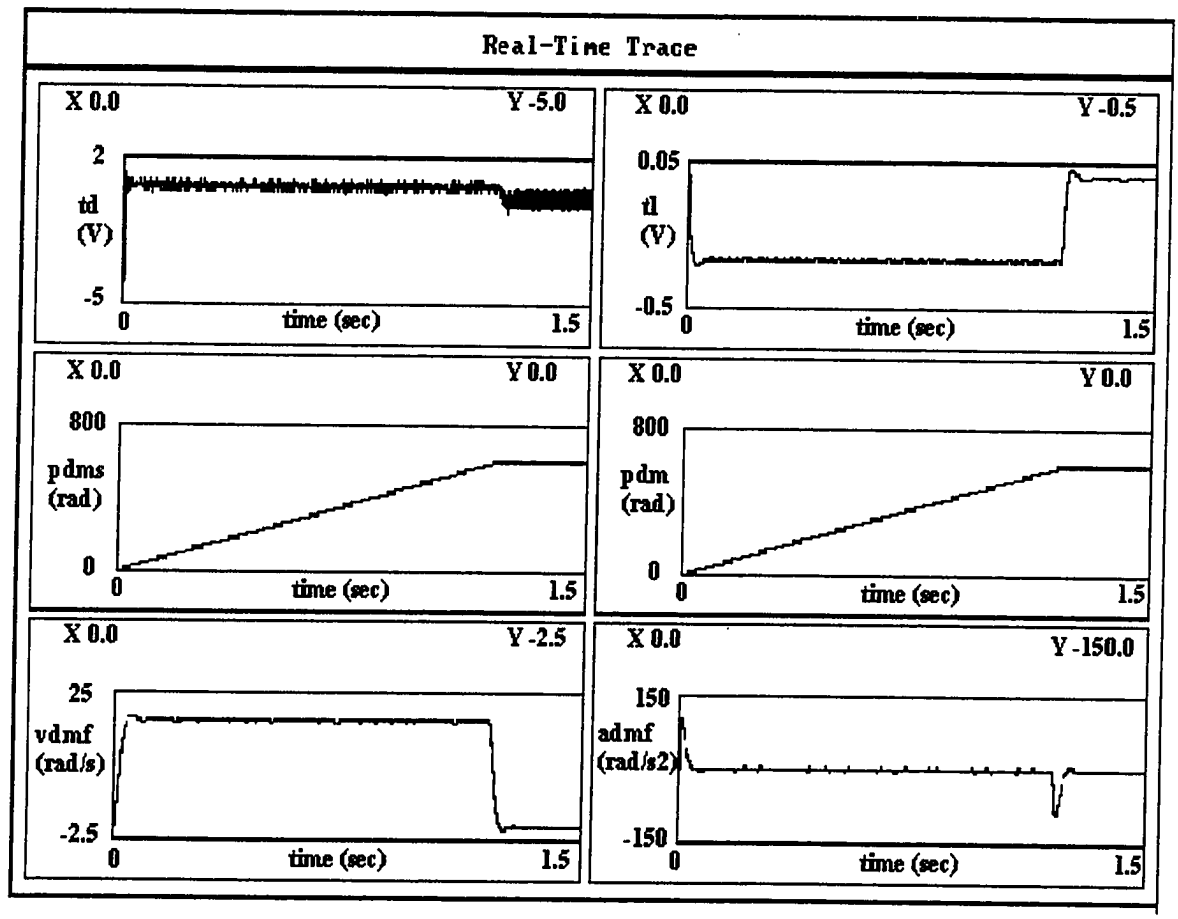


Figure 6.10 Physical simulation of the Parker linear load with ramp input

The physical simulator, in this case Pittman 9232, was used to simulate a different motor, Parker motor, using its inertia and viscosity combined with the additional current calculated with previously mentioned formula based on the signal received from the encoder. This additional signal is shown in the second window and is denoted tl. The velocity characteristic shows great similarity with the original one, which is experimentally proving the concept of physical simulation using HIL and effort-flow theoretical base. The acceleration is similar as well, showing two peaks at the same place as the original one. The difference between two velocities is in the beginning part where physical simulation shows smaller overshoot. The reason for narrower peak in the simulated control signal at the beginning of the cycle is that actual position signal is "late" compared to the desired position signal. The same effect is noticed for all the rest of the experiments. Improved models could diminish that effect and the results of simulations would show the same "wider" control signal.

6.4.1.2 Combined step and sinusoidal input

Results for three different frequencies to the combinational input, explained in equation 6.2 were recorded: 0.159 Hz, 0.796 Hz and 1.592 Hz. The purpose of that was to see if it is possible to simulate physically a real load under different frequencies and if the simulation looses accuracy for higher frequencies. For the experiments with a sinusoidal input, the same simulation data was used as for ramp input as well as the same filter cut-off frequency.

The first experiment with combined input was conducted for 0.159 Hz input frequency and results for the real load are shown in the Figure 6.11.

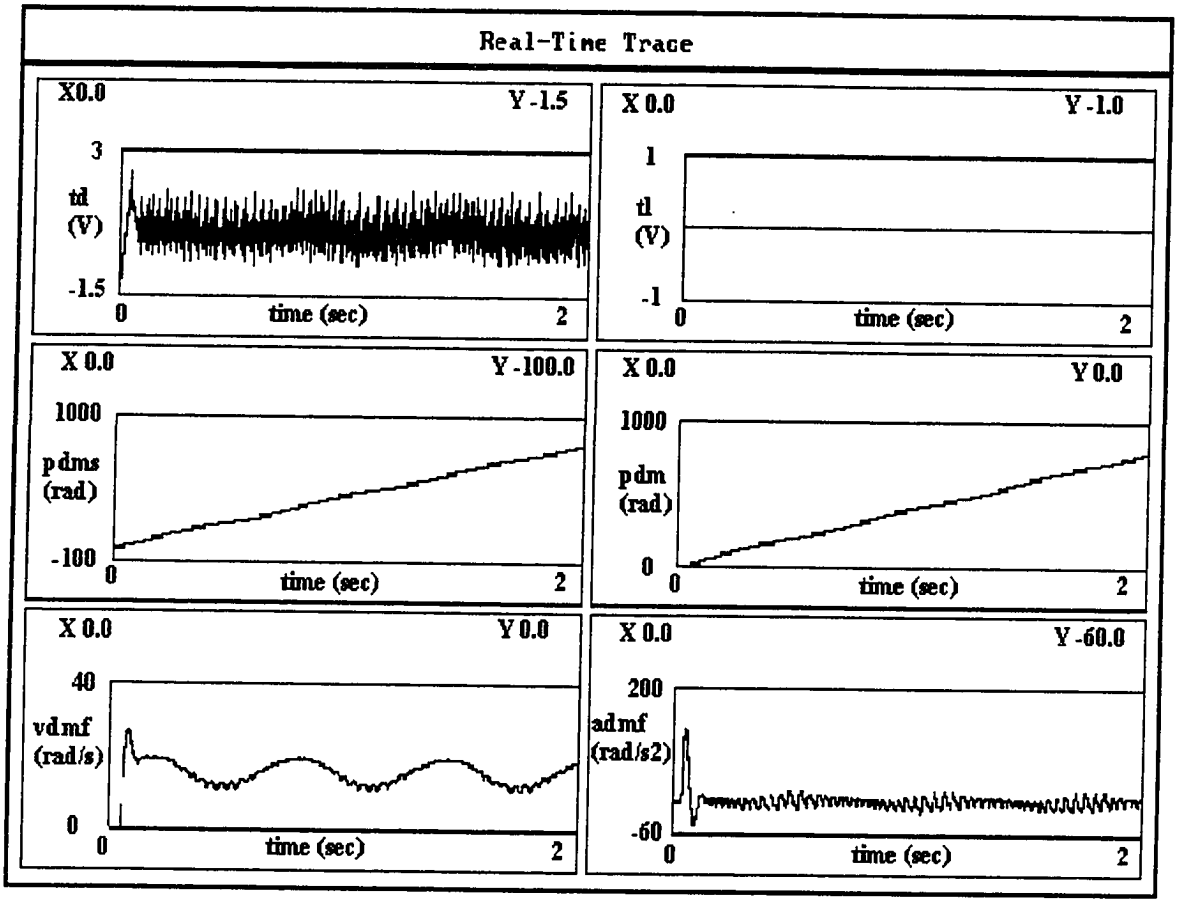


Figure 6.11 Parker motor used as a real passive linear load with combined input with 0.159 Hz frequency

The velocity characteristic follows the derivative of the input signal. The amplitude of the velocity depends on the inertia of the load, as was observed in the preliminary analysis section of this chapter. At the beginning of the cycle, sinusoidal velocity has a peak, which is the result of slower response of the position of the real load to the desired position.

The physical simulation of the Parker real load with sinusoidal input is shown on the Figure 6.12.

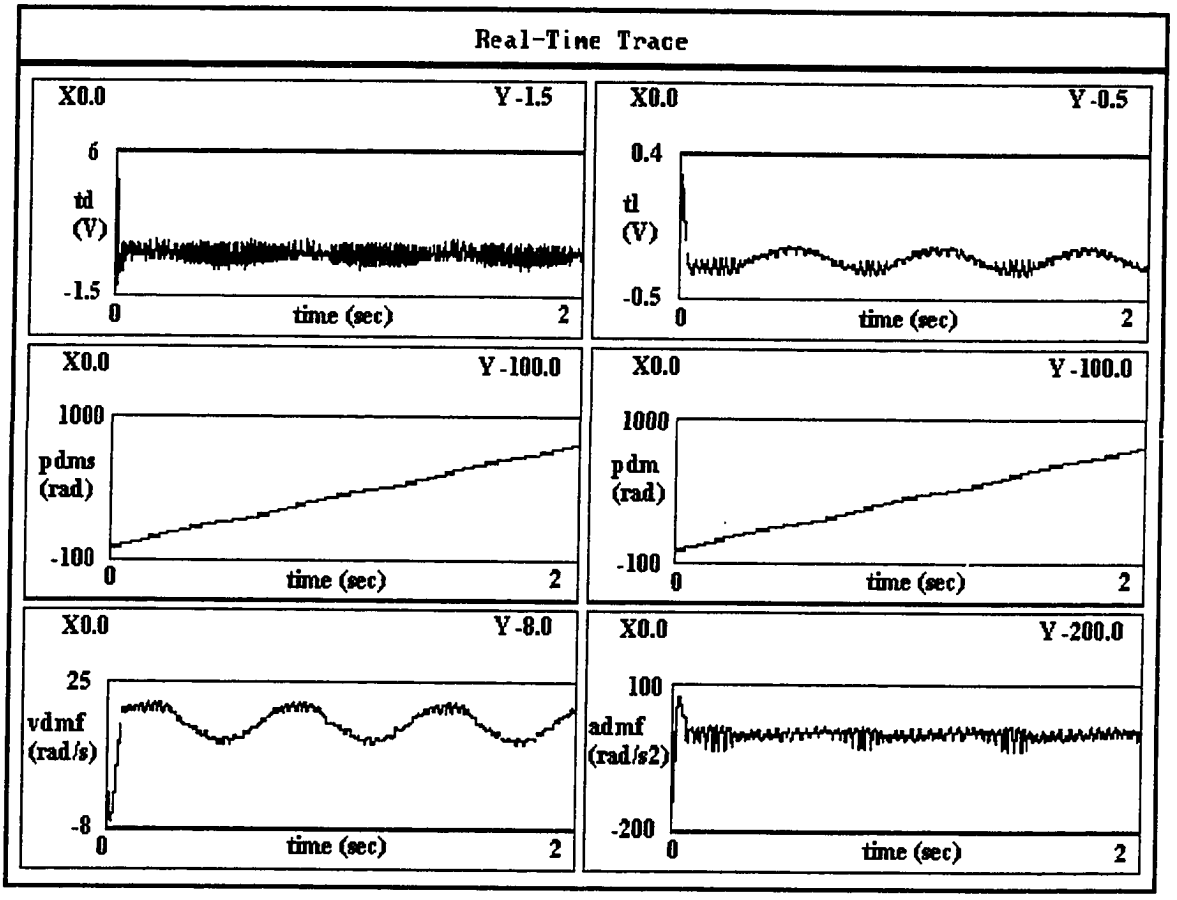


Figure 6.12 Physical simulation of the Parker linear load with combined input with 0.159 Hz frequency

The calculated signal tl used by simulator has a sinusoidal shape. The velocity characteristic is, as expected, sinusoidal with the difference at the beginning when the physically simulated load has a jump. That can be explained as the result of step input and the calculation of velocity and acceleration, which are based on the previous set of information obtained by encoder. The velocity is not changing from that point directly into a sinusoidal curve, but rather as a sharp change, due to the initial transient response. The acceleration shows steady value with more significant oscillations when the velocity is at high peak values.

The next experiment was conducted for the input frequency of 0.796 Hz with all other data remaining the same. The results for the real load subject to the sinusoidal input of 0.796 Hz is shown on the Figure 6.13.

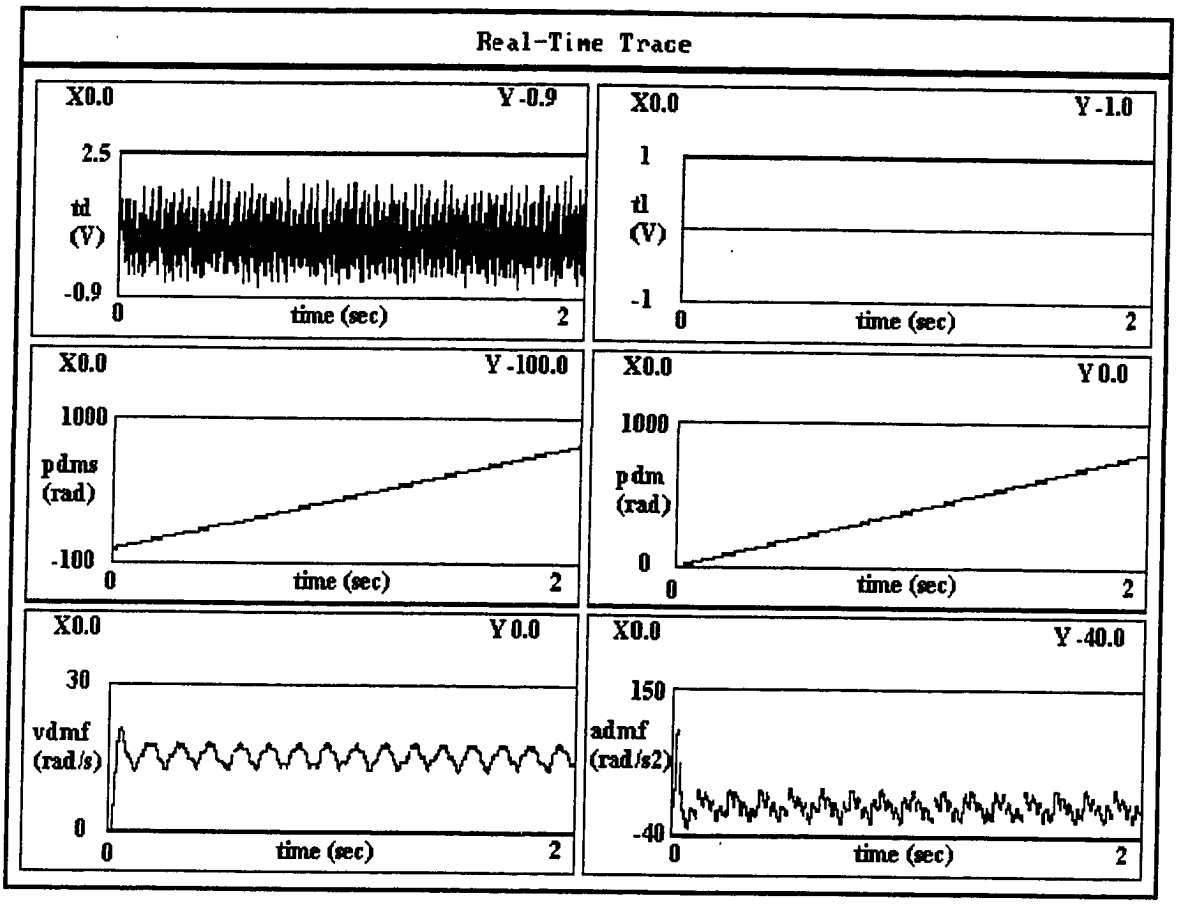


Figure 6.13 Parker motor used as a real passive linear load with combined input with 0.796 Hz frequency

While the frequency changed, the real load response looks similar. The velocity, with the smaller amplitude, follows the frequency of the input. In addition, the same problem with the step start can be noticed, producing the high-sloped response with small overshoot. Moreover, the acceleration seems to be more sinusoidal than the acceleration of the previous experiment with 0.159 Hz frequency.

The next figure, 6.14 shows the results for the physical simulator simulating the real load with the input with 0.796 Hz frequency. The velocity characteristic has almost identical shape but the initial peak is slightly lower than the original one. The transient response on the input does not have an overshoot; it is more damped. The signal sent to the simulator shows, again, sinusoidal envilope, which changes in accordance with the input. The acceleration looks identical as the one obtained for the real load shown in Figure 6.13.

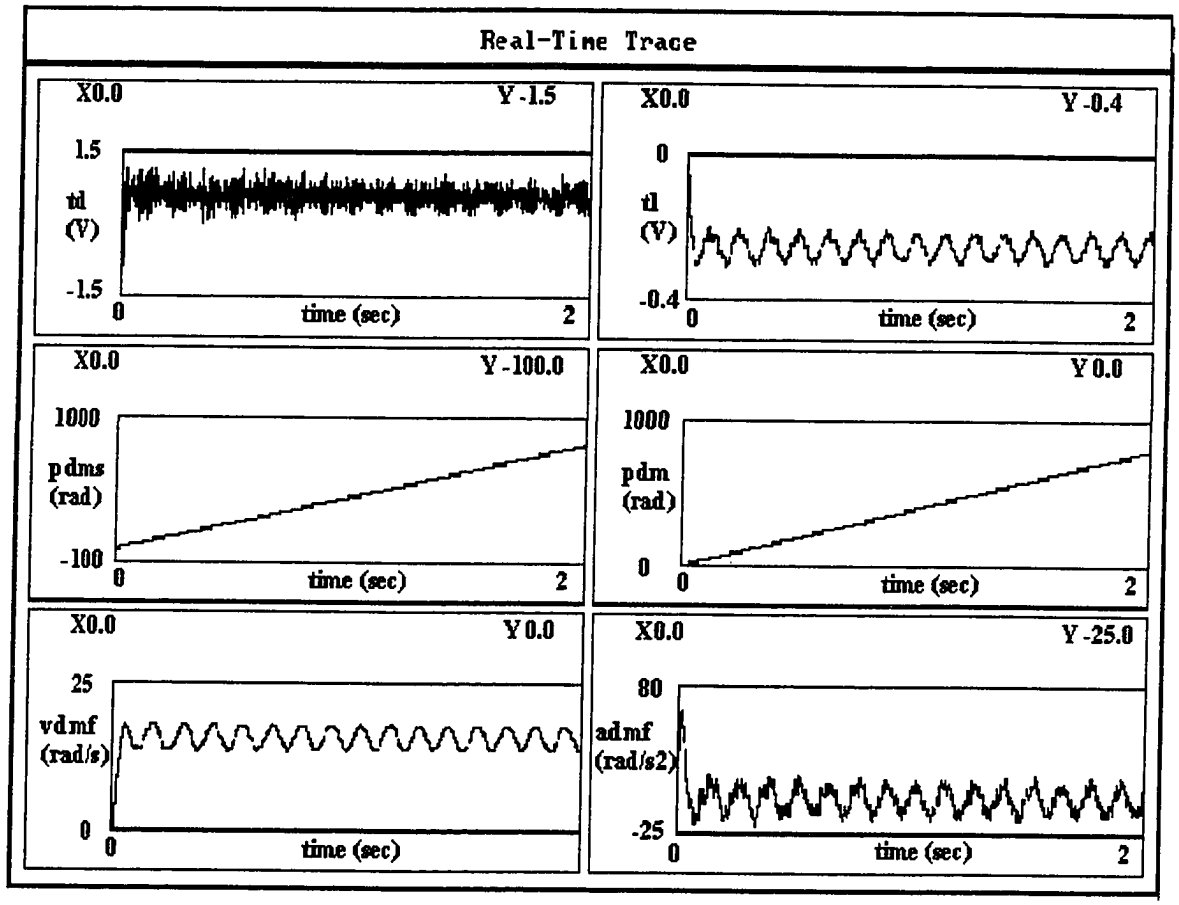


Figure 6.14 Physical simulation of the Parker linear load with combined input with 0.796 Hz frequency

The last experiment was for the sinusoidal input with 1.592 Hz frequency and the results are shown on the Figure 6.15. The real load again showed a velocity that follows the input frequency and the step response has a small overshoot at the start.

Figure 6.16 shows the physical simulation of the real load for the sinusoidal input with 1.592 Hz frequency. The similar problem encountered for 0.796 Hz appears again. The response on the starting step portion of the input is different for simulator from for the real load. This time even more visible that the simulator response is over-damped while the real load shows a certain overshoot. The differences can be explained by less accurate and/or not modeled dynamic at higher frequencies. A more elaborate system modeling and identification can address this problem.

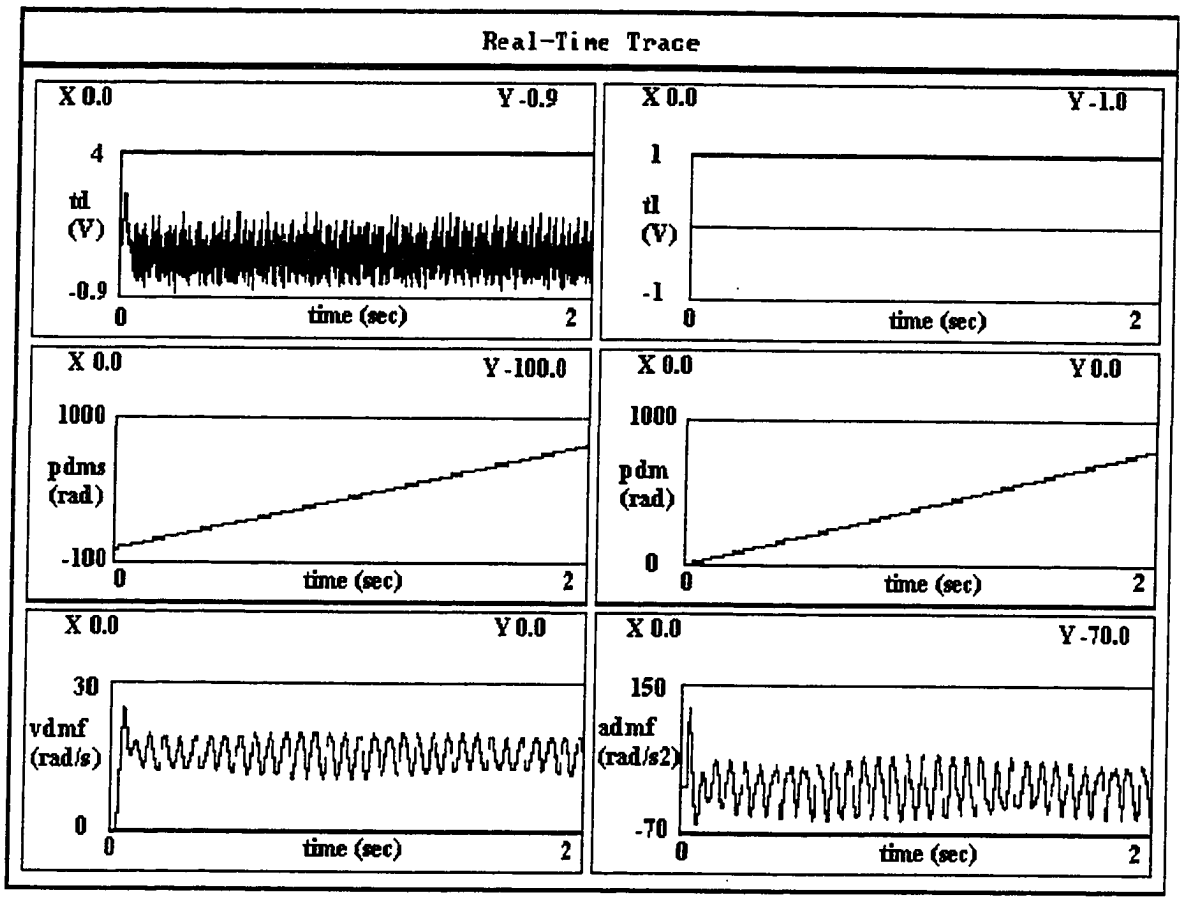


Figure 6.15 Parker motor used as a real passive linear load with combined input with 1.592 Hz frequency

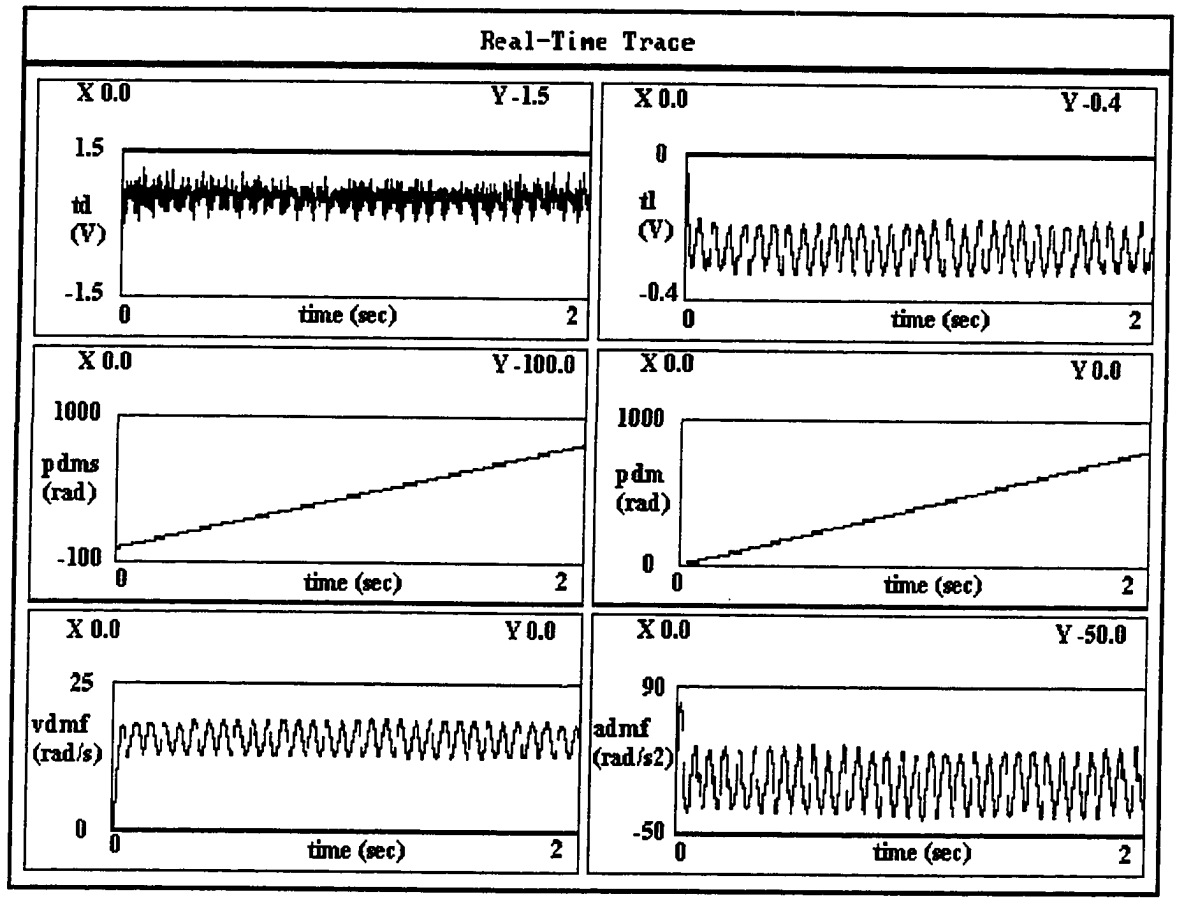


Figure 6.16 Physical simulation of the Parker linear load with combined input with 1.592 Hz frequency

6.4.2. Nonlinear load

This set of experiments was designed with the purpose to illustrate the performance of physical simulations using HIL experimental setup for a nonlinear load. The load used was a fan whose approximate inertia was calculated using the formula

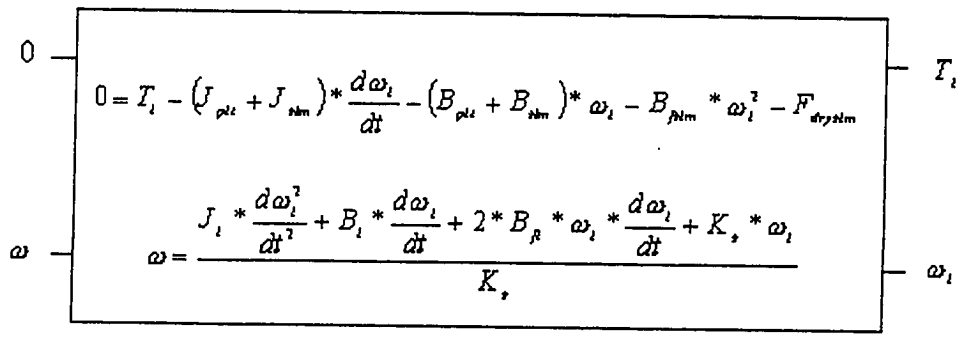
$$J = m * r^2$$

and the viscous values for B_l and Bf_l were obtained experimentally using the trial and error method, trying to approximate the curve obtained from the real load.

As mentioned in the Chapter 3, the fan torque can be modeled as

$$T_l = J_l * \frac{d\omega_l}{dt} + B_l * \omega_l + B_{fl} * \omega_l^2$$

and after deriving the block of equations the following model can be used



This block was used to calculate additional torque applied to a physical simulator while simulating real load, and the relationship between two velocities.

The same set of formulas was used to conduct this series of experiments. The real load, in this case the fan, was connected to the experimental setup and the velocity characteristics was measured. Then, the physical simulator was used in an attempt to obtain the same results. Instead of attaching the fan, the Pittman9232 was used to physically simulate the real fan load. The simulation was achieved by using the motor and the calculated torque (current) based on previously mentioned formulas.

The PD controller controlled the actuator motor in order to achieve the desired position. In addition, two different inputs were used to show that the method could handle different disturbances. The inputs are the same as the ones used for linear load and they are described in detail at the beginning of the chapter.

The Pittman 9232 physically simulates the fan using the following data for all simulations:

$J_{sim} = 0.00021375 \quad [kgm]$

$B_{sim} = 0.000035 \quad \left[\frac{Nm}{rad/s}\right]$

$B_{op} = 0.00000383 \quad \left[\frac{Nm}{rad/s}\right]$

$F_{dry} = 0.000000156 \quad [Nm].$

The Butterworth filter used to filter received signal was set on cutoff frequency of 1 Hz for the ramp input and input frequencies of 0.159 Hz and 0.796 Hz and on 5 Hz for the input of 1.592 Hz.

6.4.1.1 Ramp input

The experimental results for real nonlinear fan load, using linearly increasing input, are shown in the Figure 6.17. The velocity characteristic of the system shows large disturbance at the beginning and gradual settling until output becomes constant. After the position reached constant desired value, the system stopped but not immediately due to the large inertia. Acceleration shows nonlinearity at the beginning and at the end of the cycle. Such disturbances are reflected in the PD controller's signal, which lead to oscillations at the start and the end of the cycle.

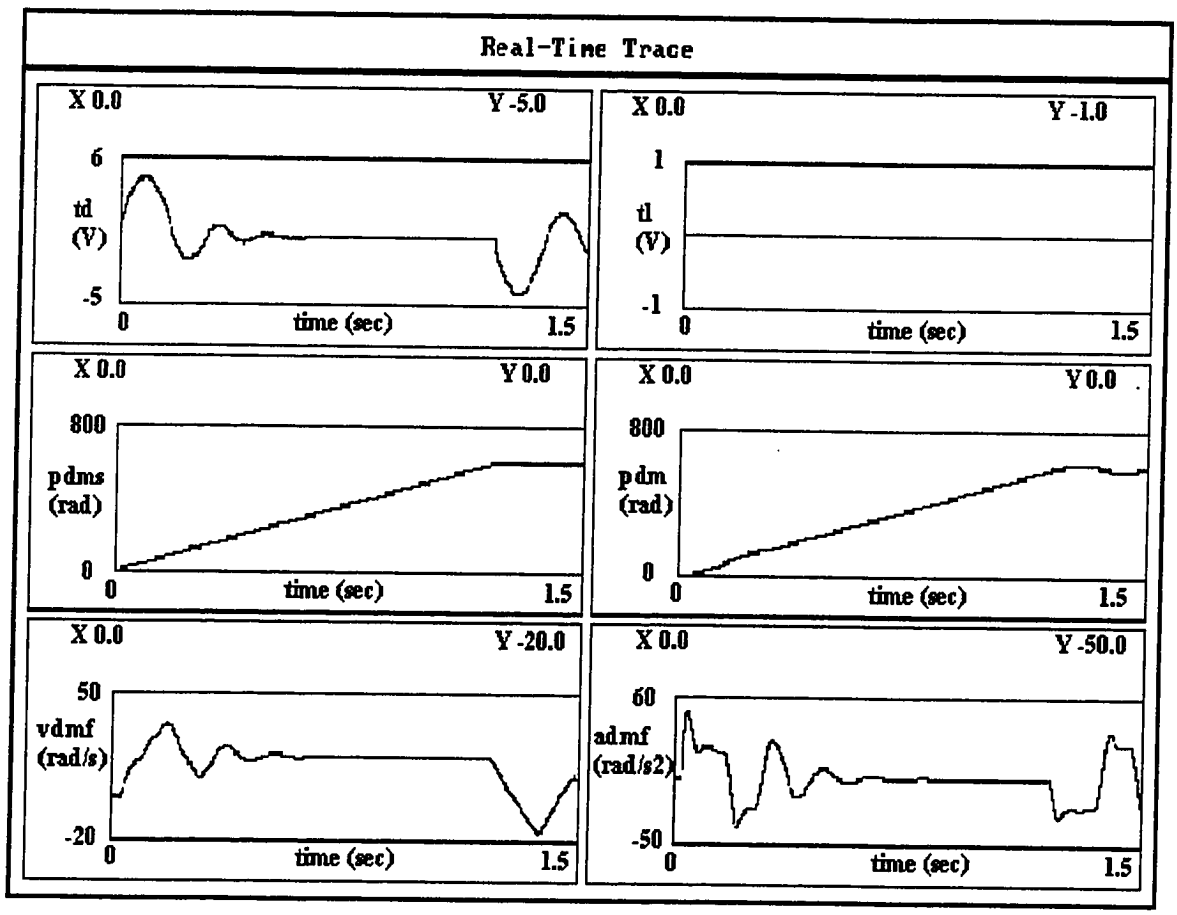


Figure 6.17 A nonlinear fan used as a real passive load with ramp input

The physical simulation of the same fan load with linear input is shown on the Figure 6.18. The signal generated to be added to the physical simulator is shown in the second window of the first row. It shows a significant peak at the beginning of the experiment due to the step input, as well as at the end, when system is about to stop. The velocity characteristics of the simulated system follows the original curve, with the difference of smaller disturbances at the beginning and at the end where simulated system due to lack of deceleration modeling, stops faster, different from the original system which decelerates slowly. The problem with simulated system occurs when the input becomes zero. The actuator stops, and since encoder shows no move, calculated velocity is zero and since that is the base for calculation of the additional torque for physical simulator, the system stops. As a result, the acceleration characteristic has an additional negative peak at the end of the cycle.

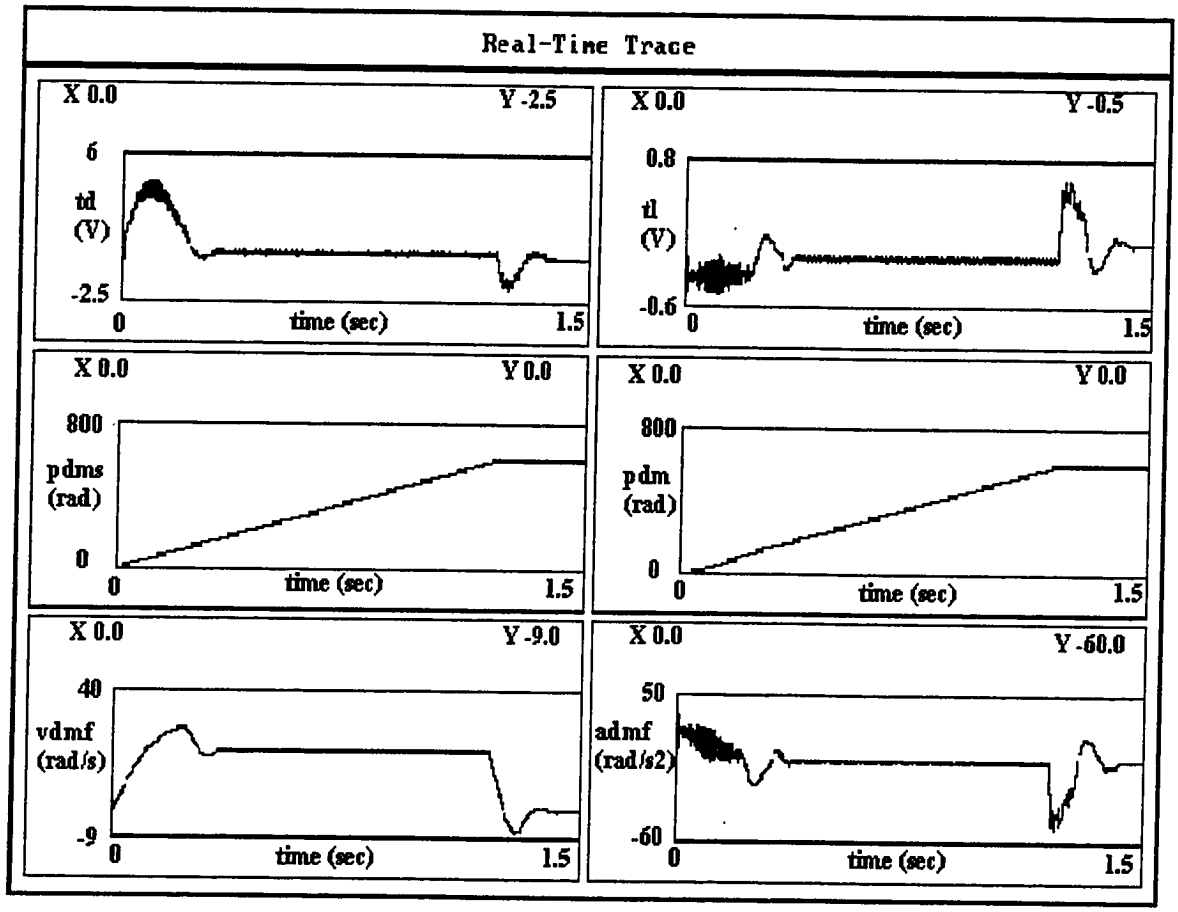


Figure 6.18 Physical simulation of the fan nonlinear load with ramp input

6.4.2.2 Combined step and sinusoidal input

The next three experiments were conducted for the same experimental setup with nonlinear fan load. Again, the same equation block was used to calculate the additional torque. The only difference was the input, which was changed to be combination of step and cosine functions with three different frequencies: 0.159 Hz, 0.796 Hz, and 1.592 Hz.

The results for the real nonlinear fan load are shown in the Figure 6.19. As expected, the load with two viscous friction constants, out of which one is multiplied by square of the velocity, shows higher nonlinearities at the beginning when the input signal steps up. The same effect is visible on the acceleration curve during the transient response time.

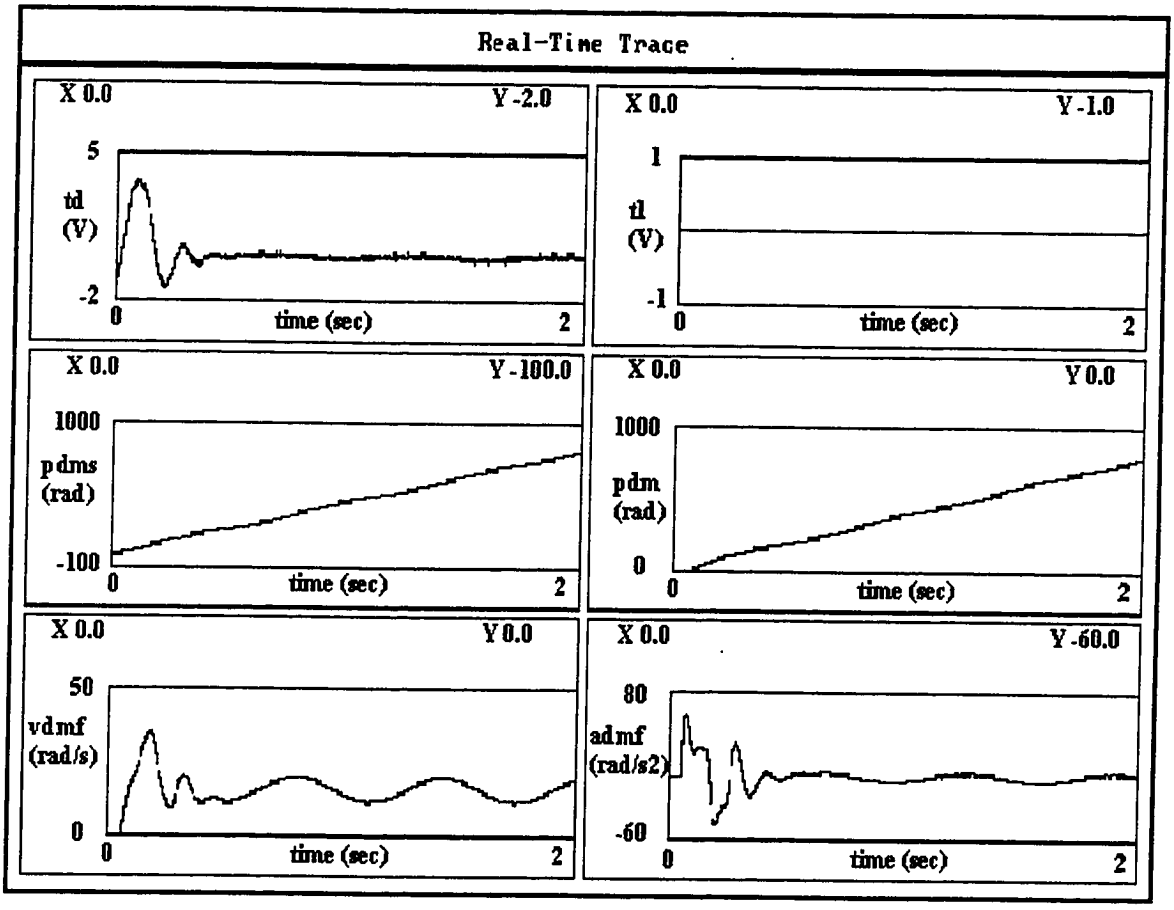


Figure 6.19 A nonlinear fan used as a real passive load with combined input with 0.159 Hz frequency

The physical simulation of the nonlinear fan load subject to the 0.159 Hz sinusoidal input is shown in the Figure 6.20. Pittman 9232 motor accomplished the physical simulation in addition to the torque signal sent to the simulator, which was calculated by the same formula used for linear input.

The velocity characteristic follows the original curve closely, which confirms the validity of the concept. The difference is in number of nonlinearity peaks, which is smaller for the simulated curve. This is due to the imperfection of the model. The signal calculated and sent to the physical simulator shows oscillations only at the beginning due to the acceleration.

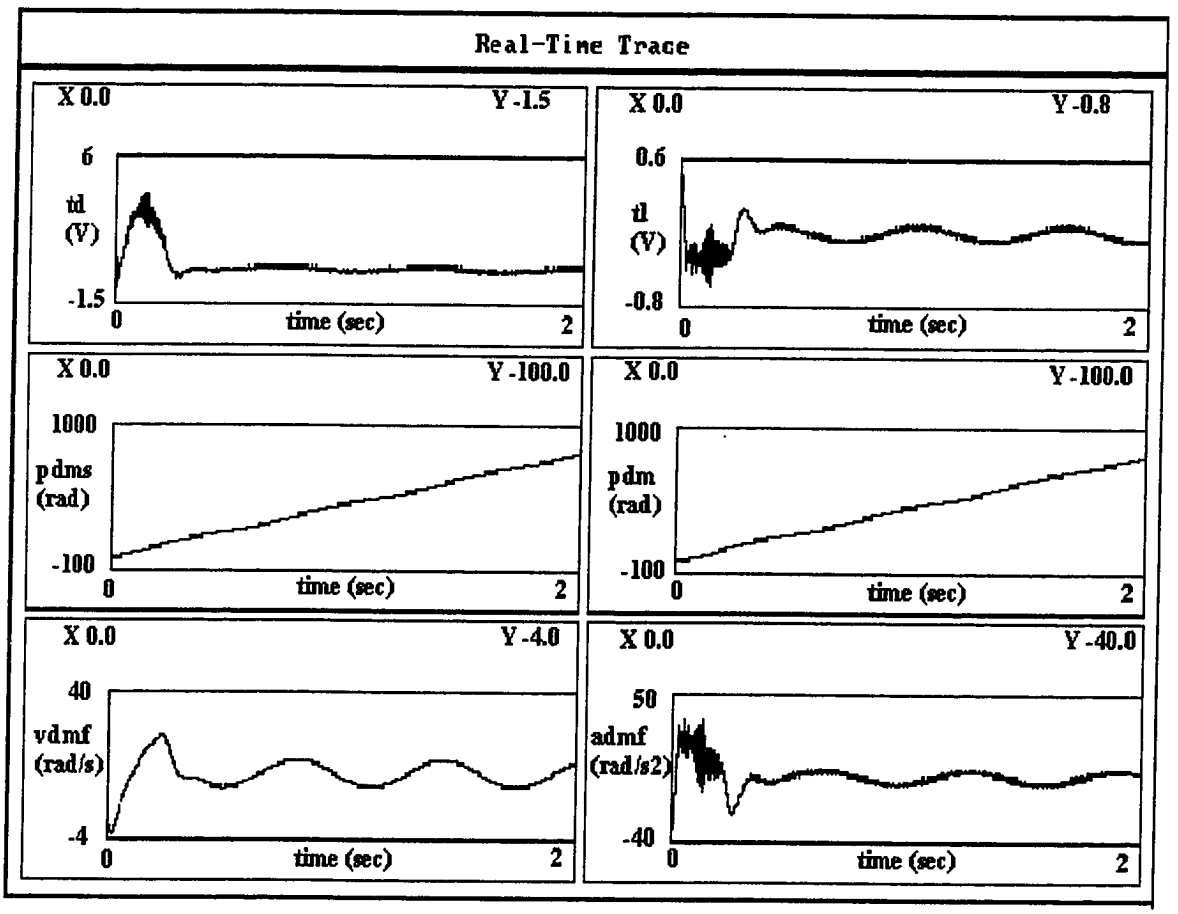


Figure 6.20 Physical simulation of the fan nonlinear load with combined input with 0.159 Hz frequency

The following experiment differs from the previous one only in frequency of the input. This time it was set to be 0.796 Hz. The real fan load was connected to the setup to

obtain the real velocity characteristic, which is shown on the Figure 6.21. The velocity characteristic is similar to the one with 0.159 Hz input frequency, with the difference in the velocity amplitude due to the different constant used for frequencies higher than 0.159 Hz, which was explained at the beginning of the chapter. The input starts again as the step function and the response shows overshoot as expected. The acceleration has a peak at the beginning of the cycle again due to high inertia and step input.

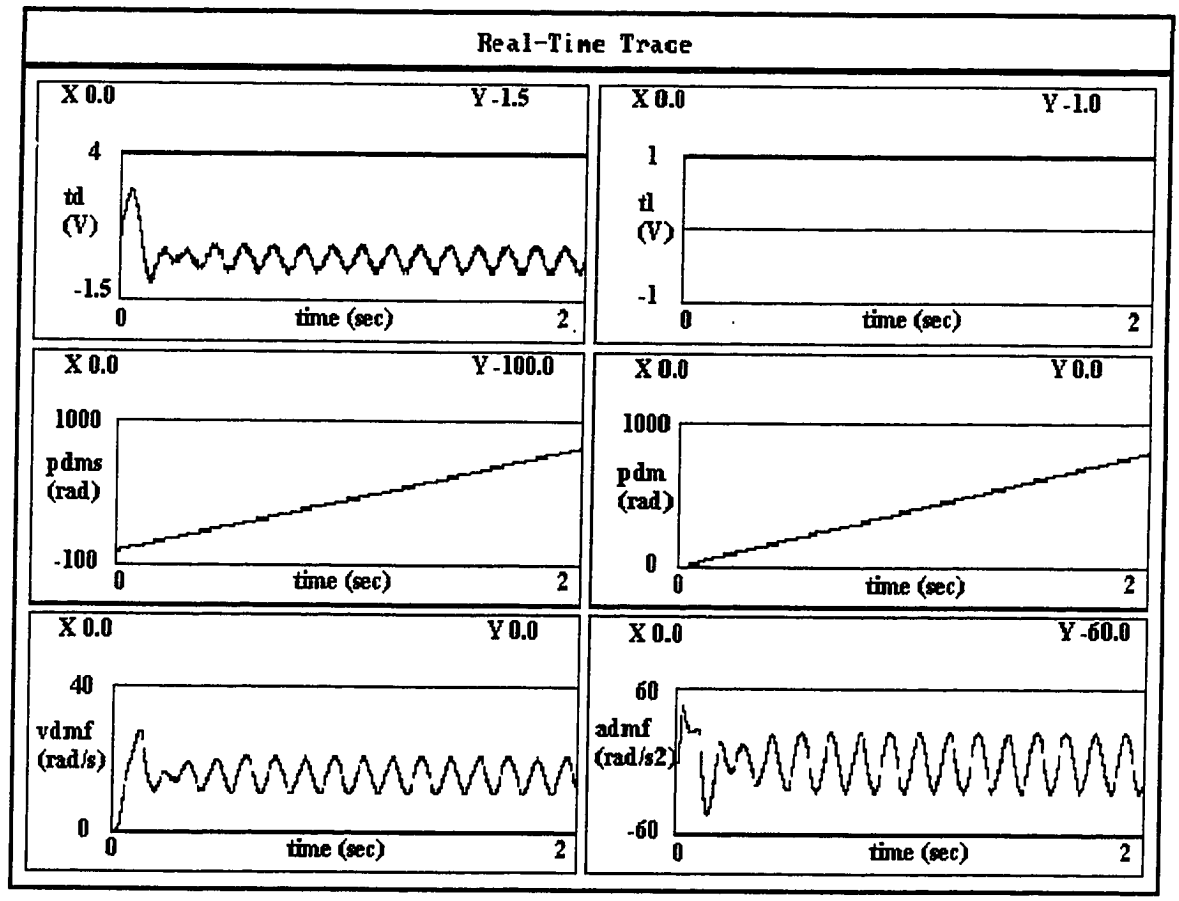


Figure 6.21 A nonlinear fan used as a real passive load with combined input with 0.796 Hz frequency

The physical simulation of the nonlinear fan load with 0.796 Hz input is shown in the Figure 6.22. Similarly, to the simulation when the system had input of 0.159 Hz, the calculated signal for the physical simulator shows instability in the area of the step input and later it becomes much clearer, but shows small amplitude. Even though the velocity characteristic, which is our main concern, follows the original real load curve, the

acceleration is very different. It is lower in the value and mostly in negative values. In addition, it does not have clear peaks at the beginning, but rather shows a smaller instability. We should remember that this experiment was set for the nonlinear load and this can result in reduced modeling accuracy. The derived set of formulas creates certain offsets for the acceleration signal, but the signal that is of interest is correct.

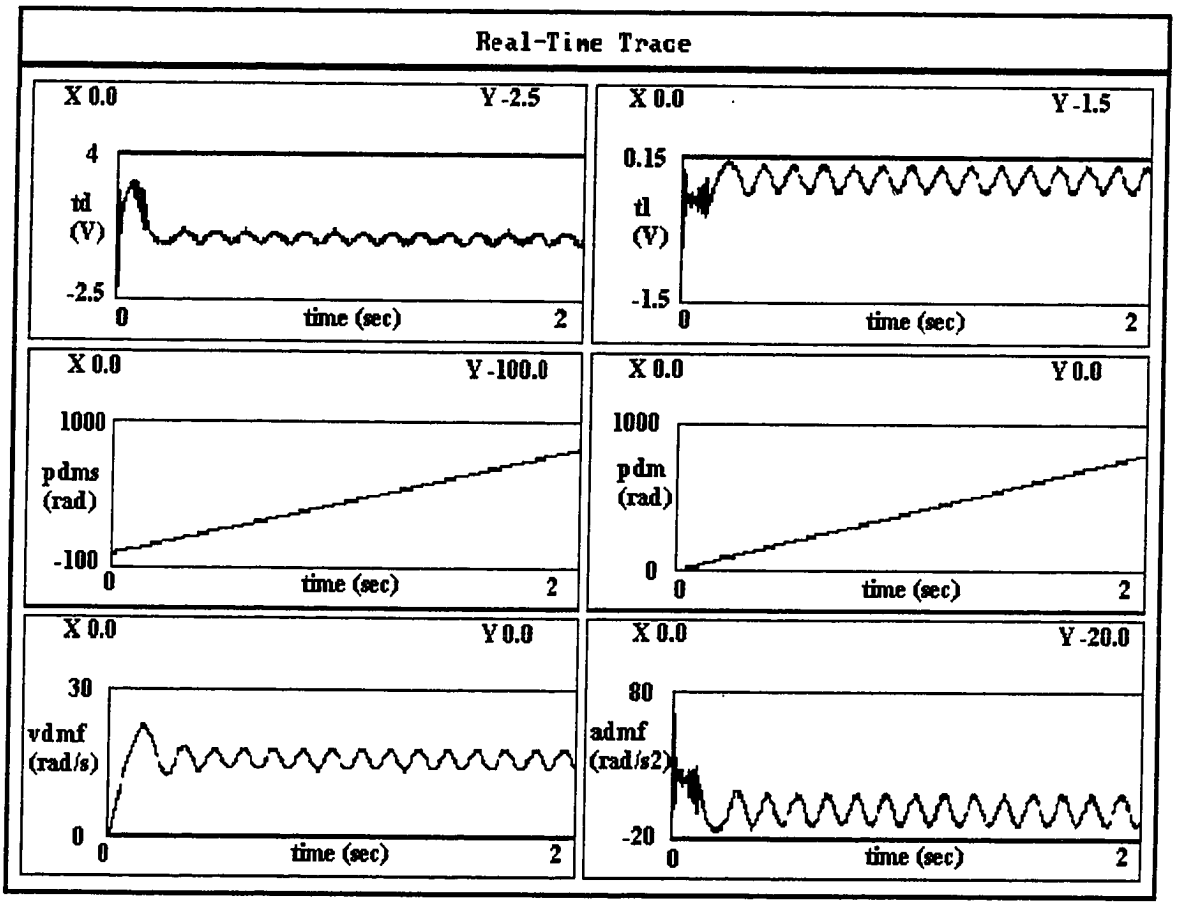


Figure 6.22 Physical simulation of the fan nonlinear load subjected to the combined input with 0.796 Hz frequency

The last experiment from this series was conducted to prove that we could raise the frequency even more and still successfully physically simulate the real nonlinear load. This time the input frequency was set to 1.592 Hz and all other values were kept as they were before. The results for real fan load are shown in the Figure 6.23. The velocity characteristic resembles the one for 0.796 Hz input, but the amplitude of the sinusoidal part is even smaller. The response on the step part of the input is very similar to the one

for 0.796 Hz input including nonlinearity peaks at the very beginning. The acceleration has a peak at the beginning of the cycle and then uniformly follows the input frequency.

The physical simulation of the real load with 1.592 Hz sinusoidal input is shown in the Figure 6.24. The simulator input, calculated as an addition torque to the Pittman9232 passive rotor characteristic, is very similar to the input created for 0.796 Hz experiment. The instability is visible at the beginning of the cycle and then a smooth sinusoidal curve follows. Similarly, the acceleration has no high starting peak, which can be noticed on the real acceleration curve. The characteristic acceleration nonlinearity noticed at the beginning of the cycle is reflected in the signal generated for the simulator. The main result, the velocity characteristic follows very well the original velocity.

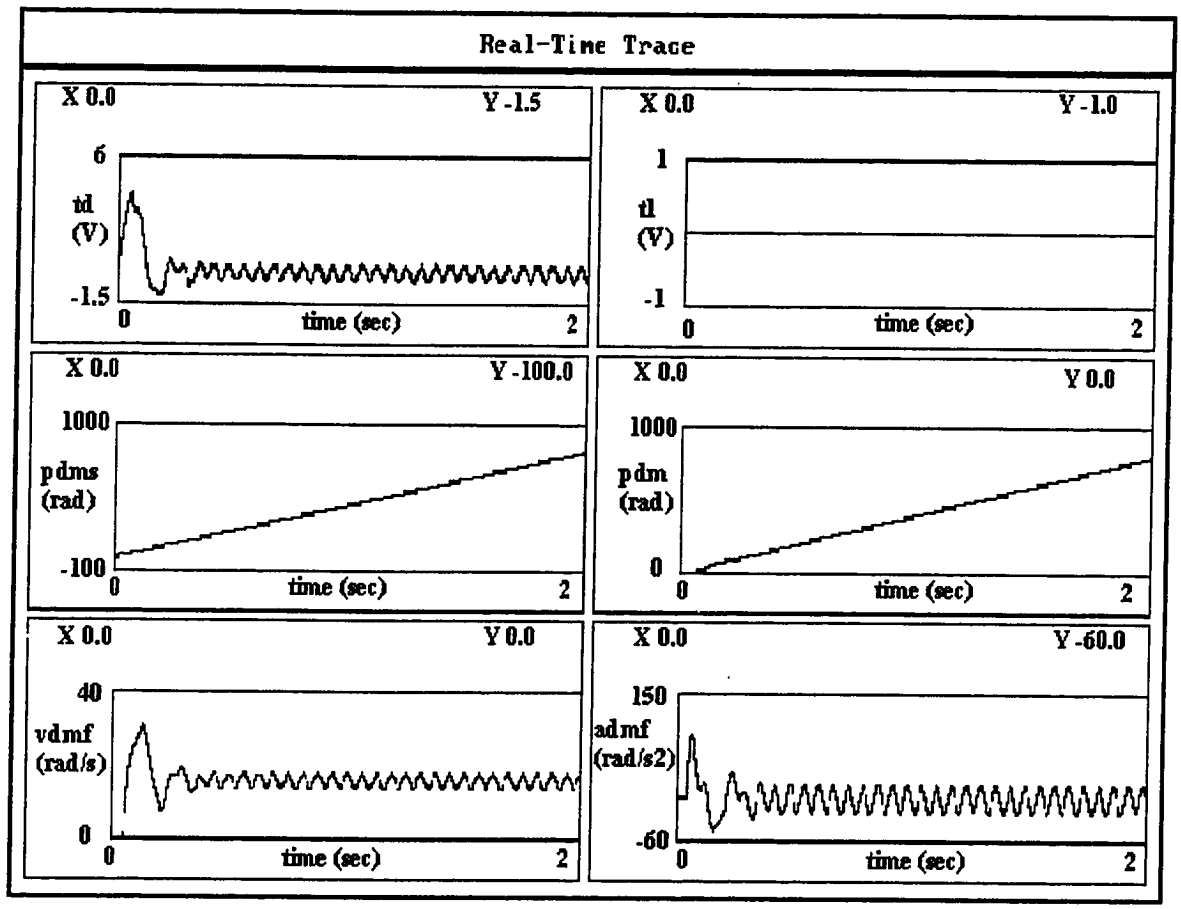


Figure 6.23 A nonlinear fan used as a real passive load with combined input with 1.592 Hz frequency

It is possible to conclude that in general, the formula used to calculate additional torque signal for nonlinear load physical simulator satisfies all criteria. The signal generated by the formula creates necessary nonlinear addition, which added to the linear rotor of the simulator, creates good match with the real nonlinear load characteristics.

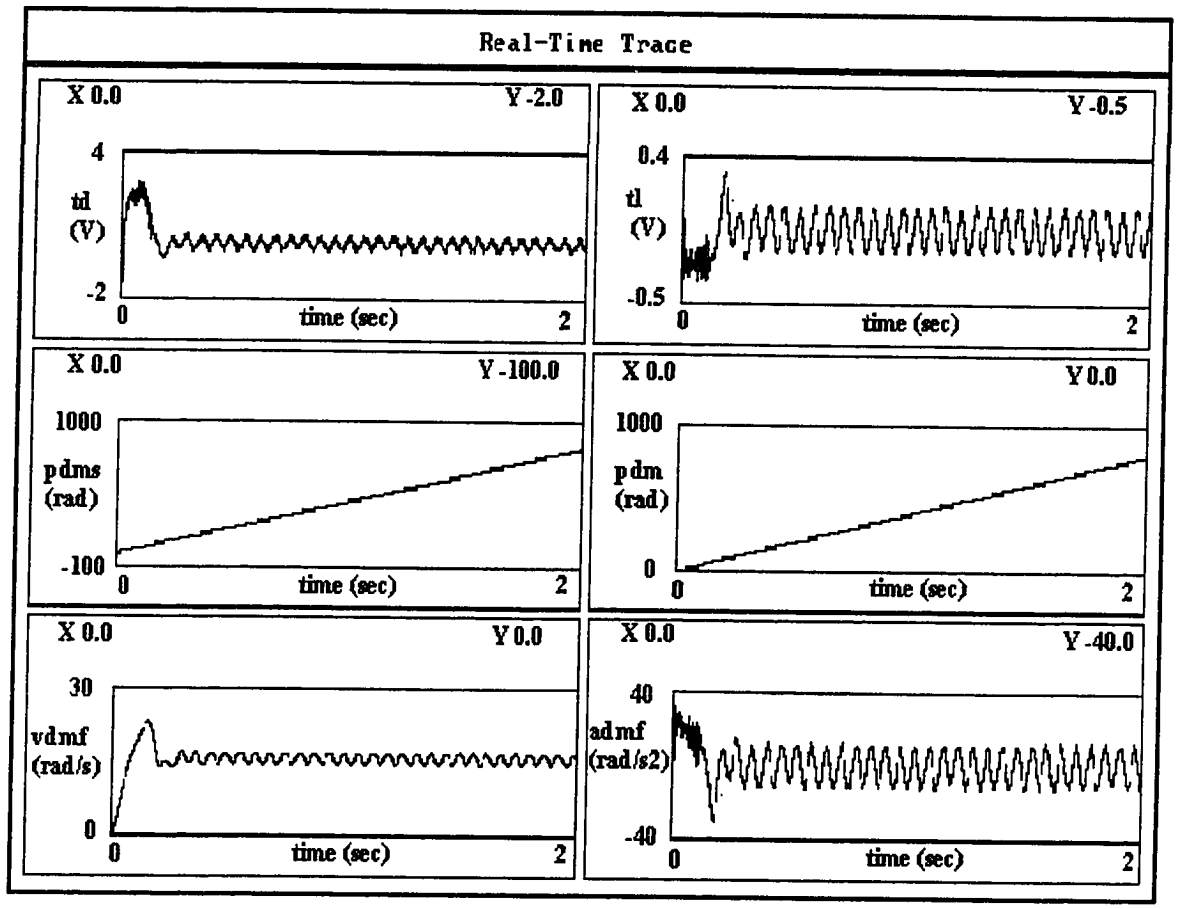


Figure 6.24 Physical simulation of the fan nonlinear load subjected to the combined input with 1.592 Hz frequency

6.5. Closed loop experiments – current feedback control

The next set of experiments was conducted on the same experimental setup with same sets of inputs and loads. The difference is in the concept described as the HIL2 in Chapter 5 and shown in Figures 5.3 and 5.4. A current feedback control was incorporated into the setup to provide the load simulator amplifier with the feedback signal. All signals are

filtered for easier analysis using the digital filter set on cutoff frequency of 1 Hz for 0.159 Hz and 0.796 Hz input frequency and on 5 Hz cutoff for1.592 Hz input frequency.

6.5.1. Linear load

6.5.1.1. Ramp input

The same real linear load, Parker DC motor, was tested again, now using the new concept. The results of the experiments with the real loads will not be shown again. For the real load subject to the ramp input, see Figure 6.9.

The physical simulation of the Parker real load is shown in the Figure 6.25. The dynamic characteristics used for the simulation are the same as for open loop experiments and they are as follows:

$J_{sim} = 0.00002609 \quad [kgm]$

$B_{sim} = 0.00025 \quad \left[\frac{Nm}{rad/s}\right]$

$F_{dry} = 0.0000234 \quad [Nm]$.

The velocity characteristic resembles the curve obtained for the test with the real load. The initial overshoot is not as noticeable but the tendency is similar. With some modifications made to the parameters, it would be possible to obtain the similar overshoot. The acceleration shows the same profile, but again with less nonlinearities. Two peaks at the beginning and at the end are clearly visible, though much simpler.

Compared to Fig 6.9, the results from Figure 6.25 are however closer than those from Figure 6.10. This confirms that closed loop control for the load simulating motor gives more accurate simulations than its open loop control. According to the results, the system seems to be over-controlled, since PD controller on one side and generated signal for the loads simulator on the other side produce almost mathematically ideal output. The real load showed much more flexibility.

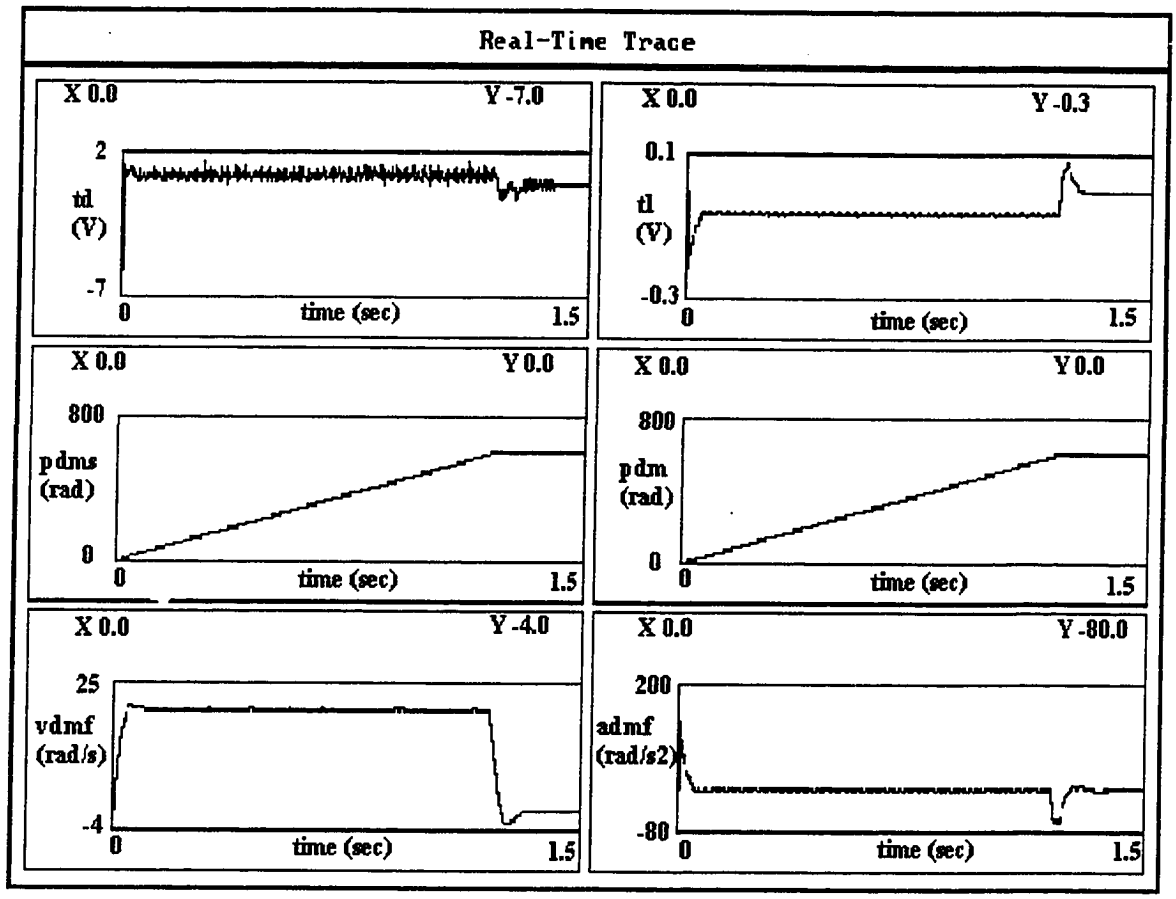


Figure 6.25 Physical simulation of the Parker linear load with ramp input

6.5.1.2. Combined ramp and sinusoidal input

The same dynamic characteristics were used to conduct experiments with a combinational linear-cosine input. Again, the experiments were designed with input frequencies of 0.159, 0.796 and 1.592 Hz. The cutoff frequency of Butterworth filter, to filter all signals, was set on 1 Hz and 5 Hz as explained earlier. First input used was with frequency of 0.159 Hz, and the results for the real load test are shown in Figure 6.11.

The physical simulation of the Parker real load, with the sinusoidal input is shown in the Figure 6.26. The same values for the parameters were used for the linear input were used again here. The sharp peak at the beginning of the velocity characteristic is almost reproduced, which proves the validity of chosen numbers and of course the second HIL configuration. Comparing to the results from the Fig. 6.11, the results from Fig. 6.26 are closer than the results from the Fig. 6.12, confirming again the advantages of closed loop versus open loop control for the load simulating motor.

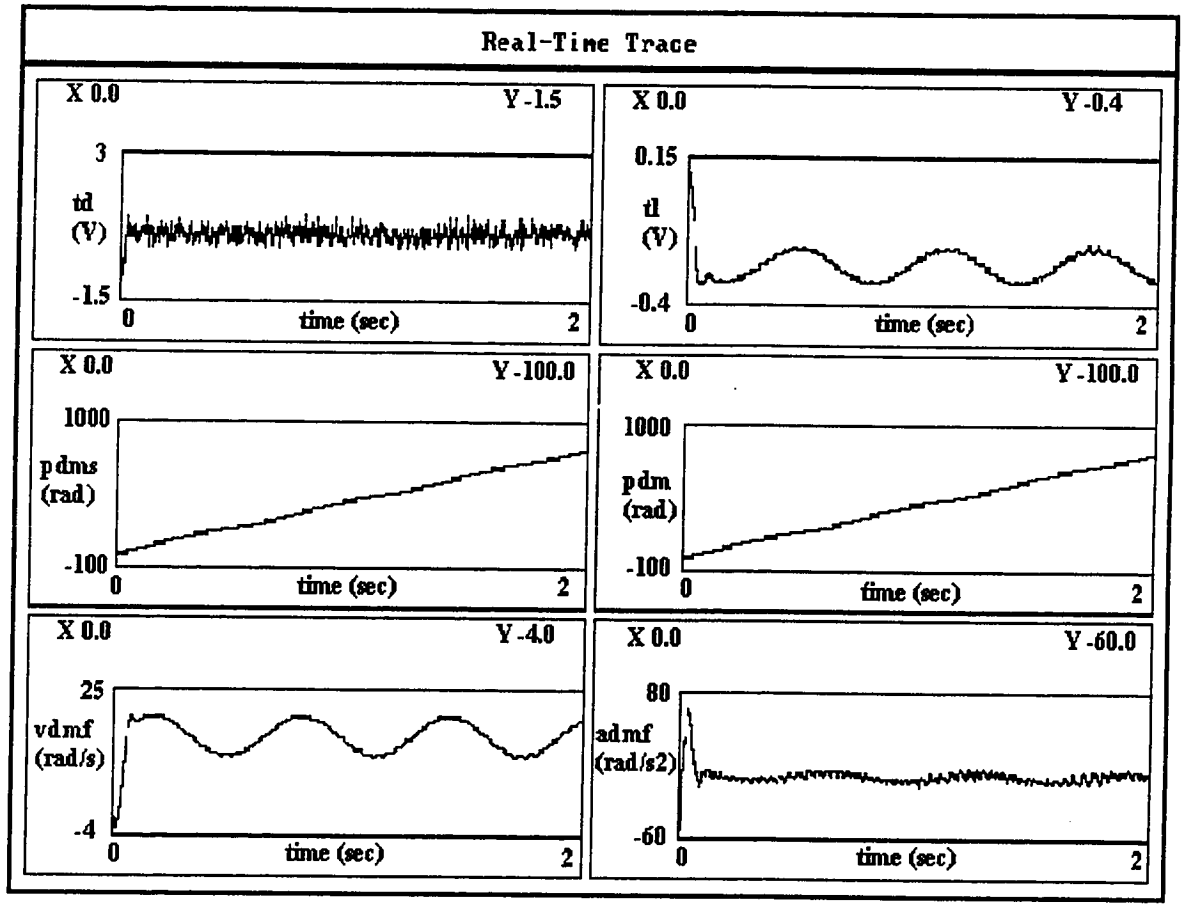


Figure 6.26 Physical simulation of the Parker linear load with combined input with 0.159 Hz frequency

The following two figures show the result for the experiments with physically simulated Parker DC motor subjected to sinusoidal inputs of 0.796 and 1.592 Hz frequency respectively. The behavior of the velocity characteristic shows similar results with the real load velocity characteristic. That brings us to the conclusion that physical

simulation of the loads is not influenced by type of the load, and it can be implemented for any low and high frequencies. The real load experimental results were shown in Figures 6.13 and 6.15 respectively.

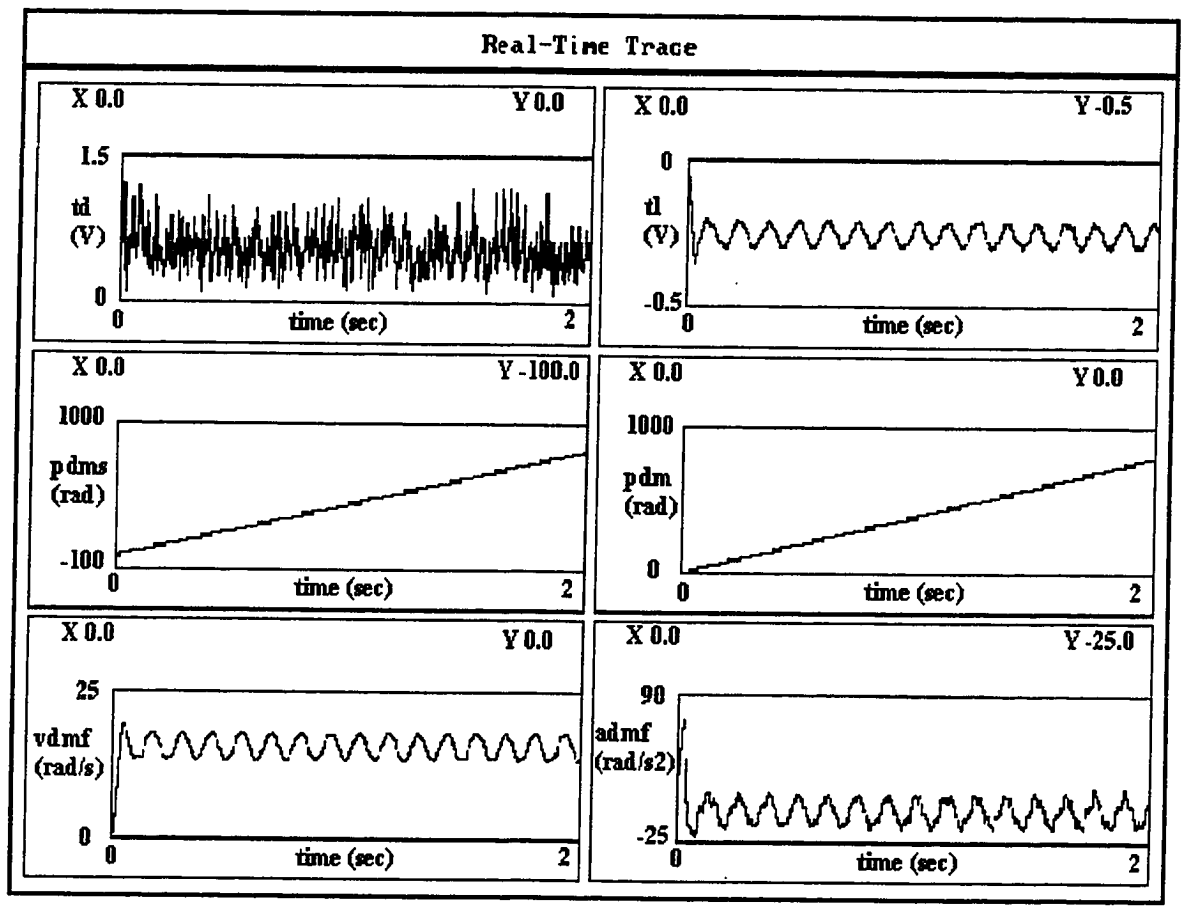


Figure 6.27 Physical simulation of the Parker linear load subjected to the combined input with 0.796 Hz frequency

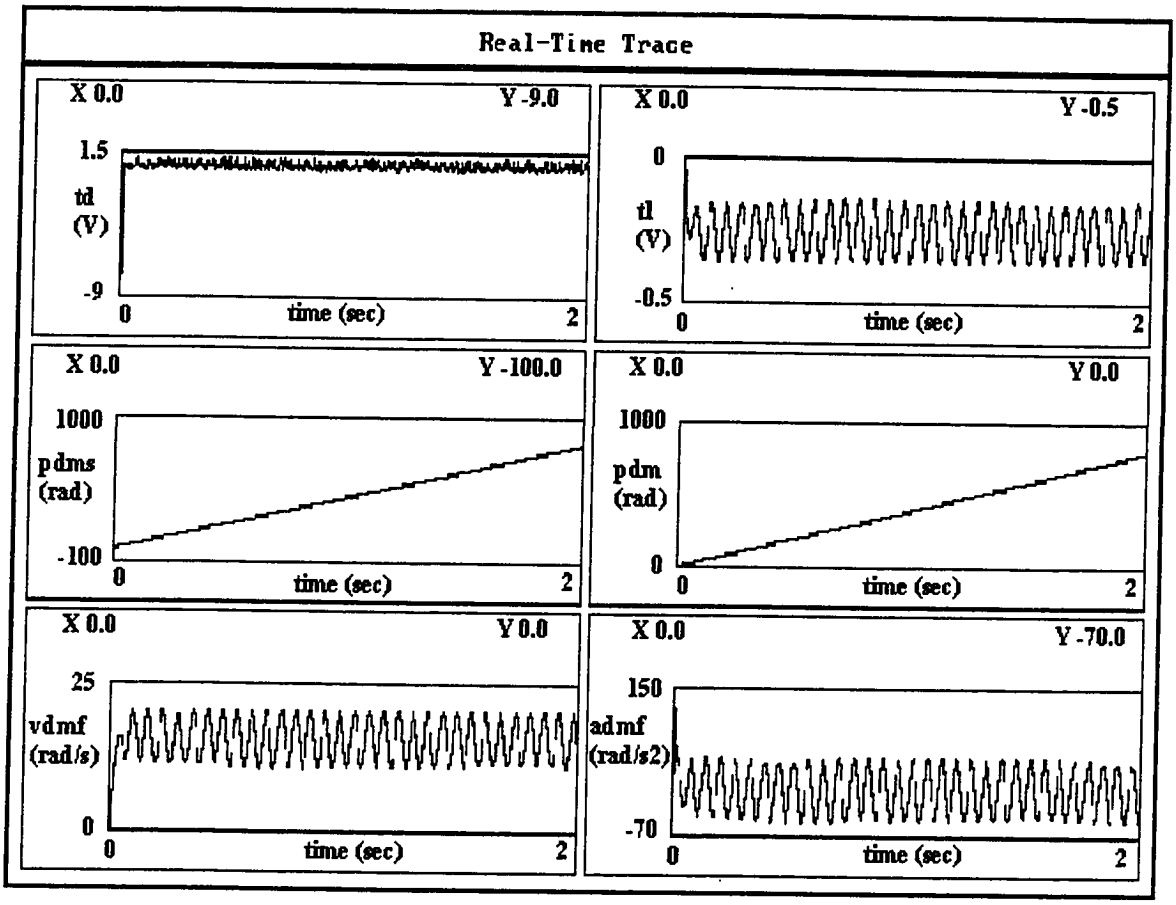


Figure 6.28 Physical simulation of the Parker linear load subjected to the combined input with 1.592 Hz frequency

6.5.2. Nonlinear load

6.5.2.1. Ramp input

The nonlinear load used in the following experiments was a nonlinear fan, the same used for the open loop experiments. The Pittman 9232 was used as physical simulator and the following set of data were used for those simulations the same as for the open loop experiments.

$J_{sim} = 0.00021375 \quad [kgm]$

$B_{sim} = 0.000035 \quad \left[\frac{Nm}{rad/s}\right]$

$$B_{op} = 0.00000383 \quad \left[\frac{Nm}{rad/s}\right]$$

$$F_{dry} = 0.000000156 \quad [Nm].$$

Again, the only difference here is the new concept used with current feedback implementation for the load simulator loop.

First experiment is, again, real load subjected to the ramp input. The results for the real fan load subjected to ramp input are presented in Figure 6.17.

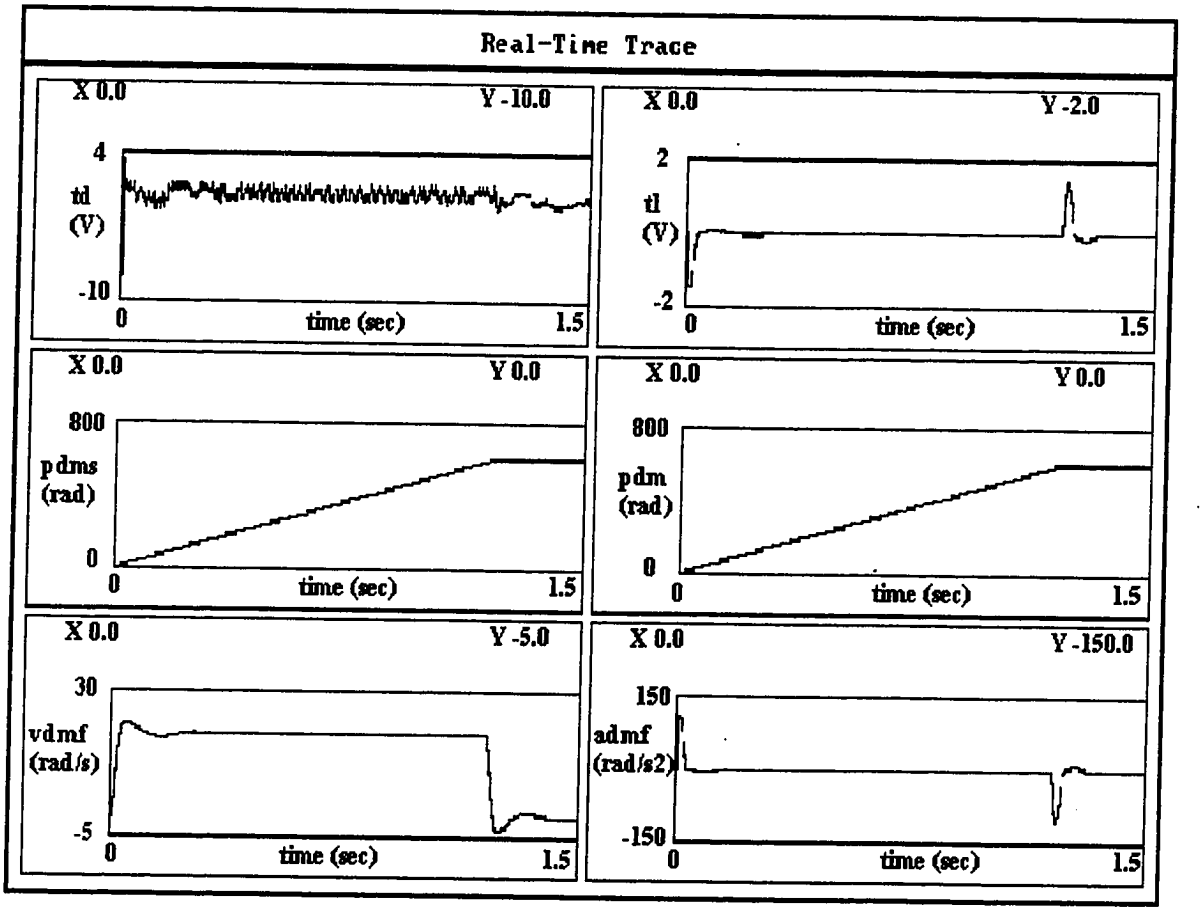


Figure 6.29 Physical simulation of the fan nonlinear load with ramp input

The physical simulation of the fan is shown in Figure 6.29. The velocity characteristic follows closely the original curve obtained with real fan. The difference makes smaller nonlinearity at the beginning of the cycle. The dynamic values are slightly offset from the real values, which causes inaccuracy. The acceleration is much smoother than in the original curve. It is interesting to notice that the shape of the signal formed to drive

physical simulator resembles the acceleration curve with the difference that it has the opposite sign and the value is different.

6.5.2.2. Combined step and sinusoidal input

Combined input experiments start with 0.159 Hz sinusoidal input. The real fan results were shown in Figure 6.19. The results for the physical simulation are shown in Figure 6.30. The velocity characteristics shows small peaks at the beginning, which was found in the original real load curve as well. The acceleration has the higher values than expected, but the shape is more accurate than the shape of the acceleration for the open loop experiment.

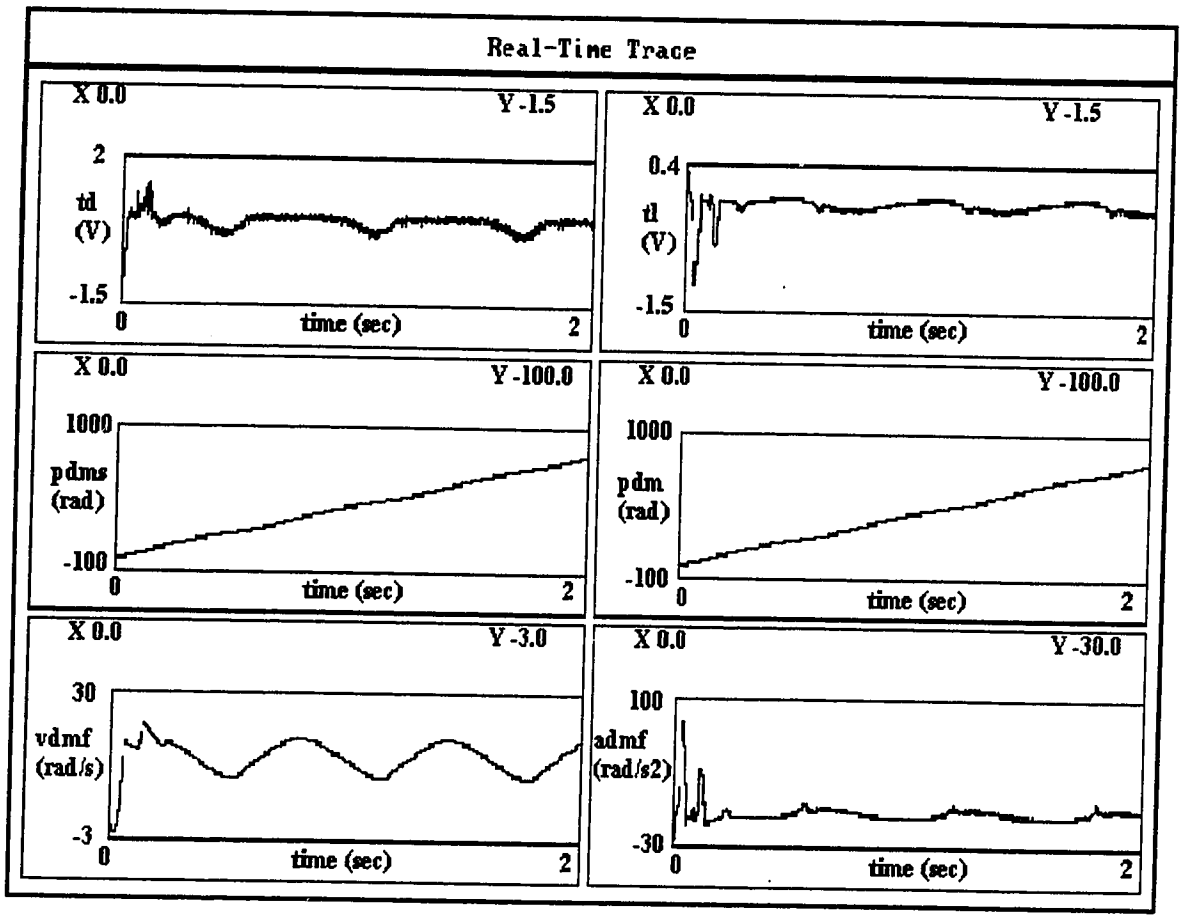


Figure 6.30 Physical simulation of the fan nonlinear load with combined input with 0.159 Hz frequency

Better results could be obtained if the parameters are identified more accurately and the load modeled in more details.

The following two figures show the results for the physically simulated nonlinear fan load when subjected to 0.796 and 1.592 Hz frequency combined input. The values of parameters used for these simulations were the same as for linear input. According to that and to the previous analysis for 0.159 Hz input the velocity characteristics are not perfect but they are close. Increasing of inertia and viscosity may improve the velocity, but also worsen the acceleration of the system.

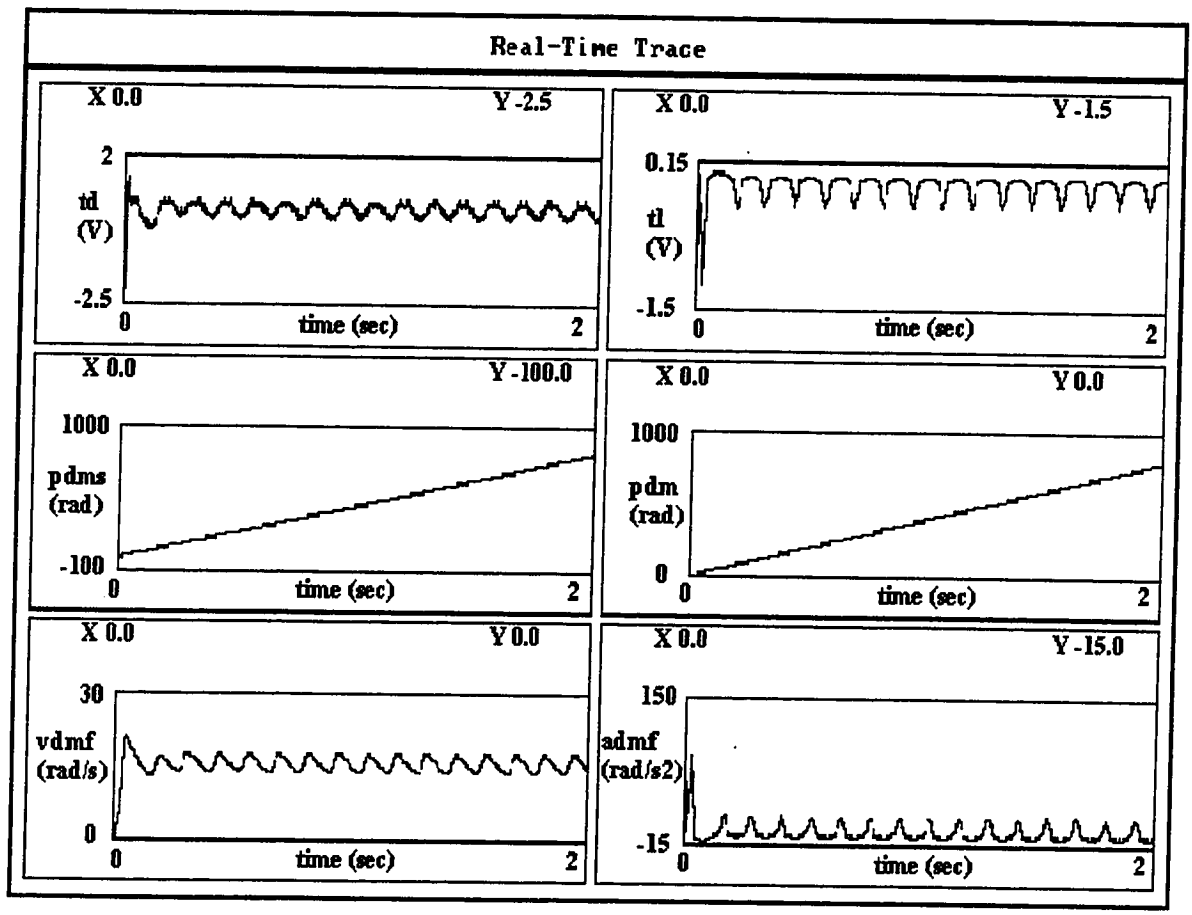


Figure 6.31 Physical simulation of the fan nonlinear load with combined input with 0.796 Hz frequency

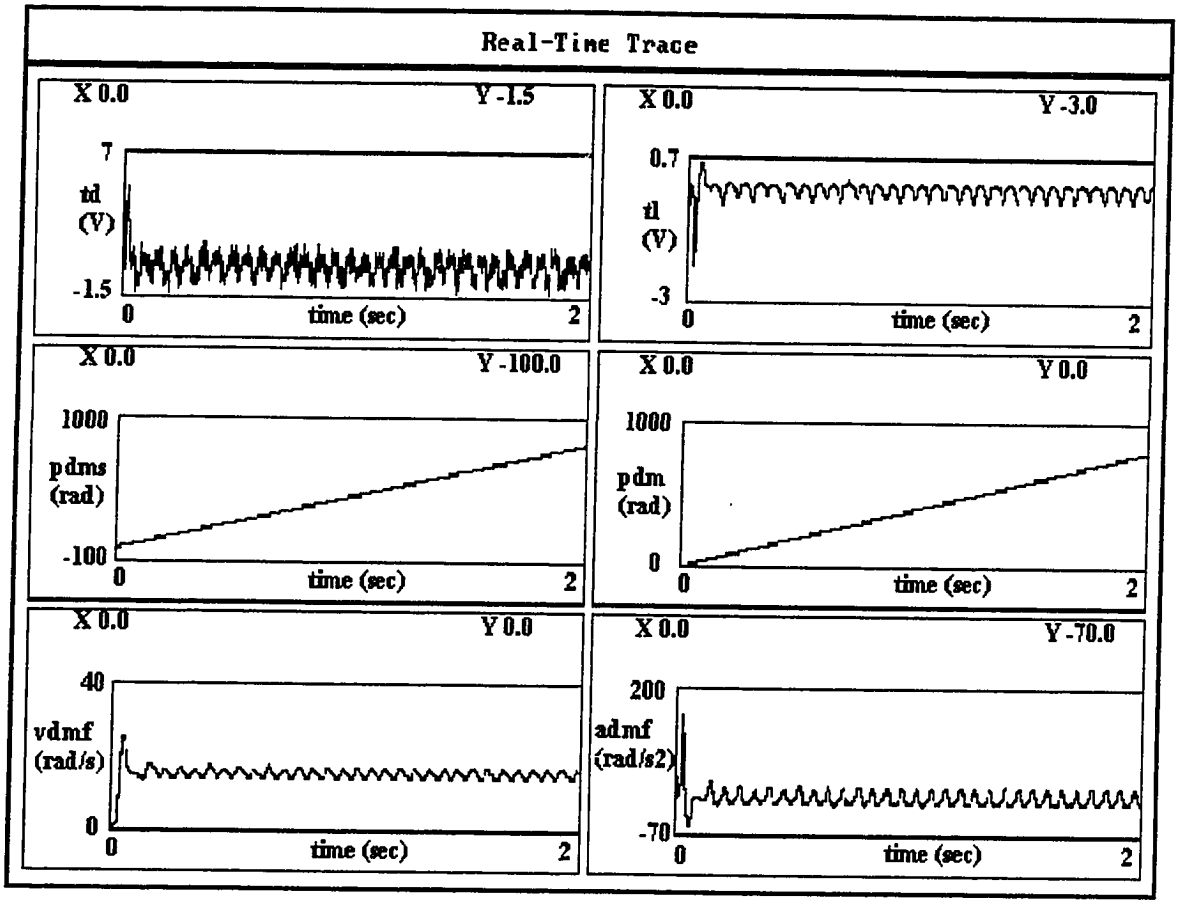


Figure 6.32 Physical simulation of the fan nonlinear load with combined input with 1.592 Hz frequency

7. POSSIBLE APPLICATIONS AND LIMITATIONS

7.1. Possible applications

The Hardware-In-The-Loop method proposed in this thesis can be used as a base for variety of applications.

One of the applications of the HIL method is in passive transmission. Transmission of the moment, velocity or storage of mechanical energy can be achieved using gear trains, shafts, flywheels etc. All these have one characteristic in common. They are passive mechanical elements, with effort-flow inputs and outputs. This connection characteristic allows them to be replaced by physical simulators.

Another application could be replacement of a stepper motor with a physical simulator made of a PM DC gearless motor. Another motor of different type can physically simulate the active load or in this case the active actuator.

In general, loads that are applicable for the method are either active or passive and, furthermore, they can be linear or nonlinear.

Knowing the types of loads the question is what would be the purpose of these applications and how would they be implemented. There are two main possibilities.

The first one is in replacing existing mechanical parts, and improving their functionality by varying the effect of various inertia and/or viscosity values using physical simulator. That way, the physical simulator would work as an instrument, which is correcting present characteristics of mechanical part. For example, a gear can be already a part of a gear train and the improvement of its inertia is needed, since it seems

to be malfunctioning. By replacing a gear with a physical simulator, or simply adding one in series with the gear, it is possible to rectify existing inertia and immediately see the effect of the improvement.

The second possibility is to use the HIL method in the case when the load is not known and is still not available for the system. A physical actuator can replace the future part and simulate it, performing the same way as if the real part was in the place. For example, a simulator attached to the engine the same way as the real flywheel can physically simulate a flywheel used in the car engine. The performance of the engine, in that case, can be evaluated within the real circumstances.

In the case of simulating active loads, physical simulator would have to perform more than for passive loads. For example, if a PM DC motor is used to simulate a stepper motor, besides simulating inertia, viscosity and dry friction, the simulator should behave as stepper motor trying to create step motion characteristic for stepper motors. That would require additional modeling and a development of a new current scheme for that specific task.

7.2 Limitations

We should clarify some limitations of the presented method. Physical simulators are designed with a motor as a main device. In this thesis, it is a PM DC motor, but it could be a direct drive motor, a stepper or an AC motor. Limitations that are the results of electrical motors are the bandwidth and the saturation torque. Every application requires a different motor, depending on the size of the load to be physically simulated and the power needed for the simulation.

Amplifiers are drivers for physical simulators and they have limited power and bandwidth.

Besides hardware limitations, software and operating system can contribute to result errors.

A real time operating system is required for precise results, since models developed in this thesis are based on precise sensor signaling. Use of some cheaper OS definitely reduces the quality of the results.

Modeling of loads is very important. Nonlinear loads and nonlinear areas for linear loads are difficult to model. One example is dry friction, whose amount and shape are usually not known.

8. CONCLUSIONS AND RECOMMENDATIONS

8.1. Conclusions

The following conclusions can be drawn from the research:

1.

The Hardware-In-The-Loop experimental setup used in this research, built as a generic setup, allowed the development of simple actuator-load models. The modularity of the setup opens the potential development of new models that can be implemented in the existing setup. Constant replacement of real loads and physical simulators was done without problems due to the same modularity of the design.

2.

Linear and nonlinear models developed during the research were simple and clear, based on the effort-flow concept. The matrix presentation emphasized effort-flow cuts and modularity of the system, which is base for the HIL implementation. Further modeling should be easy whether it is remodeling of the existing modules or adding a new ones.

3.

The software created for this research followed closely developed models. Any further software development in this research can be easy continued after new models are developed. The existing software is easily modifiable. The torque saturation was avoided before the signal was sent to the setup. More effort should be added into development of

digital filters, since it was observed that small changes in filters noticeably affect experimental results.

4.

The experiments conducted during this research clearly proved the models and the concepts on which the whole thesis was stated. The open loop experiments showed the strength of the models without feedback signal for the load simulating motor. Adding a current feedback control improved previous results. The conclusion is that a variety of loads can be physically simulated.

8.2. Recommendations

Further development of the experimental setup should include addition of a torque sensor. That way the HIL3 concept mentioned in chapter 3 can be tested, featuring full torque feedback control.

Transfer functions of amplifiers should be analyzed in more detail. With more knowledge about amplifiers, it should be easier to explain the difference between theoretical and practical results for experimental determination of transfer functions and some of HIL tests.

A better control table for amplifier on/off switches should be designed. If possible, amplifiers of the same type should be used for both motors.

Different motors should be considered as a new physical load simulator. The purpose would be to compare results for two physical simulators simulating the same real load.

New loads should be used for experiments. They should include active and passive, as well linear and nonlinear loads. The possible candidates are bearings, mixers that mix oils with different viscosities, stepper motors, torsion springs, car wheels, etc.

The next step in modeling would be the development of models for active loads such that different motors, and development of models for variety of nonlinear loads.

New tests should include the comparison of results obtained using the dSPACE and a microcontroller. In addition, it would be interesting to see the difference between results obtained on real-time and non-real-time operating systems.

9. REFERENCES

[1] Dan Necsulescu, Mechatronics, Prentice Hall, 2001

[2] Roman Baican, Dan Necsulescu, Applied Virtual Instrumentation, WIT Press, 2000

[3] Herbert Hanselmann, HIL Simulation Testing and its integration into CACSD Toolset, IEEE International Symposium on Computer-Aided Control System Design Dearborn MI- Proceedings, pp. 152-157, 1996

[4] Klaus Lamberg, Peter Waltermann, Using HIL Simulation to test Mechatronic components in Automotive Engineering, Translation of 'Einsatz der HIL-Simulation zum Test von Mechatronik-Komponenten in derFahrzeugtechnik', 2-nd Coongress on 'Mechatronik im Automobil', Munich 2000

[5] R. Isermann, J. Schaffmit, S. Sinsel, HIL Simulation for the design and testing of engine control systems, Technical University of Darmstadt, Transaction IFAC Algorithms and Architectures for Real-Time Control, Cancun, Mexico,1998

[6] dSpace, Vehicle Dynamics Simulator, Preliminary report, dSpace GmbH Paderborn, 2000

[7] Hans-Christian Reuss, Tobias G. Flamig-Vetter, Dresden University of Technology, Development of electronic control systems for automotive applications using HIL Simulation tools, Proceedings on IFAC Advances in Automotive Control, Loudonville, Ohio, 1998, pp. 241-246

[8] A. J. Beaumont, A. D. Noble, Automation of ECU Software Development: From Concept to Production level Code, report 1999-01-1174, Society of Automotive Engineers,1999

[9] Edward G. Huber Jr., HIL Simulation at Wright Laboratory's Dynamic Infrared Missile Evaluator (DIME) Facility, report, SPIE Vol. 3084, pp. 2-8, 1997

[10] Chet deCesaris, Paul Millner, Integrated three tiered approach to HIL testing, Pentagon, Washington, 1997

[11] S. Y. Yang, S. T. Park, K. S. Cho, An Efficient Simulation Platform for the Development of ABS using HIL System, University of Ulsan, Korea, 2nd International Conference on Recent Advances in Mechatronics, Istanbul, 1999

[12] Alan Ptak CRIM Canada, Khalil Foundy, CSA Canada, Real time SpaceCraft Simulation and HIL testing, Fourth IEEE Real-time Technology and Applications Symposium, proceedings pp. 230-236 , Denver Colorado, 1998

[13] P. Schafer, University of Stuttgart, HIL Simulation of multibody System models with transputers, preprints of IFAC workshop, Perugia, Italy, 1992, pp. 33-39

[14] A. Wright, P. G. Newbery, Electric Fuses, IEE Power Engineering Series 2, 1984

[15] Robert W. Smeaton, William H. Ubert, Switchgear and Control Handbook, 3rd edition, McGraw-Hill Book Company, 1998

[16] Michael A. Antony University of Michigan, Electric Power System Protection and Coordination, McGraw-Hill, 1995

[17] Nicholas Scott Hardmann, A Reconfigurable Hardware Testbed for Elastically-Coulpled Systems, Master Thesis, University of Washington, 1997

[18] Chad English, Mechanics and design of a variable stiffness actuator for use in impedance control, M.Eng Thesis, Carleton University, 1996

[19] Katsuhiko Ogata, Modern Control Engineering, 3rd edition, Prentice Hall, 1997

[20] Tak Kenjo, Electric Motors and their Controls, Oxford University Press, 1991

[21] Martin Clifford, Modern Electric/Electronic Motors, Prentice Hall, 1990

[22] B. C. Kuo, DC Motors and Control Systems, SRL Publishing Company, 1978

[23] Eugene A. Avalone, Theodore Baumeister III, Marks' Standard Handbook for Mechanical Engineers 10th edition, McGraw-Hill, 1986

[24] Robert H. Bishop, Learning with LabVIEW, Addison Wesley, 1998

[25] Larry K. Harbor, SIMSTAR Hardware-in-the-loop simulation of an Air-to-ground missile, Martin Marieta Corporation, 1989

[26] Ulrich Sailer and Ulf Essers, Heavy Vehicle Dynamics and Simulation in Braking, Steering, and Suspension Systems Real-Time Simulation of Trucks for Hardware-in-the-loop Applications, report 942297, University of Stuttgart 1994

[27] Flexishaft System Description and Modeling, Control Systems Laboratory, University of Washington, course notes- laboratory experiment, http://www.aa.washington.edu/controls/exper/fshaft/ , 2001

[28] W. Leonhard, Control of Electrical Drives, Springer Verlag, 1985

[29] T Sokira and W Jaffe, Brushless DC Motors Electronic Communication and Controls, Tab Books Inc.,Blue Ridge Summit, PA, 1990, P.124

[30] G. Escobar, R Ortega, H. Sira-Ramirez, J-P Vilain, I. Zein , An experimental Comparison of Several Nonlinear Controllers for Power Converters, IEEE, 1999

[31] H. Temeltas, M. Gokasan, O.S.Bogosyan, A Nonlinear Load Simulator for Robot Manipulators, the 27th Annual Conference of the IEEE Industrial Electronics Society, Denver, Colorado, USA, 2001

[32] Satoshi Ueno, Yohji Okada, Characteristics and Control of a Bidirectional Axial Gap Combined Motor-Bearing, IEEE/ASME Transactions on Mechatronics, Vol. 5, No. 3, September 2000, pp 310-318

[33] Bruce E. Tucker, Kenneth W. Zabel, A Technical Perspective on Developing Parallel Processing Applications for Image Processing for Hardware-in-the-loop Testing, Sparta Inc. Alabama, report, US Army Redstone Technical Test Center, 1998

[34] David Lucas, Pitu Mirchandani, Larry Head, Using Remote Simulation to Evaluate Real-Time Traffic Control Strategies, Transportation Research Record 00-1691, ATLAS Research Center, University of Arizona, 2000

[35] Li Liu, Andrew B.Wright,A Flexible Real-time Motor Control System for Robot Joints, The 7th. Mechatronics Forum International Conference, Atlanta Georgia, Sept 2000

[36] Kalev Sepp, Friction Modeling in Linear track Cart Pendulum System, University of Washington, Design of Control Systems Course White Paper, Spring 2001

[37] Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, AT&T Bell Laboratories, New Jersey, Second Edition 1988

[38] Rahim Jassemi-Zargani, Impedance Control of a Dual-Arm Robot, PhD Thesis, Department of Mechanical Engineering University of Ottawa, Ontario, 1998

[39] DSP-CITpro, Hardware Manual, v.1., dSPACE digital Signal Processing and

Control Engineering GmbH, Germany

[40] DSP-CITpro, Trace Manual, v.1., dSPACE digital Signal Processing and Control Engineering GmbH, Germany

[41] W. Snyder, Industrial Robots: Interfacing and Control, Prentice Hall 1985, p. 73

[42] De Silva, Control Sensors and Actuators, Prentice Hall, 1989

[43] G.C. Goldsmith, M.A. Amick, L.E. Jones, Integration, calibration, and testing of resistor array dynamic infrared scene projector on the outer axis of a five axis flight motion simulator for real time HIL simulations, Proceedings of SPIE – The International society for Optical Engineering;V,2741, 1996, p.119

[44] B. Sieglinger, D.S. Flynn, Hardware-in-the-loop simulation using direct signal injection, Proceedings of SPIE – The International Society for Optical Engineering;V,2741, 1996, p.219

[45] Jeffrey S. Sanders, Ground Vehicle thermal signature modeling and validation for Hardware-in-the-loop imaging infrared sensor testing, Proceedings of SPIE – The International Society for Optical Engineering ;V,2741, 1996, p.257

[46] F. Oueslati, S. Rakheja, I. Stiharu, Investigation of a Limited-State Active Suspension for Articulated Heavy Vehicles, CONCAVE Research Centre Concordia University, Montreal, 1995

[47] EG&G Torque Systems, Technical Notes for PA228 and PA245, EG&G Torque Systems, 36 Arlington Street, Watertown, Massachusetts 02172, 1993

[48] Online Catalogue, Stock Drive Products, www.sdp-si.com , New York, 2002

[49] APEX, Application notes, www.teamapex.com, 2002

[50] Ramakant A. Gayakwad, OP-AMPS and Linear Integrated Circuits, Prentice Hall, 2-nd edition, 1988, p.145-165

[51] Megatorque Motor System, User's Manual, Motion and Control Technology NSK NIPPON SEIKO K.K., Japan, 1989

[52] Parker Hannifin Corporation, System 7 User Guide, Digiplan Division, USA, 1991

[53] The Mathworks Inc., XPC Target 1.0 - New product overview, www.mathworks.com , 1999

[54] Pittman- a Penn Engineering Company, Pittman Catalogue, Harleysville, PA, www.pittmannet.com , 2000

[55] Hal Gurgenci, Course notes - Mechatronic System Design Course, University of Queensland, Australia, 2001, www.mech.uq.edu.au/courses/mech3760/chap37/s1.htm

[56] LabVIEW, National Instruments Inc, Austin Texas, www.ni.com

[57] Pa228-001 Amplifier, Cleveland Machine Controls

[58] Matlab/Simulink, The Mathworks Inc., www.mathworks.com

[59] Stock Drive Products/ Sterling Instrument, Drive Components Design - Pulleys, New Hyde Park NY, www.sdp-si.com

APPENDIX A

(Experimental determination of a transfer function)

A.1. Introduction

The transfer function of a system represents a mathematical relation between Laplace transforms of its input and output signals. For every chosen pair of inputs and outputs, one function shows the dependence of the output on the input and, in most cases, on some other state variables of the same system.

Deriving a transfer function of the system means defining ratio of the output to the input of the system. Since transfer functions use complex mathematical expression of time variables, by applying a Laplace transform it is obtained as a rational function of complex variable s. For such derivation, the system has to be linear, time invariant and the initial conditions have to be zero.

Depending on how many inputs and outputs are of interest, the same system can have many different transfer functions. If input and output vectors of the system are defined, mathematical model of the system can be presented in state-space form, which contains all systems transfer functions of interest.

Transfer functions are usually derived for families of similar systems. If one wants to derive transfer the function for a DC motor, considering all variables on both, mechanical and electrical sides, one can write differential equations describing system dynamics. After choosing the input and output of interest, performing Laplace transform and rearranging equations into rational form, a transfer function can be obtained. We can say that such transfer function is valid for all DC motors since it was derived without using specific numerical values for parameters for particular motor. Instead, algebraic notations were used, which define a general transfer function for all cases. From such an abstract transfer function, Bode diagram cannot be constructed before assigning numerical values. We can only presume the shape of the magnitude and phase shift curves. Knowing only the shape of those curves, but not the exact position, we can talk only about minimum phase and the type of the system.

To distinguish between two similar DC motors we need to know their characteristics or simply numerical values of all variables. Substituting those values into a general transfer function, we can obtain a transfer function for particular motor. Such transfer

function is easy presentable in the Bode diagram form. To do so one needs to rewrite the particular transfer function so it contains basic factors such are gain, integrative and derivative factors, first order factors and second order factors. Each of those basic factors is represented in the Bode diagram as simple straight lines or combination of two lines. The final magnitude curve is obtained as sum of basic curves since the Bode diagram uses logarithmic scale for its x-axis. The Bode diagram represents frequency response of the system, or sinusoidal transfer function of the system.

If the values of variables are not known, one cannot construct particular transfer function. On the other hand, it is possible to reconstruct transfer function of the system if the Bode diagram is known. Since we know that the Bode diagram represents sinusoidal transfer function of the system, we can construct the Bode diagram for any system. Such system has to be subjected to sinusoidal input of various frequencies and by plotting, resulting output data into the Bode diagram, magnitude and phase shift curves can be constructed. Then going backwards from what we learned about constructing the Bode diagram it is possible to reconstruct the transfer function for the system with unknown characteristics.

A.2. A DC motor with known characteristics

To prove that the method for experimental determination of transfer functions for a system with unknown characteristics works, we can show how it works for the system with known characteristics. Firstly, we can derive the transfer function from manufacturer data and secondly, transfer function of the same system will be obtained experimentally. Finally, simple comparison of those two transfer functions can prove the method.

A.2.1. Mathematical derivation of a transfer function for Pittman DC motor

Let us derive the transfer function for a DC motor. First, we will obtain the transfer function for the family of DC motors and then apply manufacturer's data to it to get

particular transfer function. The motor that we will use is Pittman 9232, which is also the drive motor for experimental setup. Manufacturer's data are given in Appendix B.

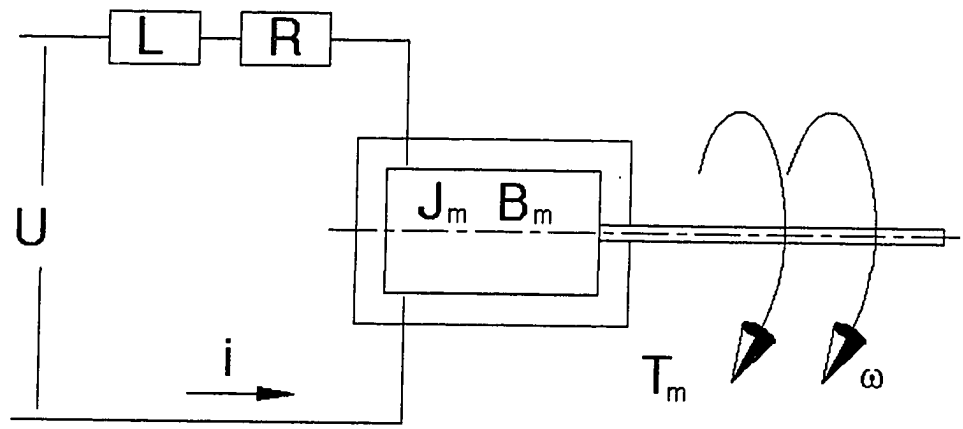


Figure A1. The diagram of PM DC motor

We can write equations:

- for the electrical side

$$U = L\frac{di}{dt} + R * i + K * \omega \tag{A.1}$$

- for the mechanical side

$$K * i = J_m \frac{d\omega}{dt} + B_m * \omega + T \tag{A.2}$$

We can presume that system is linear and time invariant so we can apply Laplace transform to it.

$$K * I(s) = J_m * s * \omega(s) + B_m * \omega(s) + T(s)$$
$$U(s) = L * s * I(s) + R * I(s) + K * \omega(s)$$

or by defining $\omega(s) = s * \theta(s)$ we can write

$$K * I(s) = J_m * s^2 * \theta(s) + B_m * s * \theta(s) + T(s)$$
$$U(s) = L * s * I(s) + R * I(s) + K * s * \theta(s)$$

If we choose $\theta(s)$ as output and U(s) as input, and eliminate I(s) and T(s) since there is no external load on motor, we can write

$$\frac{\theta(s)}{U(s)} = \frac{K}{\left[(L * s + R)(J_m * s + B_m) + K^2\right] * s} \tag{A.3}$$

The equation A.3 represents the basic transfer function for any DC motor. A PM DC motor usually has very small inductivity so L can be neglected. That leads us to the simpler, first order version of the transfer function.

$$\frac{\theta(s)}{U(s)} = \frac{K}{\left[R * (J_m * s + B_m) + K^2\right] * s} \tag{A.4}$$

Rearranging the equation A.4 into a form more suitable for frequency response analysis, we can obtain equation A.5:

$$\frac{\theta(s)}{U(s)} = \frac{\dfrac{K}{R * B_m + K^2}}{\left[\dfrac{s}{\dfrac{R * B_m + K^2}{R * J_m}} + 1\right] * s} \tag{A.5}$$

To create particular transfer function for the Pittman 9232 we should apply data to the equation A.5. After applying manufacturer's data, the following is transfer function of the Pittman 9232:

$$\frac{\theta(s)}{U(s)} = \frac{63.185}{\left[\dfrac{s}{66.97} + 1\right] * s} \tag{A.6}$$

Equation A.6 represents the transfer function of the Pittman 9232 according to the manufacturer's data. It is possible to obtain the natural frequency of the system $\omega_n = 66.97 \left[\frac{rad}{s}\right]$. In addition, total gain in the system is $G = 63.185$.

A.2.2. Experimental determination of a transfer function for Pittman DC motor without load

To determine transfer function of a system with unknown data it is necessary to obtain frequency response of such a system by means of experimental analysis. Amplitude ratio and phase shift have to be measured at a sufficient number of frequencies within the frequency range of interest [19]. We will plot obtained data on the Bode diagram.

Since we are interested only in the reconstruction of transfer function, phase shift measurement is not of interest. We will record magnitude ratio over the certain range of frequencies.

System description

The system was designed as an open loop system and consists of several components including

a) Pittman 9232 DC motor;

b) dSPACE operating system used to generate a sinusoidal signal

A special C program was created for that purpose. The input was chosen $x = X \sin \omega * t$ where X represents input amplitude (magnitude), ω is angular velocity of the motor shaft, where $\omega = \frac{2\pi}{T} = 2\pi f$, and T is the time for which shaft turns 2π. This way we can calculate different frequencies by means of changing T. For example, if we want a motor shaft to rotate at the speed $\omega = 1 \left[\frac{rad}{s}\right]$, then $T = \frac{2\pi}{\omega} = \frac{2\pi}{1} = 6.28 \ [s]$. Since a frequency f is f=1/T we can calculate the frequency f=0.1592 for $\omega = 1 \left[\frac{rad}{s}\right]$. dSPACE sampling period is set as 1000 samples per second. So for input

signal $x = X \sin\frac{2\pi}{T} * \Delta t$ the time step for $\omega = 1 \left[\frac{rad}{s}\right]$ can be calculated as $\Delta t = 0.001 \ [s]$, since it will take 6280 samples to reach one full cycle. Similar calculation applies for all chosen frequencies. To cover most of the frequency range and not to go over $\omega_{max} = 734.6 \left[\frac{rad}{s}\right]$ the range was chosen: $\omega = 0.1 - 100 \left[\frac{rad}{s}\right]$.

The Pittman 9232 is equipped with an optical incremental encoder, which was used to record position change during experiments. Data obtained were used to create amplitude of the output signal.

c) Power amplifier PA228 used to create power sufficient to move the motor.

The transfer function for the amplifier is $K = 2 \left[\frac{A}{V}\right]$. Two 12V batteries supply the amplifier.

Experimental analysis

Experimental data obtained are defined as the ratio of output position change signal Y and input voltage signal X.

The following are data obtained form two measurements

ω	Y/X (1)	Y/X (2)	Y/X (avg.)	20logY/X
0.1	953	902	927.5	59.34
0.2	527.3	436	481.7	53.66
0.4	221.3	216.2	218.8	46.8
0.6	146.6	146.2	146.4	43.31
1	87.5	86	86.8	38.77
2	39.5	39	39.3	31.89
4	14.5	15	14.8	23.4
6	8.3	8.5	8.4	18.48
10	3.4	3.75	3.58	11.08
20	0.93	0.85	0.89	-1.01
40	0.23	0.225	0.228	-12.84
60	0.082	0.08	0.081	-21.83
100	0.027	0.025	0.026	-31.7

Table A1 – Experimental data obtained for Pittman9232 DC motor (frequency response)

Experimental data plotted on the Bode diagram and they are shown in Figure A.2.

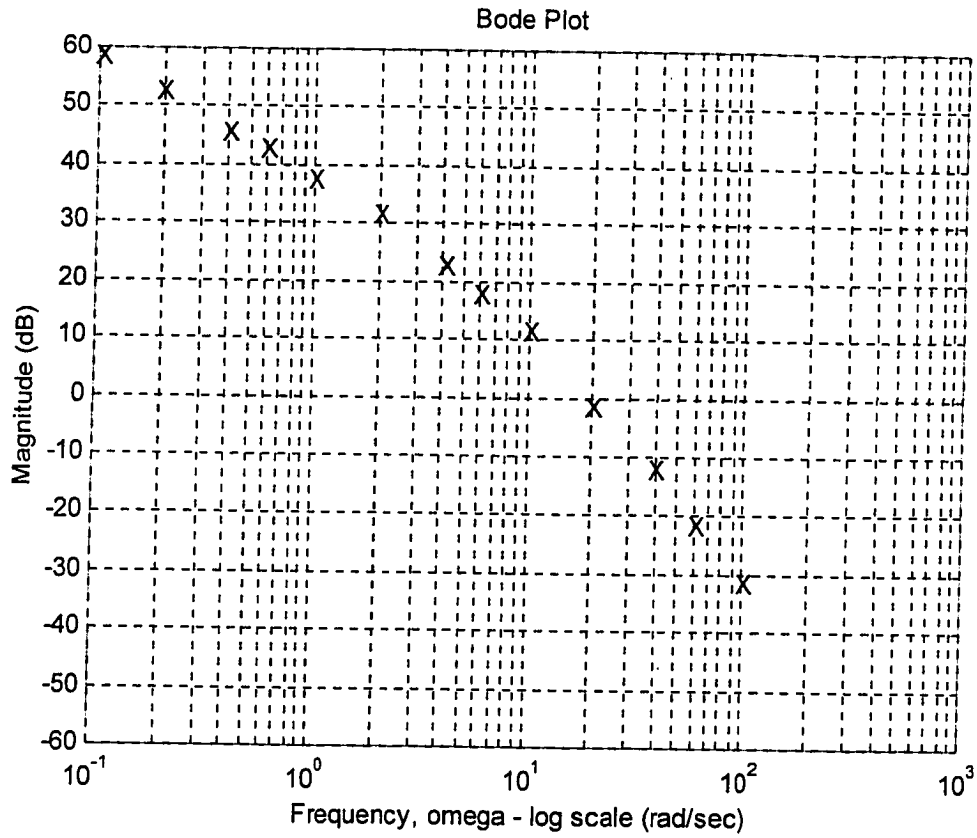


Figure A.2 Experimentally obtained frequency response data for Pittman 9232

Using the following transfer function A.5, for the family of DC motors

$$\frac{\theta(s)}{U(s)} = \frac{\dfrac{K_{amp} * K}{R * B_m + K^2}}{\left[\dfrac{s}{\dfrac{R * B_m + K^2}{R * J_m}} + 1\right] * s},$$

we can reconstruct transfer function of the particular DC motor. Clearly, four separate parts of the transfer function could be analyzed:

$$\frac{\theta(s)}{U(s)} = K_{amp} * \frac{1}{j\omega} * \frac{\dfrac{K}{R * B_m + K^2}}{\left[\dfrac{\dfrac{j\omega}{\dfrac{R * B_m + K^2}{R * J_m}} + 1}{}\right]} \tag{A.7}$$

The first one is gain of the amplifier $K_{amp} = 2 \left[\frac{A}{V}\right]$, the second is gain produced by motor, the third is integral factor $\frac{1}{j\omega}$, and the fourth one is first-order factor. These factors can be presented in the Bode diagram as well as their sum.

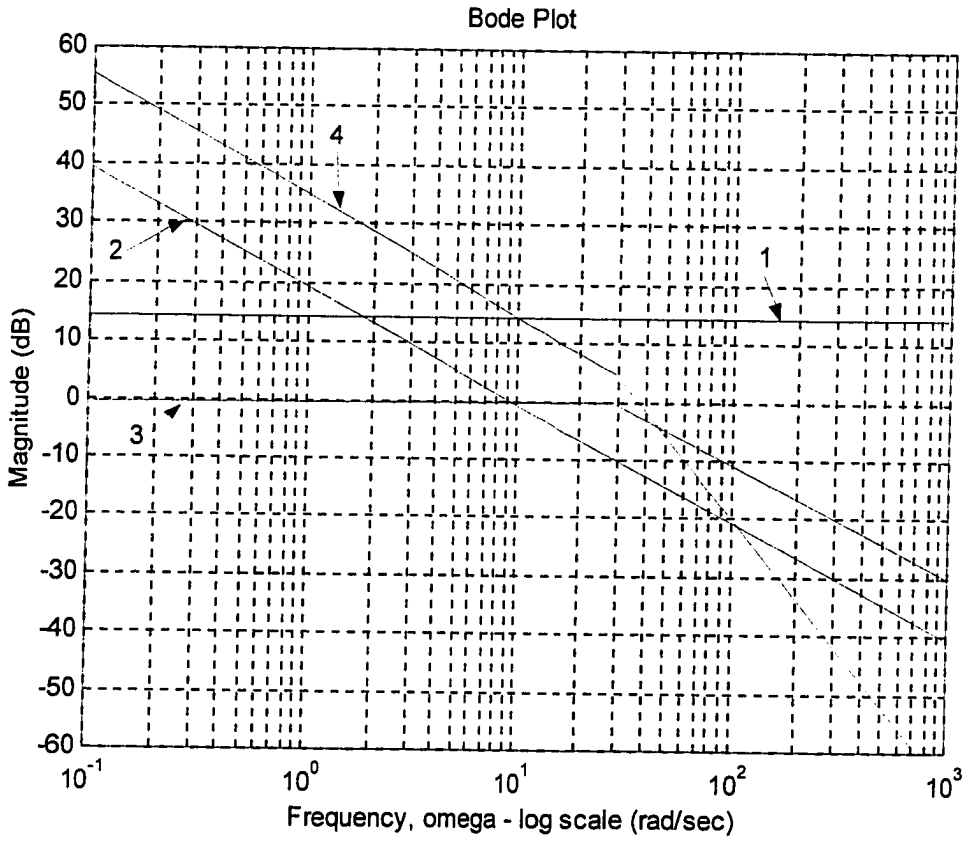


Figure A.3 Bode diagram – asymptotes and their sum

Figure A.3 shows separate factors drawn in the Bode diagram and their sum. Line 1 represents total gain of the system, line 2 integral factor, line 3 first-order factor and line 4 sum of these lines or diagram for the system.

We can draw asymptotes on the curve obtained by experimental data. From the asymptotic diagram obtained is possible to read some characteristic data of the system.

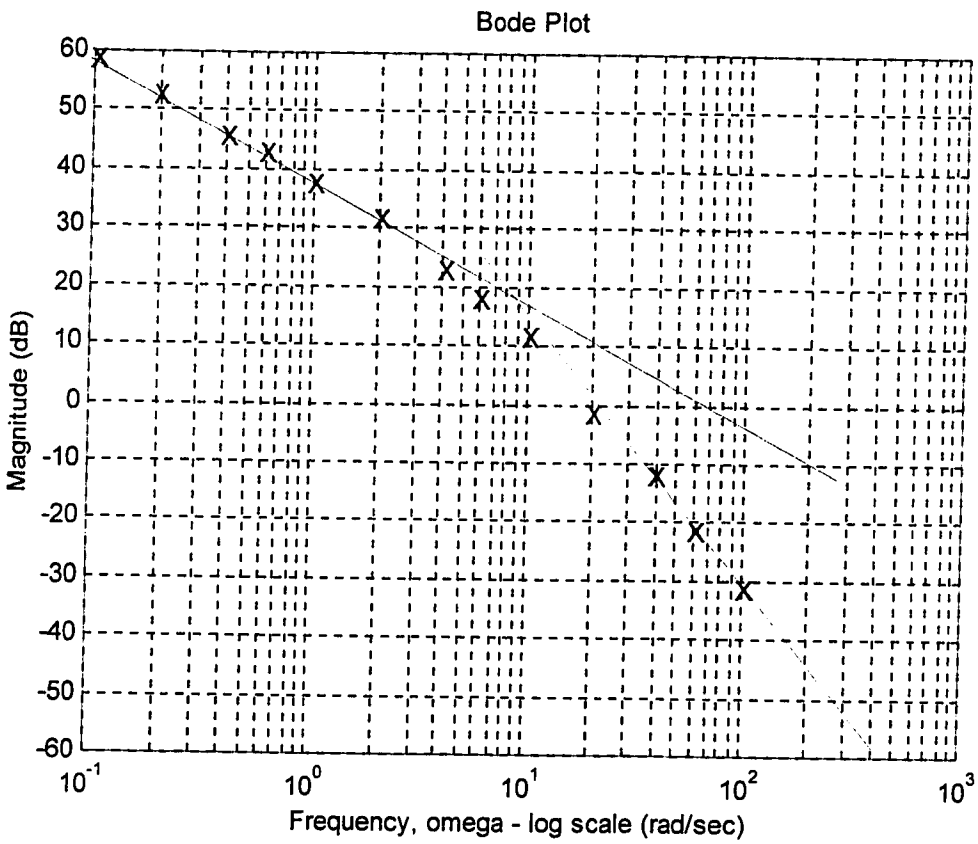


Figure A.4 Asymptotes applied to the experimentally obtained data

The total gain of the system on the Bode diagram is at the point where integral asymptote crosses line dB=0. In our case that is K= 65.

The corner frequency, when the asymptotes are changing from -20 dB/decade to -40 dB/decade, represents undamped natural frequency of the system and it is $\omega_n = 8.5 \left[\frac{rad}{s}\right]$.

Amplifier transfer function is $K_{amp} = 2 \left[\frac{A}{V}\right]$.

Finally, considering all these results we can reconstruct the transfer function of the Pittman 9232 DC motor as

$$\frac{\theta(s)}{U(s)} = \frac{65}{\left[\frac{j\omega}{8.5}+1\right]} * \frac{1}{j\omega} \quad \text{or}$$

$$\frac{\theta(s)}{U(s)} = \frac{65}{\left[\frac{s}{8.5}+1\right] * s} \tag{A.8}$$

Comparing experimentally obtained transfer function and the one derived from manufacturer's data we can see that they differ significantly as frequency increases. The experimental curve has the same shape but much smaller bandwidth. The reason for that is in the experimental setup on one side, which contains nonlinearities, and in simplified model of the transfer function for family of motors, which is neglecting some variables, on the other side. Figure A.5 shows comparison of the Bode diagrams for manufacturer's data and experimentally obtained data for the Pittman 9232. The full line represents data obtained from manufacturer, while crosses are data obtained experimentally using manufacturers model.

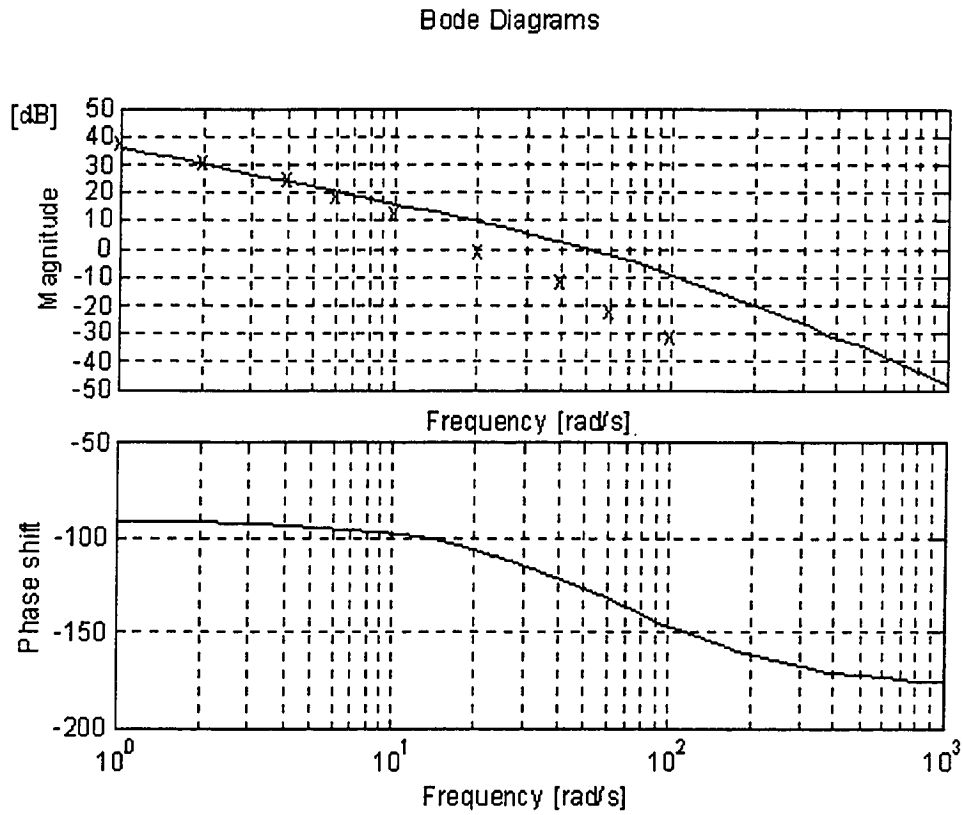


Figure A.5 Comparison of experimental data and frequency response based on manufacturer's model

A.2.3 Experimental method of determination of a moment of inertia and a viscous damping for an unknown load using data obtained from frequency response of the system

After obtaining the frequency response characteristic of the system, it is important to extract information useful for further analysis. After drawing asymptotes on the Bode diagram, we can graphically obtain first set of data for a system. The natural frequency of the system ω_n can be obtained if we find the crossing point of two asymptotes (first one has a fall of -20 dB/decade and second one has fall of -40 dB/decade for type one systems). We can read the natural frequency on the x-axis for the crossing point. The

second important parameter is gain K. If we extend first asymptote until it reaches the line of 0 dB that is the point to read the gain of the system. Last parameter interesting for analysis is damping ξ. Since it depends on the shape of the curve around the point where two asymptotes meet it is not easy to find. Comparison with some standard diagrams may be useful.

Two basic models will be considered for the analysis. The first one is a DC motor with lumped load attached to it and the second one is a DC motor with load connected to it by a flexible shaft. For both models, it is necessary to obtain a set of formulas, which can be used to obtain rotational inertia J_l and viscous damping B_l for the load, connected to the DC motor in either of two mentioned ways. In the first step, Bode diagram derived from frequency response of the system will be used to obtain characteristics of the system such as natural frequency, damping and gain. This data will be used to calculate J_l and B_l for an unknown load using appropriate formulas.

Derivation of formulas for lumped model

We can start the derivation by using simple model of a DC motor actuator. For the lumped model both the rotational inertia and the viscous damping are simply added together to obtain system's rotational inertia and viscous damping.

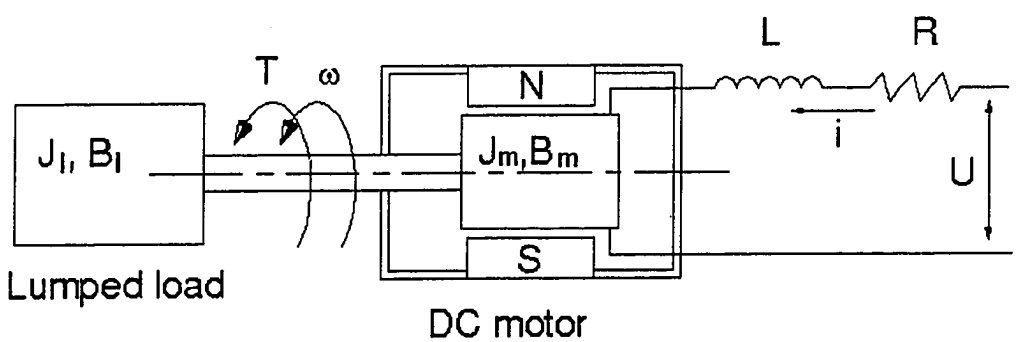


Figure A.6 DC motor with lumped inertial load

The Newton's second law can be applied on the free body diagram, for the mechanical and the electrical parts of the model. Considering two separate inertias and damping (actuator and load), the following equations can be derived:

$$K * i = (J_m + J_l) * \frac{d\omega}{dt} + (B_m + B_l) * \omega \tag{A.9}$$

- on the mechanical side

$$U = L * \frac{di}{dt} + R * i + K * \omega \tag{A.10}$$

- on the electrical side

In this case, motor is not geared, which would change equivalent inertias and equivalent damping.

If we rearrange equation A.9 we can get:

$$i = \frac{(J_m + J_l)}{K} * \frac{d\omega}{dt} + \frac{(B_m + B_l)}{K} * \omega,$$

or by differentiating it

$$\frac{di}{dt} = \frac{(J_m + J_l)}{K} * \frac{d^2\omega}{dt^2} + \frac{(B_m + B_l)}{K} * \frac{d\omega}{dt}.$$

These two formulas could be applied to equation A.10.

$$U = L\left[\frac{(J_m + J_l)}{K} * \frac{d^2\omega}{dt^2} + \frac{(B_m + B_l)}{K} * \frac{d\omega}{dt}\right] + R\left[\frac{(J_m + J_l)}{K} * \frac{d\omega}{dt} + \frac{(B_m + B_l)}{K} * \omega\right] + K * \omega$$

Inductivity L for DC brushed motors has in reality very small value so it can be neglected. After rearranging the factors and applying Laplace transform we can obtain:

$$U(s) = \frac{(J_m + J_l) * R}{K} * s^2 * \theta(s) + \frac{(B_m + B_l) * R + K^2}{K} * s * \theta(s) \tag{A.11}$$

The last equation allows us to create the transfer function for the DC motor with lumped load, with the voltage as an input and the position of the shaft as an output.

$$\frac{\theta(s)}{U(s)} = \frac{K}{[(J_m + J_l) * R] * s^2 + [(B_m + B_l) * R + K^2] * s} \tag{A.12}$$

Equation A.12 can be rearranged into frequency response suitable form:

$$\frac{\theta(s)}{U(s)} = \frac{\dfrac{K}{R*(B_m + B_l) + K^2}}{\left[\dfrac{\dfrac{s}{R*(B_m + B_l) + K^2}}{\dfrac{R*(J_m + J_l)}{R*(B_m + B_l) + K^2}} + 1\right] * s} \tag{A.13}$$

Equation A.13 gives the following information:

$$K_{eq} = \frac{K}{R*(B_m + B_l) + K^2} \tag{A.14}$$

where K_{eq} represents total gain of the system,

$$\omega_n = \frac{R*(B_m + B_l) + K^2}{R*(J_m + J_l)} \tag{A.15}$$

where ω_n is natural frequency of the system.

ω_n and K_{eq} are not known but could be obtained from Bode diagram for each particular DC motor. Once we know their values it is possible to solve these two equations for J_l and B_l. By rearranging A.14 and A.15 we can obtain the following set of equations:

$$J_l = \frac{K - K_{eq} * \omega_n * R * J_m}{K_{eq} * \omega_n * R} \tag{A.16}$$

$$B_l = \frac{K - K_{eq} * (R * B_m + K^2)}{K_{eq} * R} \tag{A.17}$$

Equations A.16 and A.17 can be used to obtain rotational inertia J_l and viscous friction coefficient B_l of the unknown passive load that is a part of the rotational system and modeled as a lumped load with an actuator. Before calculation, we need to read the Bode diagram and have ready actuator data.

Derivation of formulas for model with load attached to the actuator by flexible shaft

For the following analysis, flexibility of the shaft are assumed not negligible, so inertias and dumping of the system are not simple linearly added variables. Here we have to consider the flexibility of the shaft K_s, which is causing two different velocities in the system. The first one is the velocity of the actuator and the second one is the load velocity.

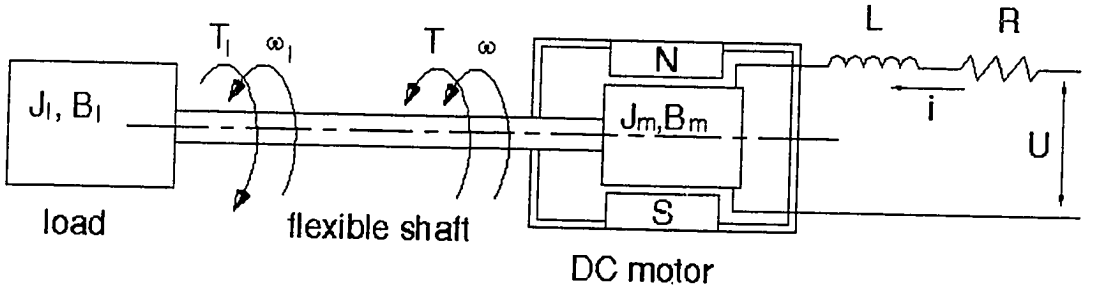


Figure A.7 DC motor with inertial load attached by a flexible shaft

While lumped model was simple enough to be modeled as a first order system, this system will be modeled as a second order system even after simplifications. For the second order systems, the transfer function looks slightly different but the approach for solving the problem of unknown inertia and damping stays the same.

We can start derivation with previously introduced equations for the actuator A.1 and A.2, with the Laplace transform already applied.

$$U(s) = L * s * i(s) + R * i(s) + K * \omega(s) \tag{A.18}$$

$$K * i = J_m * s * \omega(s) + B_m * \omega(s) + T(s) \tag{A.19}$$

For the shaft we can write

$$T(s) = \frac{K_s}{s}(\omega(s) - \omega_l(s)) \tag{A.20}$$

where K_s represents the shaft stiffness and ω is the angular velocity of the motor and ω_l is the velocity of the load. The torque produced by the motor and resisting torque of the load is equal in value but opposite in the direction.

$$T_l(s) = -T(s) \tag{A.21}$$

The torque created by the load is expressed by:

$$T_l(s) = (J_l * s + B_l) * \omega(s) \tag{A.22}$$

Combining equations A.20 and A.21 we can obtain the relationship between two velocities in the system.

$$\omega_l(s) = \frac{K_s}{J_l(s) * s^2 + B_l * s + K_s} * \omega(s) \tag{A.23}$$

Using formulas A.18, A.19, A.22 and A.23, and neglecting a small value of inductivity L, we can obtain the transfer function for the system with a load attached by a flexible shaft.

$$\frac{\theta(s)}{U(s)} = \frac{K * (J_l * s^2 + B_l * s + K_s)}{s * \left[\begin{array}{l} (R * J * B_l + R * J_l * B + J_l * K^2) * s^2 + (R * J * K_s + R * B * B_l + \\ R * J_l * K_s + B_l * K^2) * s + (R * B * K_s + R * B_l * K_s + K^2 * K_s) \end{array} \right]} \tag{A.24}$$

The last expression can be rearranged into the form suitable for the frequency analysis.

$$\frac{\theta(s)}{U(s)} = \frac{\dfrac{K * (J_l * s^2 + B_l * s + K_s)}{R * B * K_s + R * B_l * K_s + K^2 * K_s}}{\left[\dfrac{s^2}{\dfrac{R * B * K_s + R * B_l * K_s + K^2 * K_s}{R * J * B_l + R * J_l * B + J_l * K^2}} + \dfrac{s}{\dfrac{R * B * K_s + R * B_l * K_s + K^2 * K_s}{R * J * K_s + R * B * B_l + R * J_l * K_s + B_l * K^2}} + 1 \right] * s} \tag{A.25}$$

Using the general form (A.26) of the transfer function for the type-one second-order system, we can extract (from A.25) the following:

$$\frac{\theta(s)}{U(s)} = \frac{K_{eq}}{\left[\left(\frac{j\omega}{\omega_n} \right)^2 + 2 * \xi * \left(\frac{j\omega}{\omega_n} \right) + 1 \right] * j\omega} \tag{A.26}$$

$$\omega_n = \sqrt{\frac{R*B*K_s + R*B_l*K_s + K^2*K_s}{R*J*B_l + R*J_l*B + J_l*K^2}} \tag{A.27}$$

$$\xi = \frac{\sqrt{\frac{R*B*K_s + R*B_l*K_s + K^2*K_s}{R*J*B_l + R*J_l*B + J_l*K^2}}}{2*\frac{R*B*K_s + R*B_l*K_s + K^2*K_s}{R*J*K_s + R*B*B_l + R*J_l*K_s + B_l*K^2}} \tag{A.28}$$

$$K_{eq} = \frac{K*(J_l*s^2 + B_l*s + K_s)}{R*B*K_s + R*B_l*K_s + K^2*K_s} \tag{A.29}$$

Equations A.27, A.28 and A.29, represent natural frequency of the system, damping ratio and equivalent total gain respectively, in case of a system with an inertial load attached to the actuator with a flexible shaft.

These values are easy to calculate for the system when all data are known. In our case J_l and B_l are not known. Using the Bode diagram we can read the values of natural frequency damping ratio and gain for the system. Knowing those values it is easy to calculate J_l and B_l.

Using equations A.27 and A.28, we can obtain formulas for the calculation of J_l and B_l.

For easier presentation, we can denote the following

$$\Gamma = R*B_m + K^2 \tag{A.30}$$

Then we can write:

$$J_l = \frac{K_s*\left[\begin{array}{l}-R^2*J_m^2*\Gamma*\omega_n^3 + 2*\xi*R^2*J_m*B_m*\Gamma*\omega_n^2 + (R^2*J_m*K_s*\Gamma \\ -R^2*B_m*K_s*\Gamma - 2*R*B_m*K^2*\Gamma - K^4*\Gamma)*\omega_n + 2*\xi*R*K^4*K_s\end{array}\right]}{\begin{array}{l}(R^2*J_m*K_s*\Gamma - R^2*B_m*K_s*\Gamma - 2*R*B_m*K^2*\Gamma - K^4*\Gamma)*\omega_n^3 \\ +(2*\xi*R^2*B_m*K_s*\Gamma + 2*\xi*R*K_s*K^2*\Gamma)*\omega_n^2 + R^2*K_s^2*\Gamma*\omega_n\end{array}} \tag{A.31}$$

$$B_l = \frac{K_s * (R * J_m * \Gamma * \omega_n^3 - 2 * \xi * \Gamma^2 * \omega_n^2 + R * K_s * \Gamma * \omega_n)}{-R * (\Gamma^2 - R * J_m * K_s) * \omega_n^3 + 2 * \xi * R * K_s * \Gamma * \omega_n^2 - R^2 * K_s^2 * \omega_n} \quad \text{(A.32)}$$

It can be seen that both J_l and B_l are functions of the natural frequency of the system ω_n.

A.2.4. Experimental determination of the transfer function for the Pittman DC motor with passive load

For this experiment, which is an illustration for previous theoretical analysis, two DC motors Pittman9232 were connected together by a shaft. One motor was considered an actuator and another was a load. The actuator receives the position control signal produced by the dSPACE system, while the second motor's rotor simulates a passive load.

ω	Y/X (1)	20logY/X
0.1	N/A	N/A
0.2	324.55	50.22
0.4	157.1	43.92
0.6	100.15	40.01
1	60.65	35.65
2	28.4	29.06
4	13	22.27
6	7.575	17.59
10	3.65	11.24
20	1.1	0.828
40	0.2925	-10.68
60	0.1153	-18.7
100	0.0425	-27.4

Table A2. Experimental data for two Pittman DC motors connected by a flexible shaft

This was done for easier control of results, since all data for both motors were known. The purpose was to check if the theory of the experimental determination of the moment of inertia J_L and the coefficient of viscous friction B_L works. Using the same

experimental environment, the following data were obtained after several tryouts. Only the average values are shown in the Table 2. Using this set of data, the following graph shown on Figure A.8 was constructed. Asymptotes were drawn through points and it was possible to read the corner frequency of the system $\omega_n = 6 \left[\frac{rad}{s}\right]$. If compared with the Pittman motor frequency characteristic, when the motor did not have any load, this characteristic has smaller bandwidth. That was to be expected since additional load always tends to reduce the bandwidth of the system. The natural frequency obtained mathematically is much larger than the one obtained experimentally. This result is similar to the one analyzed for a single motor without load.

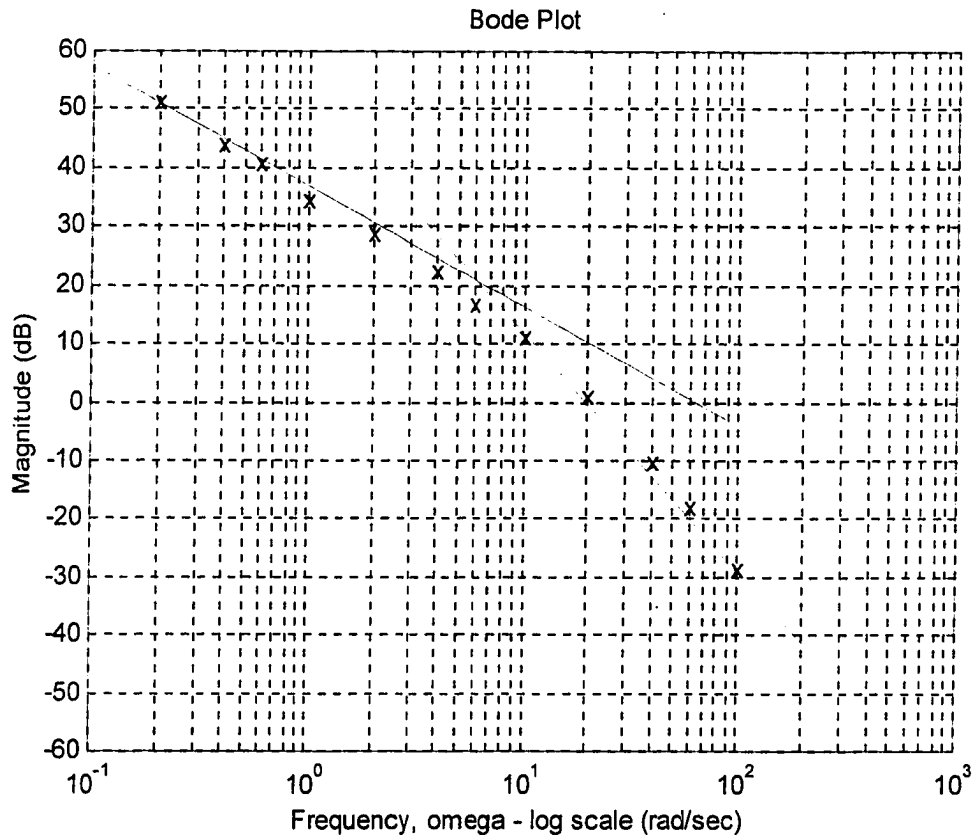


Figure A.8. Experimental data for two motors

APPENDIX B

(Specifications of the Motors, Drivers and Sensors)

B.1. Motor

Motor data	Units	Pittman 9232
Continuous Torque	[Nm]	$16.2*10^{-3}$
Peak Torque [stall]	[Nm]	$97.5*10^{-3}$
Motor Constant	$[Nm/\sqrt{W}]$	$11.4*10^{-3}$
No-load Speed	[rad/s]	734.6
Friction Torque	[Nm]	$3.5*10^{-3}$
Rotor Inertia	$[kgm^2]$	$1.91*10^{-6}$
Electrical Time Constant	[ms]	0.63
Mechanical Time Constant	[ms]	14.4
Viscous damping (zero imp.)	[Nm/rad/s]	$1.83*10^{-6}$
Viscous damping (inf imp.)	[Nm/rad/s]	$1.31*10^{-4}$
Maximum winding temperature	$[°C]$	155
Thermal Impedance	$[°C/watt]$	22.7
Thermal Time Constant	[min]	7.21
Motor weight	[g]	197.9
Motor length	[mm]	46.4
Reference voltage	[V]	12
Torque Constant	[Nm/A]	$15.6*10^{-3}$
Back EMF Constant	[V/rad/s]	$15.6*10^{-3}$
Resistance	$[\Omega]$	1.93
Inductance	[mH]	1.16
No-load Current	[A]	0.32
Peak Current	[A]	6.22

Table B.1 – Mechanical and Electrical characteristics of Pittman 9232 PM DC motor

B.2. Optical encoder

Specification	Remark	Opt Enc.9040
Number of channels		3
Number of pulses per revolution		500
Number of connection pins	#2695	5

Table B.2 – Optical encoder data

B.3. Power amplifiers

PA228:

Specification	Units	PA228
Power supply		
Power supply	[V]	19-32
Amperes – full load	[A]	7.5
Output rating		
Power	[W]	160
Internally limited – Amps	[A]	7.5
Gain characteristics		
Open loop pre amp	[kV/V]	20
Power stage DC gain	[V/V]	5.5
Dimensions		
Length	Inches (mm)	4.7 (119)
Width	Inches (mm)	2.8 (71)
Height	Inches (mm)	0.72 (18)

Table B.3.1 – Specification of PA228 power amplifier

Power amplifier PA228 can work as voltage, current or servo amplifier. To facilitate appropriate feedback configurations, Universal plug-in socket PC-221 was supplied [EG&G Torque Systems 1993]. This socket is a printed circuit card containing standoff terminals for input output wiring and the circuitry required for amplifier conversion into a one of three possible configurations.

If the amplifier is set to work as current amplifier, the transfer function is:

$$\frac{i_o}{e_{input}} = 2 \quad \left[\frac{A}{V}\right]$$

The torque produced by the motor windings can be calculated using the motor constant and amplifier current transfer function. The torque produced per each volt of input is:

$$T = 2 \quad \left[\frac{A}{V}\right] * 0.0156 \quad \left[\frac{Nm}{A}\right] = 0.0312 \quad \left[\frac{Nm}{V}\right]$$

PA12:

Specification	Units	PA228
Power supply		
Power supply	[V]	±10 ÷ ±50
Amperes – full load	[A]	15
Gain characteristics		
Open loop pre amp	[dB]	110
Minimum gain for closed loop	[V/V]	3
Input		
Voltage	[V]	±Vs
Temperature		
Operating temperature range	°C	-55 ÷ +125

Table B.3.2 – Specification of PA12 power amplifier

PA12 power amplifier, made by APEX, is a very high output current operational amplifier, is designed to drive resistive, inductive and capacitive loads [49]. It shows excellent linearity and accepts wide supply voltage in the range from ±10 to ±50 V. The high output current, up to 15 A, and small compact size, makes this device very attractive. It does not come with circuit built for it, but with simple custom-made circuits, it is possible to make this amplifier ready to use in various applications. See chapter 4 for more information about custom-made circuit.

For more information on PA12 visit APEX web page at www.teamapex.com .

B.4. Pulleys

Specification	Units	Pulley A6A16M090NF6008
Number of grooves		90
Diameter	[mm]	58.2
Bore diameter	[mm]	8
Length	[mm]	19.1

Table B.4 – Shaft pulley specification

Specification	Units	Pulley A6A16M012DF6003
Number of grooves		12
Diameter	[mm]	7.8
Bore diameter	[mm]	3
Length	[mm]	14.3

Table B.5 – Tachometer pulley specification

The following procedure was used to determine length of timing belt used in experimental setup:

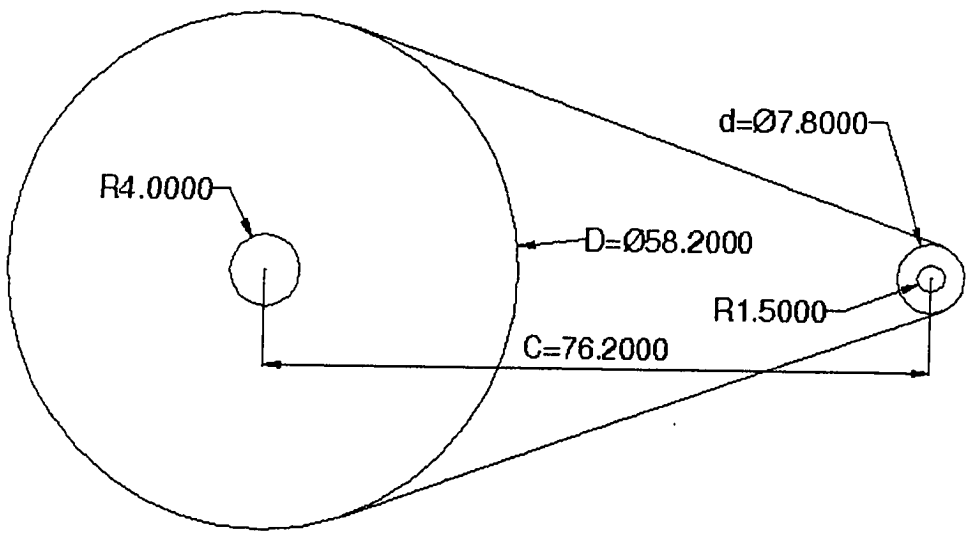


Figure B.1 – Pulleys layout

According to the pulleys layout the following formula [59] was used to determine the length of the timing belt:

$$L = 2*C + 1.57*(D+d) + (D+d)^2/4*C \qquad \text{(B-1)}$$

According to the Figure B.1 and the formula B-1 resulting length is $L_b = 264.21$ $[mm]$.

Using the table from Stock Drive Products manual the timing belt has to have 130 grooves.

The ratio produced by using these pulleys is $R_p = 7.46$

B.5. Tachometer

Specification	Units	Tachometer
Tachometer constant	[V/rpm]	20.8/1000
Tachometer constant	[V/rad/s]	0.198625
Ripple Voltage	[%]	3

Table B.5 Tachometer specification

APPENDIX C

(Technical drawings for the parts of the experimental setup machined in the shop)

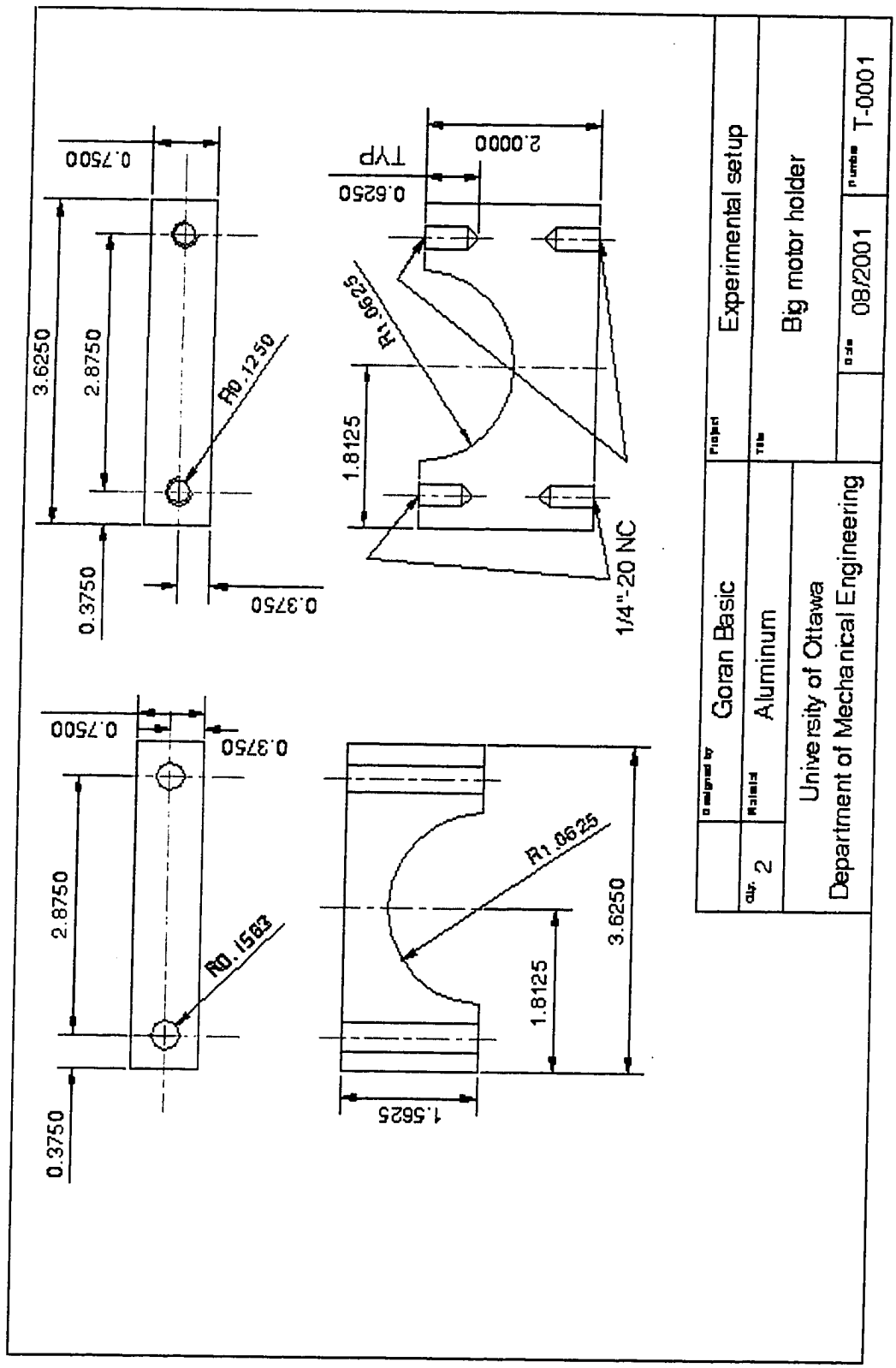


Figure C1 – Motor holder – big diameter

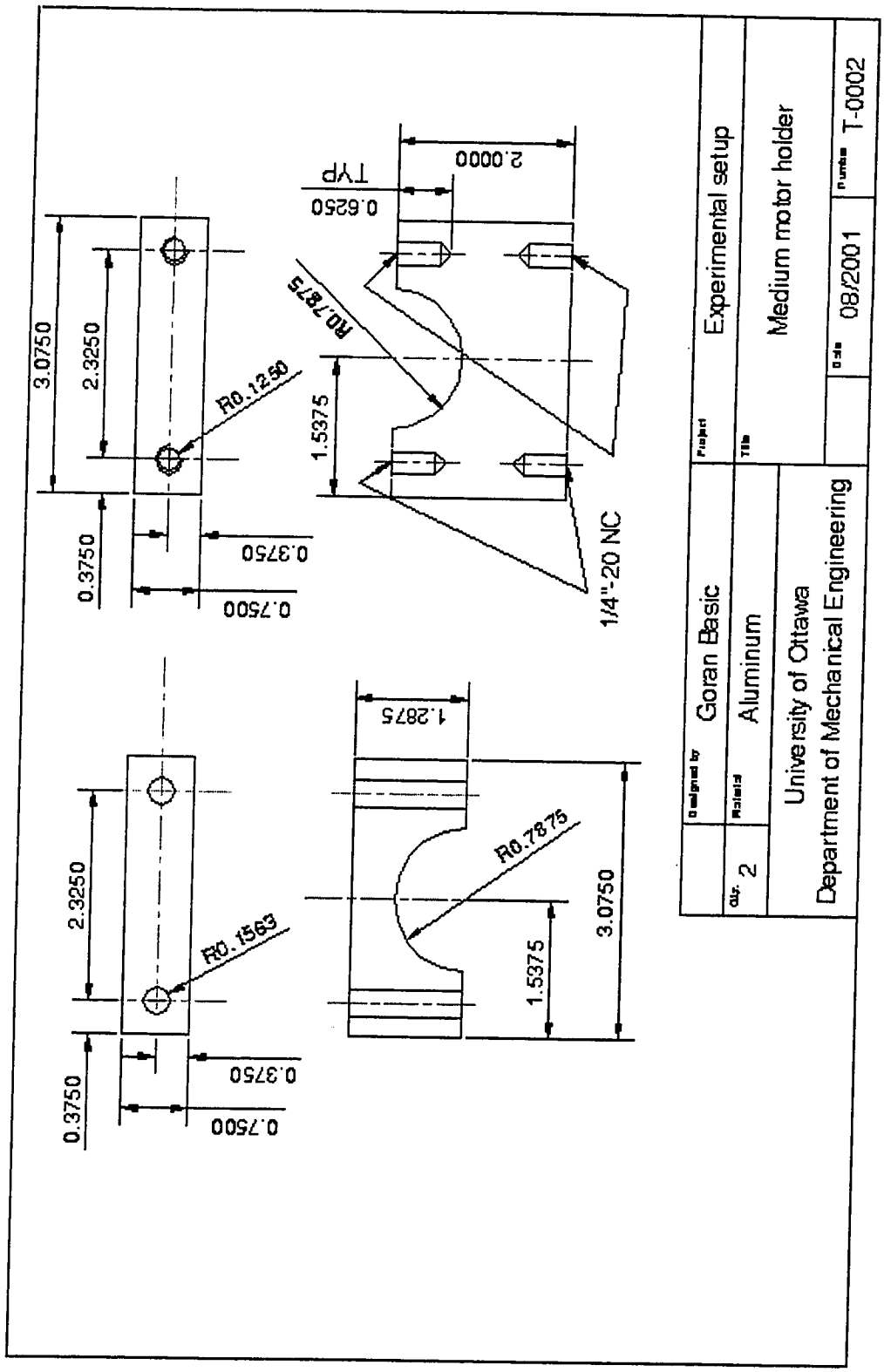


Figure C2 – Motor holder - medium diameter

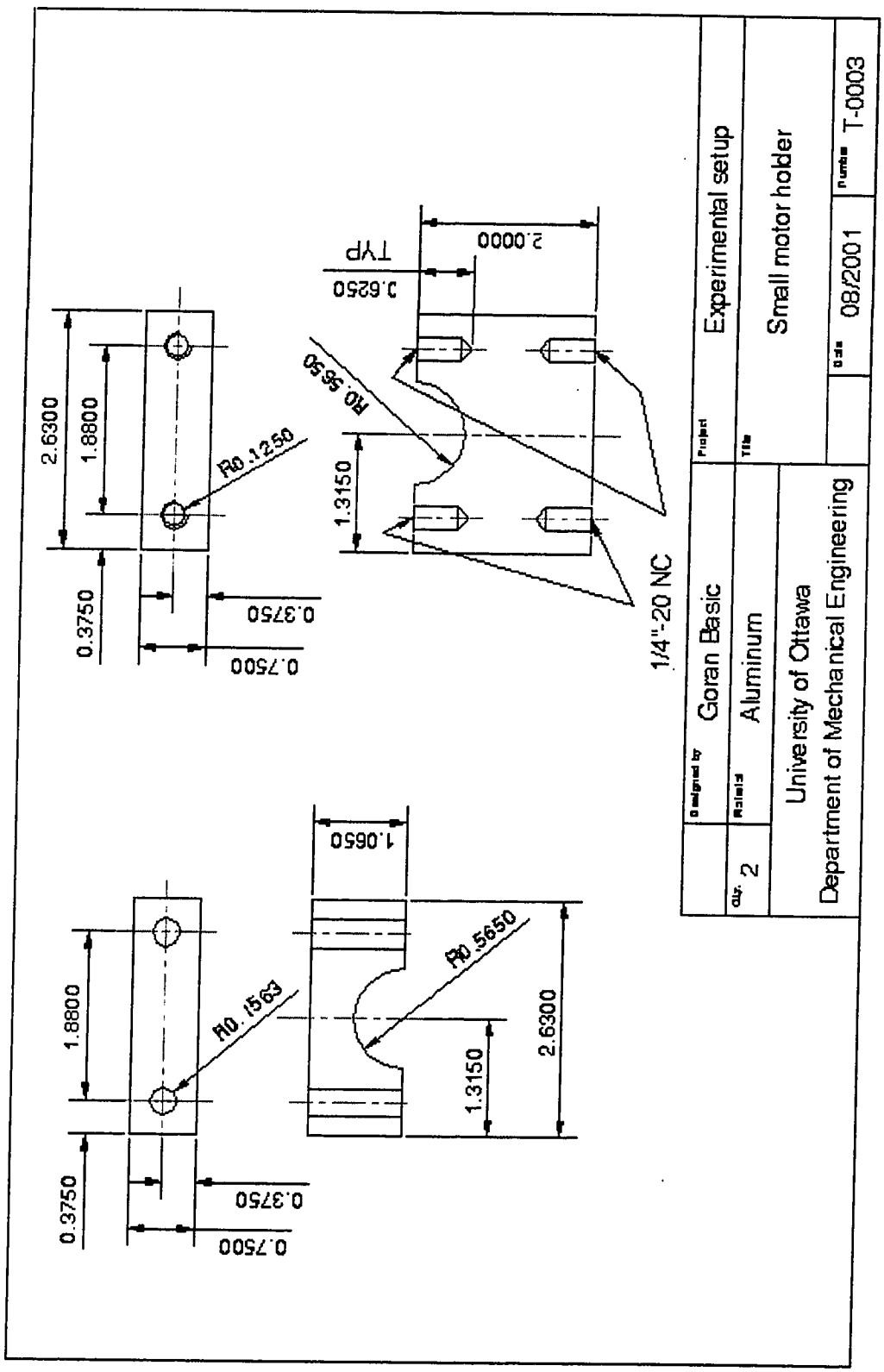


Figure C3 – Motor holder – small diameter

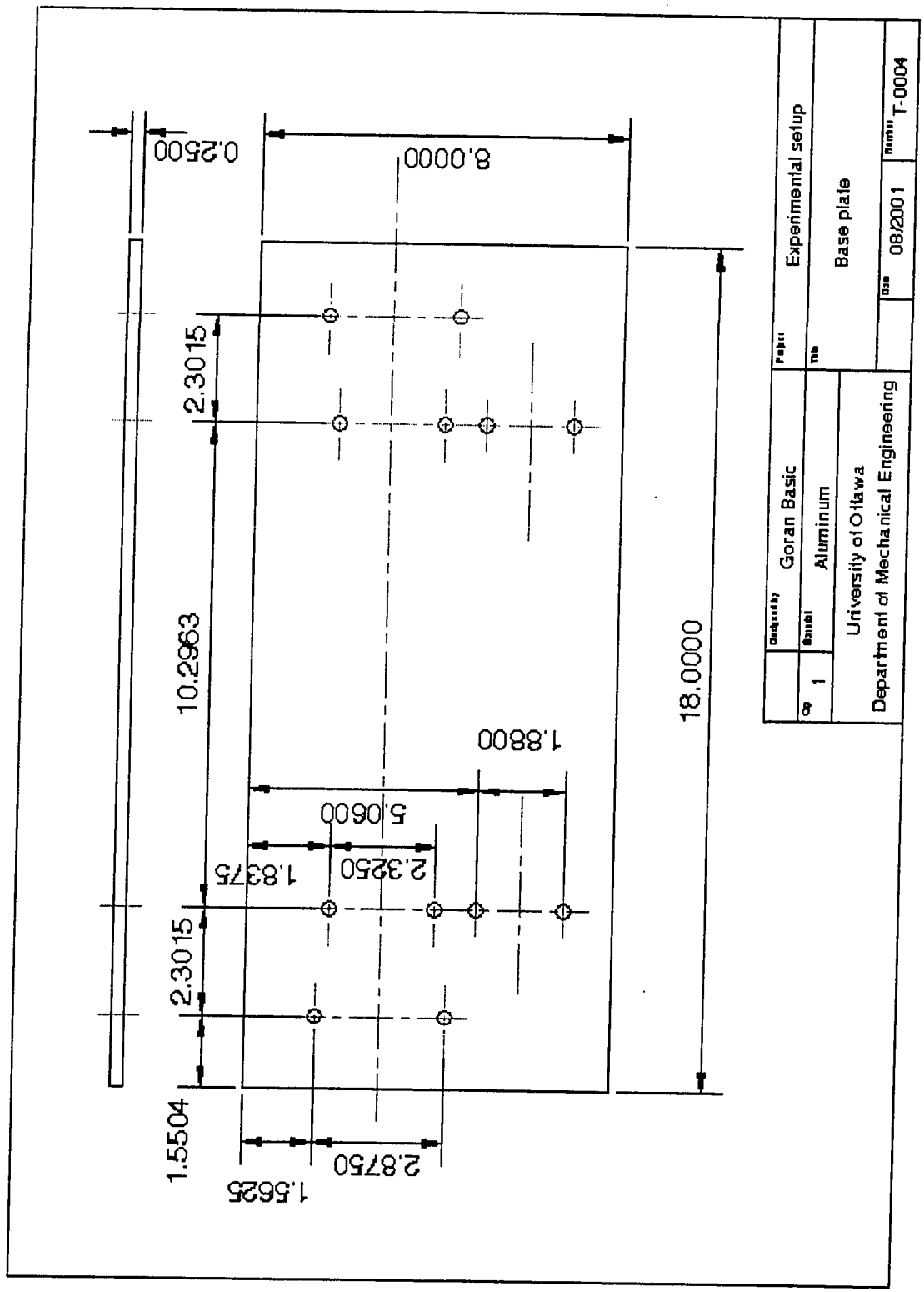


Figure C4 – Base plate

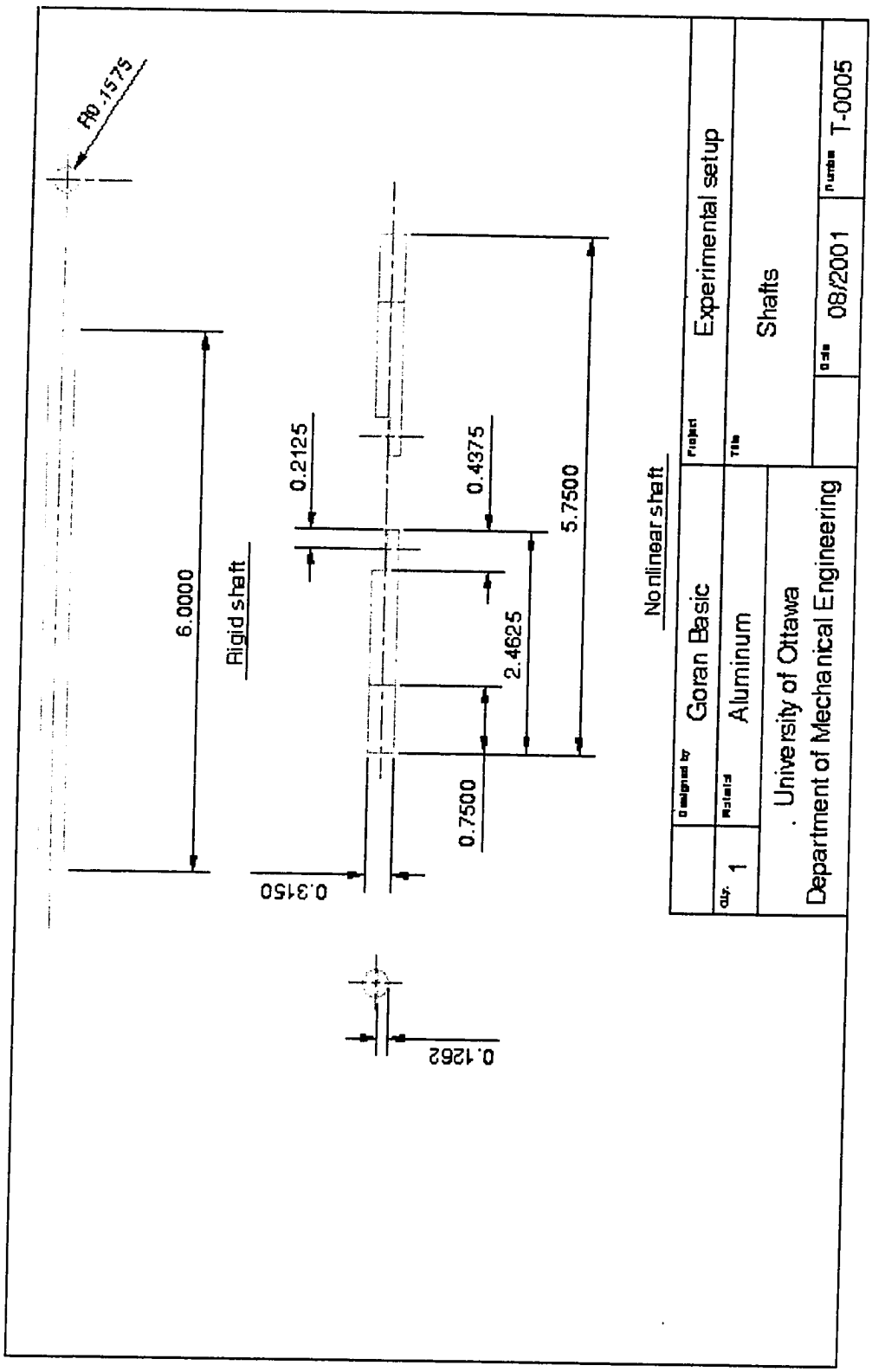


Figure C5 – Shafts

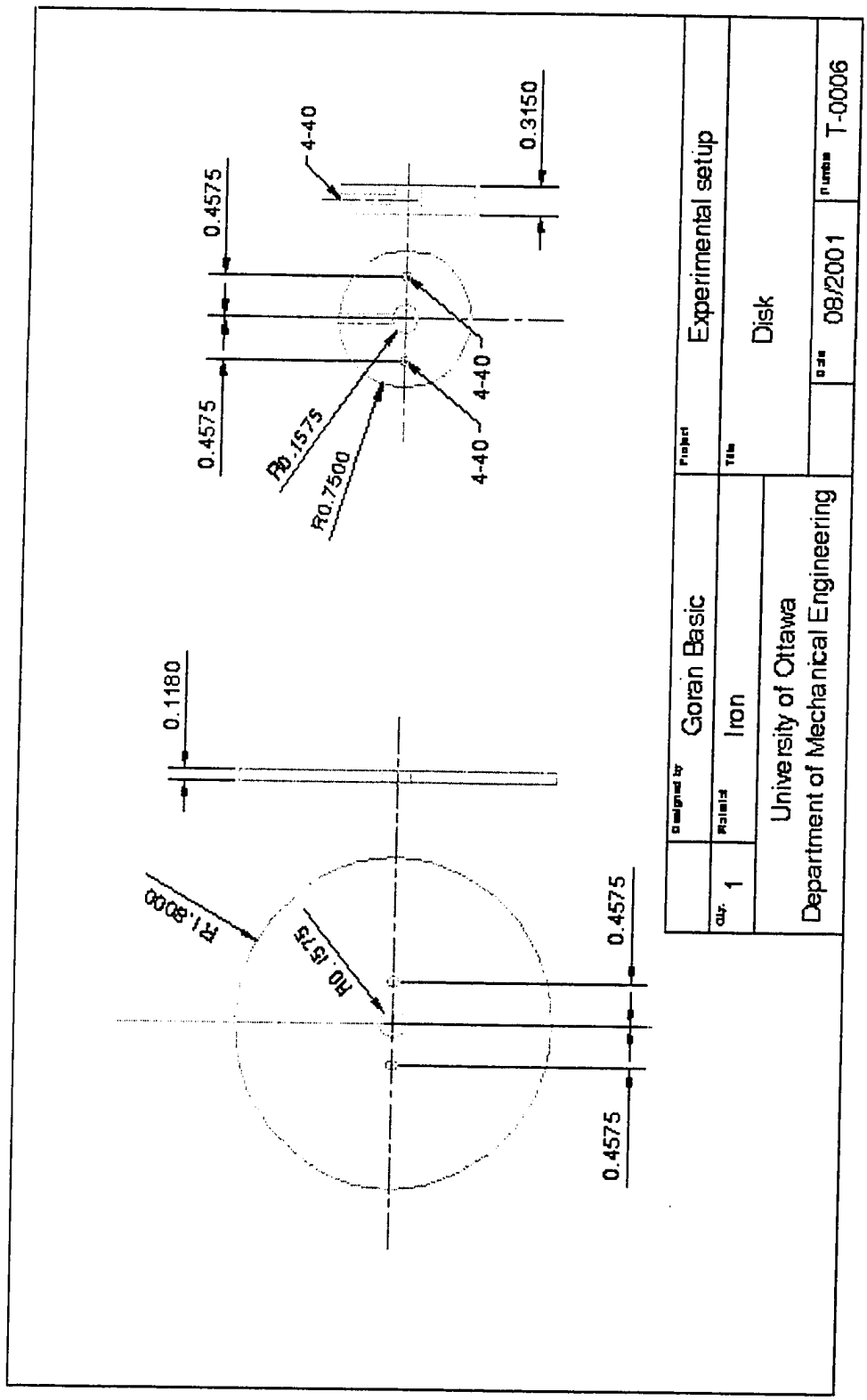


Figure C6 – Disk

APPENDIX D

(PA12 power amplifier – preparing, balancing and troubleshooting)

D.1. Introduction

The following Appendix describes the procedure on how to setup, balance and troubleshoot PA12 power amplifiers. It is intended to serve as a short introduction to the topic. More information about the amplifier can be obtained on the APEX web Application notes [49]. More details about balancing and amplifier circuits can be found in books about amplifiers, for example [50]. The information included is from the above two references.

D.2. Voltage controlled current source layout

The amplifier PA12, manufactured by APEX, comes as an eight-pin chip, where pins are set in the circular configuration. Such configuration is called TO-3 package and it is a standard chip package. Figure D.1 shows the TO-3 package and designates the pins. In is the input, Vs is the voltage source, Cl is the current limit, F.O. is the foldover and OUT is the output.

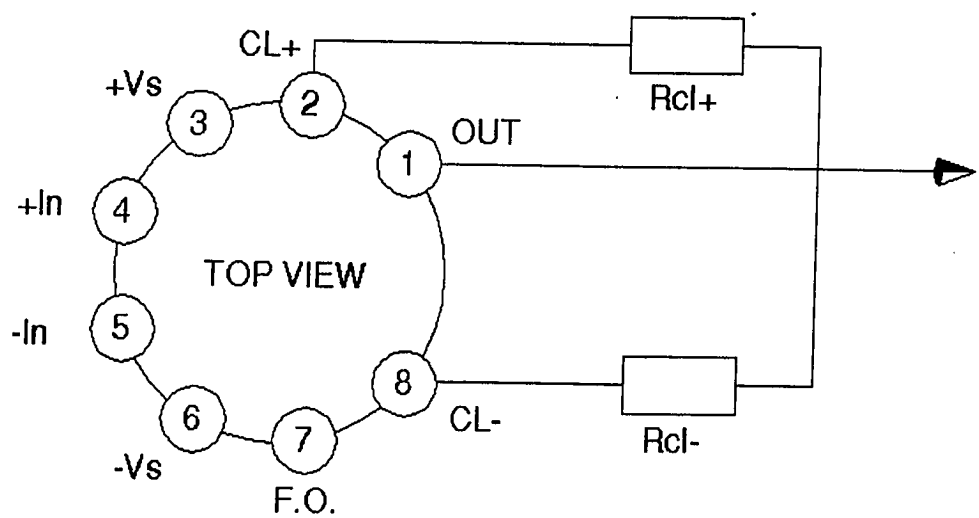


Figure D.1. TO-3 package

It is very important to include current limit resistors. Since amplifiers are capable of delivering large currents, it is necessary to limit that amount to prevent damaging of components in the circuit. The basic formula for the static current limitation is

$$I_{cl} = \frac{0.65}{R_{cl}}$$

where I_{cl} represents maximum current delivered by amplifier and R_{cl} is current limiting resistor.

For the purposes of this thesis, the PA12 power amplifier was considered to be used as an amplifier with the current feedback, since it was used for the second part of experiments, when feedback was introduced. See chapter 4 for more information.

PA12 Power amplifier is recommended for gains higher than -3 in inverting configuration and higher than +4 in non-inverting configuration. The inverting configuration was used in this thesis. The figure D.2 shows the circuit built around PA12 in inverting configuration. Chosen resistors give amplification of 3.9, which can be calculated by formula for inverting amplification

$$A = \frac{-R_f}{R_1} \qquad (D.1)$$

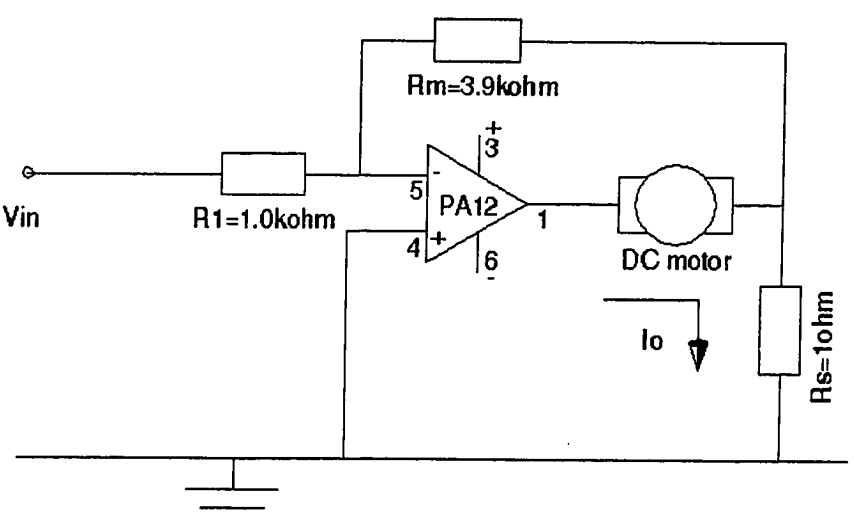


Figure D.2 Voltage controlled current source circuit

The DC motor in this configuration represents a floating load. According to the Kirchhoff's rule and the knowledge about amplifiers, we can derive the transfer function of the amplifier shown in the Figure D.2.

$$\frac{i_o}{V_{in}} = -\frac{R_f}{R_1}\frac{1}{R_s} \tag{D.2}$$

The explanation for this formula is simple. The voltage between inputs 4 and 5, for the balanced amplifier, has to be zero. In addition, the amplifier input impedance is infinite so that there is no current flowing into the amplifier. The current is flowing only through the branches with the DC motor and the R_s resistor. To keep the balance, the input voltage is the same as the voltage across the R_s resistor. In that case, the relationship is

$$V_{in} = i_o * R_s \tag{D.3}$$

The equation D.3 is combined with the equation D.1 producing the D.2 transfer function.

When choosing the R_s resistor, it is important to verify that it can handle the current limited by the current limit resistors. For example, the resistor chosen for this research has the following specification: $R_s = 1 \ \ [ohm]$ and power dissipation is $P = 10 \ \ [W]$. In this case, resistor can handle up to 3.16 amperes of current without being damaged.

D.3. Amplifier balancing

Amplifiers are active devices and it is not possible to produce them without internal imperfections. One of them is unbalanced input, which means that when the power source is connected to the amplifier, there is a voltage difference between plus and minus input pins. The consequence of having unbalanced input is nonzero output for zero input, which is noticeable immediately after connecting the power supply to the amplifier. For example, if a DC motor load is connected, and input to the powered amplifier is zero, the motor will still turn due to the unbalanced inputs.

To overcome that problem, a simple balancing solution is proposed. Figure D.3 shows the circuit with the balancing addition. Since the inverting configuration was used for the amplifier, the balancing will be achieved on the side of noninverting input. The

resistor R_a represents a potentiometer, which is connected to the power source, while the adjusting leg is connected to the resistor R_b. The rule for choosing resistors is $R_b > R_a > R_c$ and the following relationship is valid for covering the whole range of the unbalanced voltage.

$$V_{io} = \frac{R_c}{R_b} * V \tag{D.4}$$

where V is the supply voltage and is equal to $V = |+Vs| = |-Vs|$, and V_{io} is maximum offset voltage read from the amplifier data sheet or measured on the amplifier inputs.

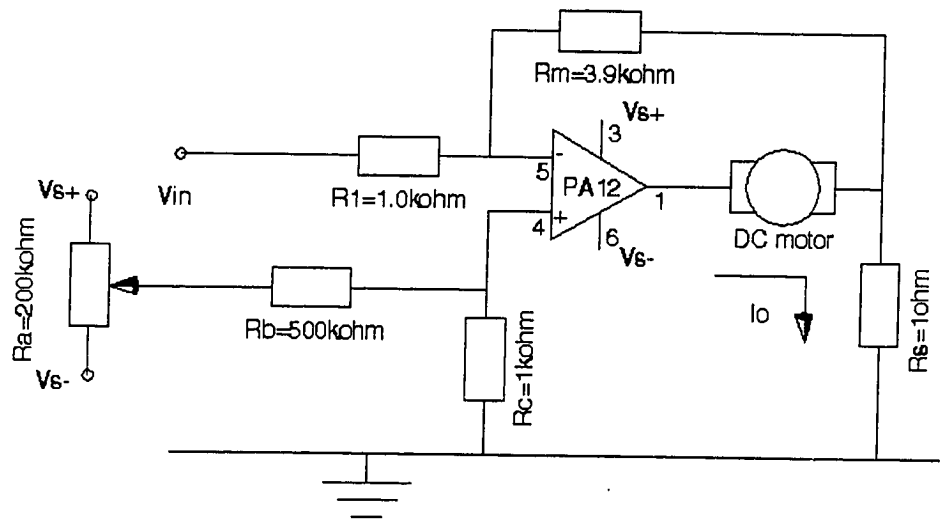


Figure D.3. The circuit with the balancing addition

The balancing principle is based on the following procedure: when one of the inputs has more voltage than the other, the method is to add or subtract the voltage on the noninverting input. If the inverting input has more voltage than the noninverting input, than by adjusting the middle pin on the potentiometer we can add the required amount of the voltage to the noninverting leg accomplishing the balance between two inputs.

D.4. Troubleshooting

The following are the problems encountered during the research. For all following examples, amplifier is considered to be balanced.

If the output oscillates, (and this can be checked on the oscilloscope), and the input voltage is zero, we have to consider adding a second feedback loop containing a resistor and a capacitor. The calculation method for the resistor and the capacitor can be found in the Application notes 19 from APEX. This way the circuit obtains the stability loop.

If the amplifier output oscillates with zero input voltage, consider adding bypass to power supply. To each power supply pin, we should add two capacitors. One of them should be of the value $0.1-0.22\ \ [\mu F]$ and should be connected as close as possible to the pin. The second capacitor should be of the value $10\ \left[\frac{\mu F}{A}\right]$ of the peak output current and should be connected to the pin at the distance at least 2 inches away from the pin. All connected capacitors should be grounded.

It is important to check all connections to the ground. Sometimes there is potential between two grounded points and it causes ground currents, which lead to the amplifier instabilities.

It is required not to use the same power source for PA12 and the amplifier of any other type. That may cause unexpected currents and input voltage imbalances.

If not possible otherwise, a heat sink is required for the amplifier since power dissipation is significant. A fan to cool down the amplifier can be used for this purpose.

APPENDIX E

(Computer code)

Note:

Computer source code will be available on request, to those interested in this research.

APPENDIX F

(Tutorial on how to use dSPACE)

F.1. Environment

The dSPACE operating system contains the following boards:

ds1002 – processor board

ds2002 – MUX ADC board

ds2101 – DAC board

ds3001 – incremental encoder board

The dSPACE operating system software:

SED30 – setup software

MON30 – uses a setup file and downloads object file to the processor board. Used to reset and hold signals

TRACE30 – traces signals in real-time

Files created by a user in the order to run experiments:

.C – a source file containing a program written in C programming language

.TRC – a file that contains variables to be traced by TRACE30 – has to have the same name as the C file

.STP – a setup file created using SED30 software, needed for each C file. It sets up channels used on boards for that particular experiment – has to have the same name as the C file. There is main setup file in the computer. Do not edit that file. Each of your applications should have separate setup file.

.BAT – a batch file. This file contains should contain instructions to the C compiler and the linker

Files created by the machine after compilation:

.MAP – a mapping file. It links all addresses of variables to be traced [39][40].
.OBJ – a compiled C file. The file loaded by MON30 to the processor board ds1002.

The dSPACE computer is connected to the host PC computer, which is used by a user to communicate with the dSPACE. All commands to the dSPACE are given from the host computer. The dSPACE executes a program module in real-time and is not interrupted by other processes.

F.2 Procedure

Note: For all steps that follow, the host PC computer is considered to be on.

Preparation steps:

Building files

1. Build a C program, which should contain instructions for the boards and calculations required for an experiment. This will be your control program.
   Save program as a .C file

   Example: *MyProgram.C*

2. Compile the C program. Use .BAT file prepared earlier. As a result, you should obtain .OBJ file.

   Example: prepared .BAT file is Complink.BAT
   At command line type: *Complink MyProgram*

3. Create .TRC file. Use an example, which is stored in the computer. This file should

contain names of variables that you want to trace with TRACE30 utility. Save file as .TRC with the same name as the .C file

Example: *MyProgram.TRC*

4. Turn on the dSPACE.

Channel setup

5. Type: *sed30* ->Enter

6. Press **1** to enter the name of the setup file

Example: *MyProgram* ->Enter

7. Press **4** to edit a monitor settings.

Chose a board to edit channels. For the each board assign channels that will be used by your application. Those channels should be called inside of your C program.

8. Press **q** to quit setup utilities (this will save your new setup file)

Example: *MyProgram.STP*

Press **q** one more time to exit the SED30 program.

Tracing setup

9. Type: *trace30* ->Enter
10. When TRACE30 program opens scroll down to second line "Enter the filename to trace the signals" and press Enter. Enter the name of trace file that will be used in this session.

Example: *MyProgram.TRC* -> Enter

11. Scroll down to he "Setup" line and press Enter. Edit the tracing time required by your application.
12. Scroll down to the END and press Enter to end the TRACE30 setup.
13. Turn off the dSPACE to finish setting procedure.

Note: The steps 5-12 are necessary to be completed only once for the particular experiment (C program). In case of changing the experiment, it is necessary only to complete step 10 to load the name of the different TRC file. Steps 1 and 2 can be repeated as many times as needed. If a C program is changed and compiled, it is not necessary to repeat steps 3-12 again if settings for new program are still satisfactory.

Execution steps:

Note: The following steps are to be repeated every time when experiment is to be conducted. These steps cannot be started unless all preparation steps are completed.

14. Turn on dSPACE
15. Type: *mon30* ->Enter
16. Press **1** to load an object module. Enter the name of the file you want to load. This step downloads the object (compiled) C file to the dSPACE processor.

    Example: *MyProgram* ->Enter (no extension)
    ->Enter
17. Press **2** to restart DSP. This is starting execution of the downloaded program.
18. Press **q** to leave DSP running but to quit the MON30 program.
19. Type: *trace30* ->Enter
20. Scroll down to highlight Trace and press Enter. Tracing program is activated and it will last according to the time given in step 11.
21. When the Trace stops, (it becomes highlighted again), go to the plot and press Enter. This will display graphs of traced variables set in the step 3.
22. If needed the plot can be saved on disk or printed by pressing **p** on your keyboard.
23. Enter **Esc** to go back to main menu.
24. Scroll down to highlight End and press Enter to exit TRACE30 program.
25. Type: *mon30* ->Enter ->Enter
26. Press **d** to quit MON30 program.
27. Turn off the dSPACE.

APPENDIX G

(Modeling analogy)

G.1. Linear analogy

Let us consider a simple example (Figure G.1).

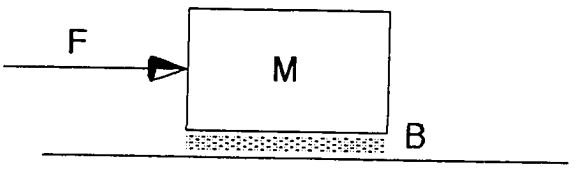


Figure G.1 Mass M moved by a force F

Assume a mass M subject to a force F. Between the mass and the ground there is a film of the oil characterized by the damping coefficient B. For the force $F(t)$, which moves the mass over the distance x the following equation of motion can be written:

$$M\frac{d^2x}{dt^2}+B\frac{dx}{dt}=F \tag{G.1}$$

Using Laplace's transformation with zero initial conditions, we can write:

$(Ms+B)*v=F$ for $v=\frac{dx}{dt}$.

Transfer function, which considers velocity as an output and force as an input, is:

$$\frac{v}{F}=\frac{1}{Ms+B} \tag{G.2}$$

This was the case of the system with real load. The same system can be modified by using different load. Instead of using mass M we can attach different mass M_1 (Figure G.2).

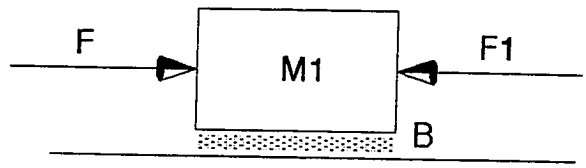


Figure G.2 A system with mass $M_1 < M$ and a compensating force F_1 used to simulate the system shown in Fig. G.1.

In this case, the system would be described as follows. We know that mass M_1 can be heavier or lighter than mass M. To consider only one case we will assume that $M_1 < M$. To obtain the same transfer function as in the first "real load" case with mass M we have to add a force acting on the mass M_1 in the opposite direction to simulate the effect of M-M1. Let us call that force F_1. In this case, using $M_1 < M$ and F_1 the system shown in Figure G.2 can be made to behave as the system shown in Fig. G.1. Again, we can write an equation using D'Alembert principle:

$$M_1 \frac{d^2x}{dt^2} + B\frac{dx}{dt} = F - F_1 \tag{G.3}$$

After applying Laplace transformation, we can obtain the following transfer function.

$$\frac{v(s)}{F - F_1} = \frac{1}{M_1 s + B} \tag{G.4}$$

For mechatronics objects, inside of transfer function should remain the same since we can change only inputs and outputs (effort-flow concept). To preserve the original right side of the transfer function, the mass M will be represented as composition of two masses

$$M = M_1 + M_{sim} \tag{G.5}$$

where M_{sim} is the mass simulated by F_1. It is possible, using this notation, to rewrite equation G.3.

$$M\frac{d^2x}{dt^2} - M_{sim}\frac{d^2x}{dt^2} + B\frac{dx}{dt} = F - F_1 \tag{G.6}$$

After applying Laplace transformation and rearranging numbers the following transfer function is derived.

$$\frac{v(s)}{F - F_1 + M_{sim} * s * v(s)} = \frac{1}{Ms + B} \qquad (G.7)$$

The final transfer function (G.7) retains all inside features of the object (right hand side of the function) as in original state with real passive load M-B. The only difference is achieved with effort-flow variables F_1 and v from the left hand side. To make that side the same as the one in original transfer function we have to eliminate the extra terms by equating them with zero.

$$-F_1 + M_{sim} * s * v(s) = 0$$

or

$$F_1 = M_{sim} * s * v(s).$$

Using inverse Laplace transformation it can be shown that

$$F_1 = M_{sim} * \frac{dv}{dt} \qquad (G.8).$$

This equation shows what is the value required for the force F_1 at any given moment. Given that M and M_1 are known and that the acceleration of the mass is easy to measure, the simulation of the system with passive load M-B is feasible with F_1 and $M_{sim} = M - M_1$.

This simple example illustrates the basic concept on which this thesis was developed. The term simulated mass will be used throughout the thesis and it will represent the virtual mass needed to create a compensation force. Here, we presumed mass M_1 to be smaller than mass M. For the case of $M < M_1$, the force F_1 would be the same in scalar value and opposite in acting direction.

G.2. Rotational analogy

Previous example showed a translational motion case with linear force acting on the mass. The following example illustrates the case of a rotational motion. Figure G.3 shows a load driven by a DC motor.

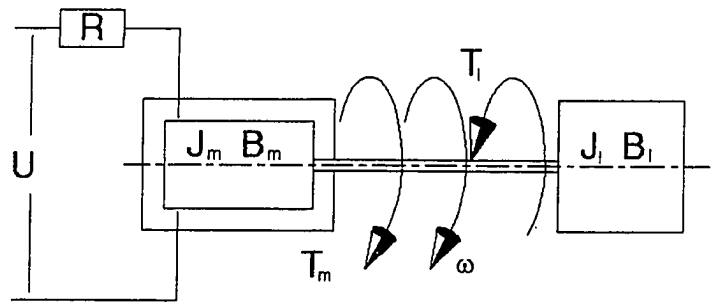


Figure G.3 A DC motor driving a inertial-viscous load

A load is attached to the motor with some coupling, which will be considered stiff enough such that in this case the velocity of the load will be the same as the velocity of the motor. The inertia of the motor's rotor is J_m and the coefficient of viscous damping is B_m. The inertia of the load is J_l and the coefficient of viscous friction of the load is B_l. Torque produced by load dynamics T_l opposes motor torque T_m and acts in direction opposite to the direction of the angular velocity ω. The value of the torque T_l is

$$T_l = J_l * \frac{d\omega}{dt} + B_l * \omega \qquad \text{(G.9)}$$

Applying the D'Alembert principle, we can write:

$$(J_m + J_l) * \frac{d\omega}{dt} + (B_m + B_l) * \omega = T_m \qquad \text{(G.10)}$$

where $(J_m + J_l)$ and $(B_m + B_l)$ represent lumped inertia and lumped coefficient of viscous friction of the system, respectively.

After applying Laplace transformation on the equation (G.10) and rearranging it, the following transfer function with angular velocity as an output and motor torque as an input is obtained as:

$$\frac{\omega(s)}{T_m(s)} = \frac{1}{(J_m + J_l) * s + (B_m + B_l)} \tag{G.11}$$

This transfer function represents the system with real load. The same system can be modified, and instead of the original load, we can connect a different load to the motor. Let new load is considered to have inertia J_{l1} and coefficient of viscous friction B_{l1}. The new system is shown in figure G.4.

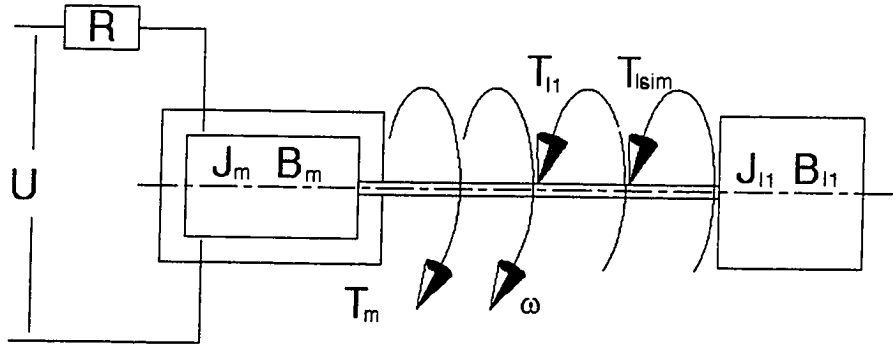


Figure G.4 A DC motor and new load with compensating torque

Again we can presume that new load is "lighter" than original and that both $J_l > J_{l1}$ and $B_l > B_{l1}$, which is just one of four possible cases. The toque produced by the dynamics of the new load T_{l1} acts again in opposite direction to the angular velocity and has the value

$$T_{l1} = J_{l1} * \frac{d\omega}{dt} + B_{l1} * \omega \tag{G.12}$$

To be able to get the same angular velocity as for the load J-B, the system has to be affected by an additional torque T_{lsim}, which has to act on the same direction as the load torque. For the system described, the following equation can be derived:

$$(J_m + J_{l1}) * \frac{d\omega}{dt} + (B_m + B_{l1}) * \omega = T_m - T_{lsim} \qquad \text{(G.13)}$$

and the transfer function obtained from it is in the following form:

$$\frac{\omega(s)}{T_m(s) - T_{lsim}(s)} = \frac{1}{(J_m + J_{l1}) * s + (B_m + B_{l1})} \qquad \text{(G.14)}$$

As it was mentioned before, for mechatronics objects, inside of transfer function, should remain the same since we change only inputs for which we will obtain corresponding outputs. To preserve the original right hand side of the transfer function, inertia of the original load J_l and coefficient of the viscous friction B_l will be represented as a composition of two parts.

$J_l = J_{l1} + J_{lsim}$

$B_l + B_{l1} + B_{lsim}$

where J_{lsim} and B_{lsim} are simulated inertia and simulated coefficient of viscous friction respectively. This is valid only if the original inertia and damping coefficient are smaller than the same of the original load is. In other cases, the sign would change accordingly. Now it is possible to rewrite original equation G.13 as follows:

$$(J_m + J_l) * \frac{d\omega}{dt} - J_{lsim} * \frac{d\omega}{dt} + (B_m + B_l) * \omega - B_{lsim} * \omega = T_m - T_{lsim} \qquad \text{(G.15)}$$

Rearranging and using of Laplace transformation, the following transfer function results:

$$\frac{\omega(s)}{T_m(s) - T_{lsim}(s) + (J_{lsim} * s + B_{lsim}) * \omega(s)} = \frac{1}{(J_m + J_l) * s + (B_m + B_l)} \qquad \text{(G.16).}$$

This transfer function looks similar to the original transfer function for real load. The only differences make factors assigned as simulated torque, inertia and damping on the side of the input. If we eliminate them by equating them with zero the transfer function will be the same as the one for original load.

These terms are eliminated, when

$-T_{lsim}(s) + (J_{lsim} * s + B_{lsim}) * \omega(s) = 0$

or, by using inverse Laplace transformation:

$$T_{lsim} = J_{lsim} * \frac{d\omega}{dt} + B_{lsim} * \omega \qquad \text{(G.17)}$$

The equation G.7 shows the method of calculation for the simulated load. This way we know exactly the value of the compensation at every moment of the simulation since angular velocity can be measured and simulated coefficients are easily calculated.

The previous two simple examples represent the theoretical basis for the implementation of real simulations and is used in the thesis. In reality, it is possible to simulate various linear and nonlinear loads by using different passive loads and adding or subtracting compensational torques that compensate for the difference in dynamics. In this thesis, this compensation is implemented using an active load achieved by a DC motor. Motor torque is therefore used to simulate the difference in reaction torques.