Robust Servo Design for Dual-Stage Disk Drives

by

Daniel J. Hernandez

Submitted in partial satisfaction of the
requirements for the degree of

Master of Science

in

Mechanical Engineering

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA at BERKELEY

December 1998

ABSTRACT

A dual-stage disk drives consist of a primary actuator (the VCM), which is used mainly for track seeking and a secondary actuator (a microactuator), which is used for high bandwidth, high accuracy track following. In this report, the design of multi-input, multi-output and single-input, multi-output controllers for dual-stage disk drives using mu-synthesis is discussed and several case studies are presented. Mu-synthesis is a modern controller design method that uses an iterative H_∞ technique to design high performance controllers for linear time-invariant systems with norm bounded uncertainties. It has been found that bandwidths of a couple of kilohertz can be achieved using either electrostatic microactuators that move the head at the end of the sliders or piezoelectric actuators that move the suspension from the base of the suspension. With bandwidths this high, it should be possible to achieve track densities from 25,000 to 100,000 tracks per inch. A Matlab program with an extensive GUI was written for quickly and easily designing controllers for dual-stage disk drives using the μ-Analysis and Synthesis Toolbox.

ACKNOWLEDGEMENTS

I would like to thank Professor Roberto Horowitz for providing me with a great lab to work in and also for his guidance during my graduate research. I would also like to thank Professor Andy Packard for being so enthusiastic about control theory and teaching, and for being such an all-around nice guy. Many thanks also to Sungsu Park, a friend and co-worker who helped me on this project. Lawrence Lee is a close friend of mine who brought so much life to the lab from the moment he arrived, and to him I am very indebted. Sat Pannu, who welcomed me to the lab, provided so much guidance, was always candid, and was always a friend, is another person that I need to thank. And, of course, thanks to Candace. We are always at each other's side.

CHAPTER 1

Introduction

1.1 An Overview of Disk Drive Mechanics

Disk drives are remarkably complex integrations of mechanical and electrical systems that store data magnetically on rotating disks. A schematic of a disk drive is shown in Figure 1.1. The spindle motor spins the disks at speeds typically ranging from 5,400 to 10,000 rpm with accuracies of about 0.1%. The data is stored on the disks in concentric tracks that are typically about 4 microns wide. A group of tracks on different disk surfaces with the same track number is called a cylinder. Each cylinder is divided up radially into sectors, which are typically 512 bytes long, with about 50-256 sectors per cylinder. Hence, disk drives read and write data in blocks of 512 bytes, and cache the contents of several sectors in memory. Data is only read or written using one head at a time, since the associated electronics, collectively referred to as the read channel, are expensive. Data is recorded to and read from the disks using the magnetic read and write elements embedded in the trailing edge of the slider, which is visible at the end of the arm assembly. Although error correction code is used, it is important for the heads to be centered over the middle of the desired track when the drive is reading data. It is absolutely critical for the heads to be centered over the center of the track when the drive is recording data, or else data will be unintentionally and permanently obliterated. The measured value of the distance from the read head to the track center line is called the position error signal (PES), and is read directly from the disk. The PES is a random variable with a nearly Gaussian probability distribution. As a result, the industry standard figure of merit for tracking performance is the 3σ value of the distribution, which by definition means that the magnitudes of 99.57% of the PES samples would be smaller than this number if the distribution were truly gaussian. This figure of merit is called the track misregistration (TMR).

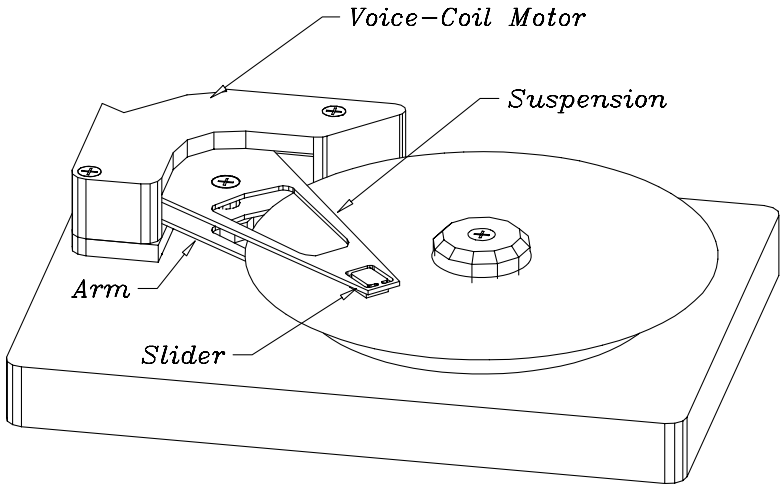


Figure 1.1 *Schematic of disk drive.*

The read/write heads are positioned over the tracks by the voice coil motor (VCM), which rotates the arm assembly over the disks. The mechanics of the arm assembly are fairly complex. The arm itself can safely be treated as a rigid body up to frequencies of about 7 kHz, however, the suspension and gimbal are flexible structures by design. The slider flies on an air bearing between 20 to 60 nm over the disk surface. The purpose of the gimbal and suspension are to allow the slider the freedom of motion to safely fly over the surface. The gimbal allows the slider to pitch and roll while the suspension allows the slider to translate vertically. However, the first torsional mode of the suspension is typically between 2 to 4 kHz, and after this mode the dynamics of the arm assembly become very complicated and uncertain.

A feedback control algorithm running on a DSP or a microcontroller uses a measurement of the head to track center line misalignment called the position error signal (PES) that is read directly from the disk to center the head over a track. The PES is read as the head flies over each servo sector, which is a narrow wedge-shaped section of the disk surface that goes from the inner diameter to the outer diameter. There are typically 50-80 servo wedges at evenly spaced angles on a disk, hence the system is a sampled-data system. The PES is read by integrating in time the signal from the read head as it flies over a checkerboard-like pattern in the servo wedge which is written by a special high-accuracy instrument called a servowriter. Each magnetized square of the checkerboard pattern is called a servo burst and by subtracting the integrals of the signals as the head flies over two consecutive bursts, a measurement of how far the head is from the track center is generated. There are typically 4-6 servo bursts that are offset from each other to achieve greater accuracy and so that displacements greater than a half of a track will not be confused. Until recently, the standard control algorithm was a PID controller with a notch filter at the frequency of the first

resonance. Currently, the standard control law uses a reduced order state estimator with state feedback and a notch filter. For more complete discussions of disk drive mechanics and servomechanics, the reader is referred to [1], [2], [3], [4] and [5]. TMR requirements and the different types of TMR are discussed in detail in [7]. More information on PES sensing and demodulation can be found in [6] and [4].

1.2 Issues in Increasing Areal Density

A major goal for each company in the disk drive industry is to increase the areal bit density on the disk surfaces of their drives as quickly as possible. If a company can pack more data on to its disks while the prices of the rest of the components remains roughly the same, then the company can sell a drive for fewer cents per megabyte and beat competitors in the marketplace. From about 1970 to 1991, areal densities in disk drives were increasing at an annual rate of 30%. In 1991, IBM set a new pace for the hard disk drive (HDD) industry to follow. The annual rate of increase became 60% with the introduction of MR (magnetoresistive) heads and PRML (partial response, maximum likelihood) read channels. This rate, which is commonly known as “Moore's Law,” has been maintained mainly through increases in media and read-channel technologies, with servo technologies barely keeping pace. The result is that increases in areal density, which is the product of the linear bit density and the track density, have been mainly due to increases in linear bit densities. Thus, bits tend to be much shorter than they are wider, with aspect ratios of about 25:1 [7], [11]. Remarkably, areal densities of 20Gb/in^2 have already been demonstrated and drives with 5.7 Gb/in^2 are commercially available.

There are many challenges in increasing areal density. The linear bit density can be increased by making the bits shorter, thus requiring more sensitive read heads, faster write heads, and higher quality read channels to keep the signal to noise ratio at acceptable levels. In fact, with the introduction of MR and GMR (giant MR) read heads, this is how most of the gains in areal density have been achieved in recent years. The track density can be increased by making the bits narrower, but ultimately the highest track density allowable is dictated by the ability of the servo engineer to keep TMR levels acceptably small. A rule of thumb in disk drive design is that the track pitch should be 5-8 times larger than the TMR. Sources of TMR are classified as repeatable runout (RRO) and non-repeatable runout (NRRO). RRO is highly repeatable motion of a track relative to the heads caused by spindle bearing eccentricities and spindle modes. RRO occurs at the harmonics of the spindle motor frequency and is phase-locked to the disk, so special feedforward control is used for RRO cancellation. Usually, RRO compensation is only at the lower harmonics,

since they are much larger and easier to measure. Unfortunately, RRO at the higher harmonics is amplified by the closed-loop sensitivity. Shrinking the track widths means that compensation for the RRO, which is on the order of several track widths, must be even more accurate. Sources of NRRO include the vibrational modes of the disks (called disk flutter), the forces on the slider from the air flow around it (called windage), the inaccuracies in modeling the suspension modes, and the low-frequency stiction effect in the VCM pivot bearing. All of these effects vary from drive to drive and many of these effects, such as RRO, disk flutter, windage, and VCM pivot nonlinearity varies from disk to disk and track to track. Lastly, the PES introduces quantization errors. For more on these tracking error sources, see [8], [9], [1], and [10]. Although the servo controller attempts to use the VCM to compensate for these TMR sources, the VCM has serious limitations. The VCM provides a non-collocated control problem where there are many suspension modes between the head and the arm, has bearings that excite other modes and limit accuracy at low frequencies, and gives a substantial phase loss due to relatively massive size of the VCM. Again, shrinking the track width means that the servo code, which is restricted to a fairly limited number of the available instruction cycles during each servo interrupt, must be able to reduce TMR levels in the face of all of these disturbances.

To make matters worse, there is the superparamagnetic limit, which is currently estimated between 20-40 or 40-100 Gbit/in^2 for the industry standard cobalt alloy on aluminum media [12]. Higher areal densities cannot be achieved with the current technology because the tiny magnetic domains become unstable and merge together. Last but certainly not least, cost efficiency is always a huge challenge. The disk drive is extremely cost competitive and several companies have been forced out of it in recent years. A 6.4 GB drive can be bought for as low as $109, giving a cost per megabyte of only 1.7 ¢/MB [SJ Mercury, Sat. of Memorial Day Weekend?]. It has been noted that the cost of disk storage drops 1% every week.

Since the disk drive industry faces such a huge problem of increasing drive capacity while reducing cost, it is important that the industry considers all of the possible solutions. The purpose of this report is to explore the servo design for a dual-stage disk drive, which is one such solution to the problem of providing higher capacity, cheaper drives. It is possible areal densities beyond the superparamagnetic limit will be achieved through the use of emerging technologies such as optically assisted recording (where domains are heated using a laser) [12], near field recording (e.g., using TeraStor’s special optics) [13], vertical recording (where the domains go into the media instead of along the surface), or patterned media. If this occurs with-

out concurrent improvements in servo technology, then bit aspect ratios will continue to grow. In this case, there is even more to be gained by shrinking track widths. On the other hand, track widths may decrease if improvements in the drive mechanics allow the servo system to work better. For example, disks made of materials other than aluminum may have significantly less flutter and hydrodynamic (fluid) spindle bearings may reduce NRRO due to bearing modes, spindle modes, and flutter modes. It is unlikely that the first torsional mode of the suspension will be pushed out much farther, because of the constraints on the first bending mode, which is at a much lower frequency. It is worth noting that these solutions are all improvements of drive components. To achieve the highest performance possible in a drive, all of the available component technologies would have to be integrated into the same drive.

1.3 The Dual-stage HDD

An alternative solution to the problems posed by increasing the areal density is the dual-stage disk drive. In a dual-stage disk drive, the VCM is the first stage and a smaller, higher-bandwidth microactuator located farther out on the arm assembly is the second stage, as in Figure 1.2. The main purpose of the VCM is then track seeking while the main purpose of the microactuator is tracking. Assuming that a method for surpassing the superparamagnetic limit is made available, the dual-stage HDD can compensate for all of the sources of TMR. By increasing the bandwidth of the drive beyond the first few suspension modes, the drive can reject disturbances over a much wider frequency range. In addition, friction effects in the bearings will not matter because the second-stage handles the fine positioning. Although it may turn out that dual-stage HDDs will be too expensive to be viable, many of the microactuators being developed enjoy the benefits of the semiconductor industry’s batch fabrication techniques. Moreover, since the second stage compensates for all TMR sources, the design constraints on the other mechanical systems in the drive can be relaxed, so cheaper components can be bought. Finally, it is worth noting that dual-stage actuators, although in a somewhat different form, have been standard in CD players for many years. Ultimately, the trade off between cost and performance dictates what technology makes it into a commercial disk drive, but it seems that it is only a matter of time until dual-stage actuators are adopted.

In this report, a methodology for designing dual-stage servo systems is presented. Several case studies are also presented, as well as the software tools used for designing the dual-stage servo systems for the case studies. These case studies were done assuming areal densities of 10 GB/in2, which translates to a track density of 25,000 tracks per inch (tpi) or a track width of 1 μm. According to the projections in [7], the

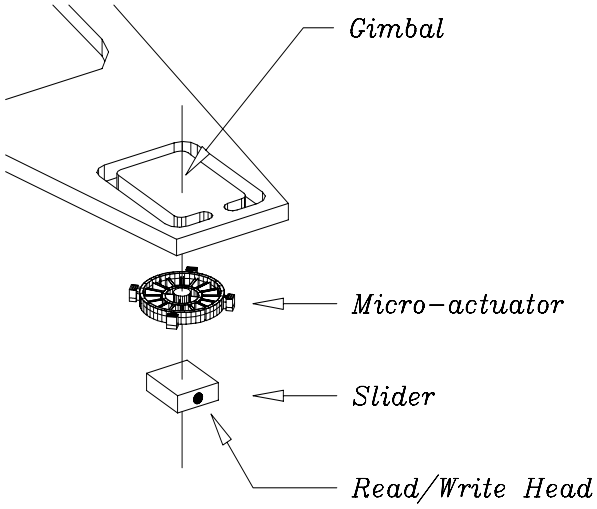


Figure 1.2 *Exploded view of one possible micro-actuator assembly. In this architecture, the micro-actuator is sandwiched between the slider and the gimbal at the end of the arm.*

required servo system needs a bandwidth in excess of 2 kHz with a tracking accuracy of about 100 nm. These are the numbers used as target specifications in this report.

CHAPTER 2

Actuator Modeling

2.1 Original Rotational 3-Mass Model

The coarse actuator in a dual-stage disk drive is the voice coil motor (VCM), which is a rotary actuator (see Figure 1.1). The second stage in a dual-stage disk drive is the microactuator, which does high-bandwidth fine positioning. The original microactuator designed for this project was a rotary polysilicon microactuator (see Figure 2.1) designed by David Horsley and fabricated by Horsley at the U.C. Berkeley Microlab using the HexSil process [14]. Since both the VCM and the microactuator are rotational actuators, the first dual-stage hard disk drive (HDD) model considered was a planar rotational mass-spring-damper model, as shown in Figure 2.2. The model has three rigid bodies, which represent the VCM's arm, the suspension, and the microactuator. This model allows the first important resonant mode of the suspension, which is typically the first roll (torsional) mode at about 2-5 kHz, to be modeled by the spring between the arm and suspension links.

Some of the parameters of the model, such as the suspension length and rotational inertia, were based on data sheets for suspensions and measurements from disk drives. The spring strengths were calculated from assumed values of the resonances. The microactuator parameters were taken from Horsley's design calculations. The linearized model, which is very accurate, shows that the system can be modeled as a standard translational mass-spring-damper system. This is important since it verifies that there is little dynamic coupling from the VCM to the microactuator. This result should not be surprising, because for a rotational microactuator the inertial torques on the microactuator due to the acceleration provided by the VCM cancel each other nearly perfectly. The linearized version of the model was used for the initial controller

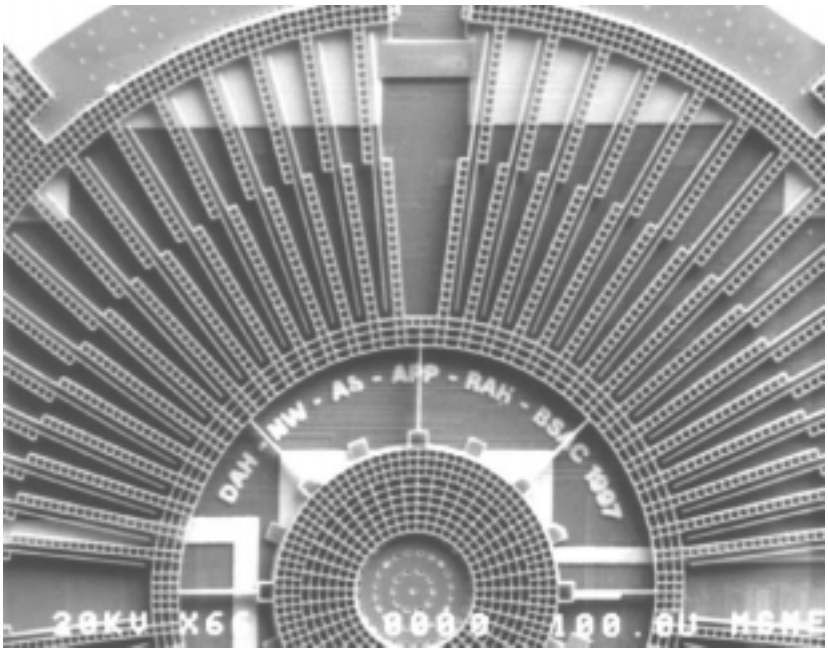


Figure 2.1 *SEM of D. Horsley's 100 μm tall, 2.6 mm diameter rotational microactuator.*

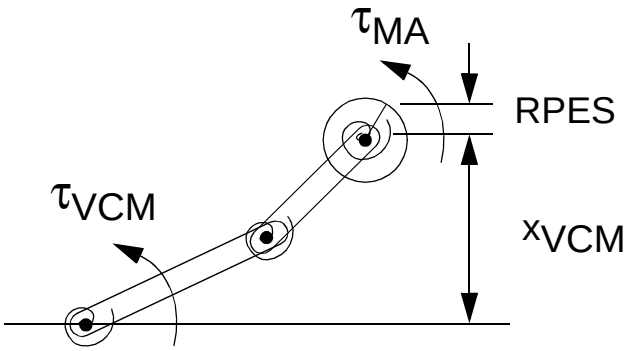
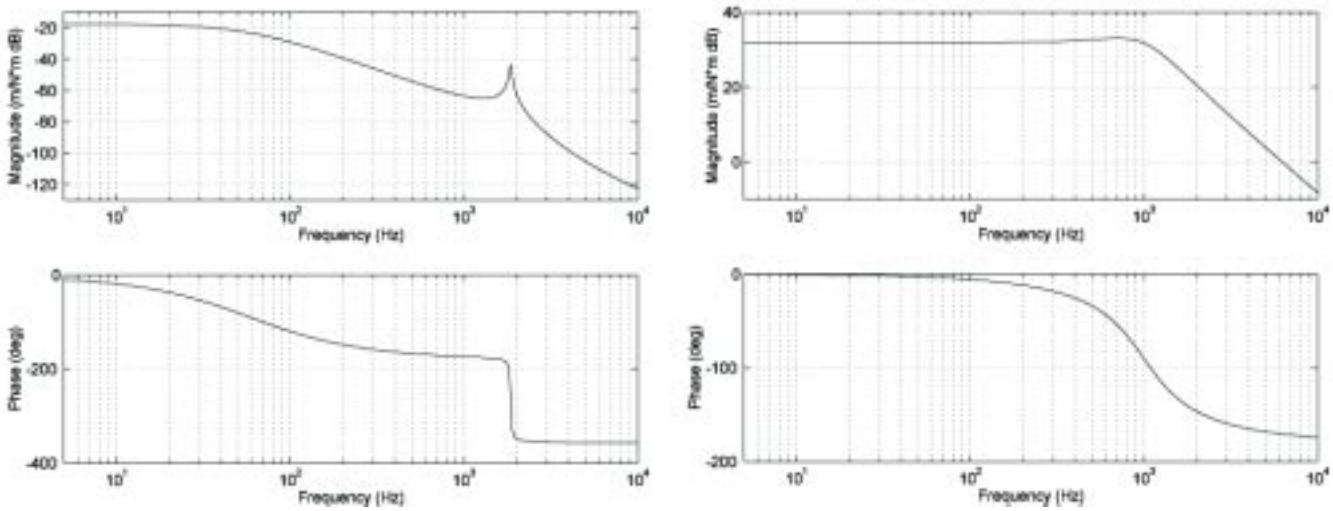


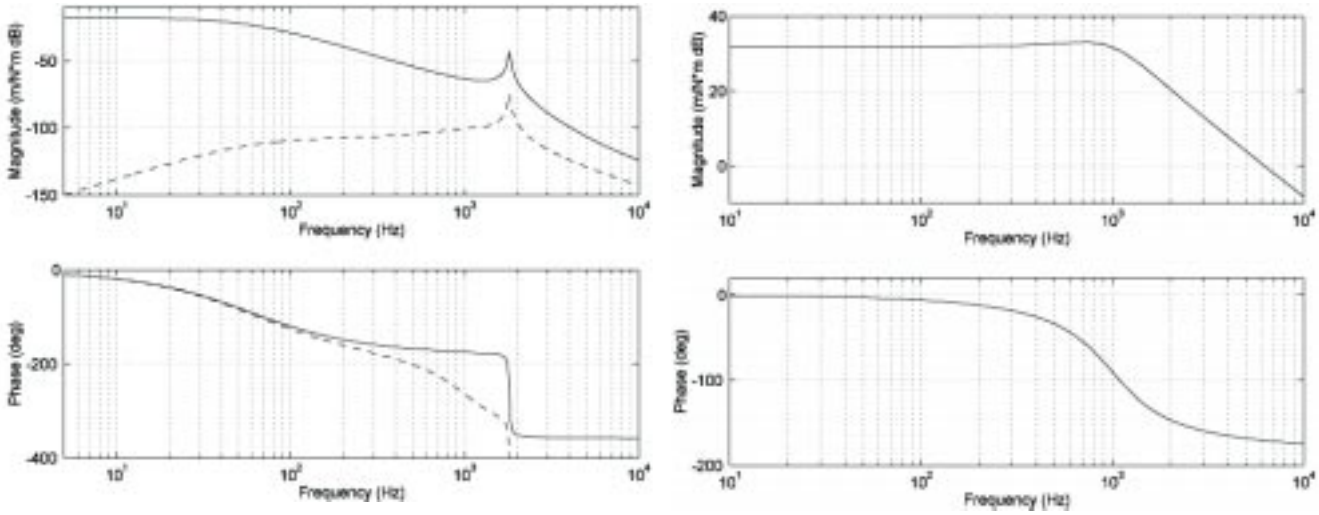
Figure 2.2 *The original three-rigid body rotational mass-spring-damper model used for the initial controller designs and simulations. The inputs are the torques τ_{VCM} and τ_{MA} and the outputs are the position of the tip of the suspension (x_{VCM}) and the displacement of the read/write heads due to the deflection of the microactuator (RPES, the relative PES).*

designs and simulations. The drawbacks of this model are that it may be difficult to get reasonable numbers for the all of the parameters and that, if the designer is interested in truly pushing the performance limits or if several resonances are important, then the model may be a gross over-simplification. The frequency responses of the VCM and microactuator models are shown in Figure 2.3. The frequency responses of the coupled 3-mass dual-stage HDD model using these VCM and microactuator models are shown in Figure 2.4.



(a) *(b)*

Figure 2.3 *The (a) VCM and (b) microactuator models used in the rotational 3-mass model.*



(a) *(b)*

Figure 2.4 *The transfer functions in 2 x 2 transfer function matrix for the coupled 3-mass model. (a)* τ_{VCM} *to* x_{VCM} *(solid) and* τ_{MA} *to* x_{VCM} *(dashed). (b)* τ_{MA} *to RPES (solid) and* τ_{VCM} *to RPES (dashed).*

2.2 Translational 3-Mass Model

The translational three-mass model is very similar to the rotational three-mass model except that a translational microactuator is assumed. Instead of having the microactuator rotate at the tip of the suspension link, the motion of the microactuator mass is restricted to translation in the direction perpendicular to the suspension tip, as shown in Figure 2.6. This model was developed after Horsley designed and fabricated a working translational microactuator (see Figure 2.5), since there were difficulties assembling the rotational

microactuator after release because there were two separate stators. The frequency response of the translational microactuator model is shown in Figure 2.7. The frequency responses of the coupled 3-mass mixed translational/rotational dual-stage HDD model using the translational microactuator model and the same VCM model as before are shown in Figure 2.8. The model indicates that there should be significant dynamic coupling from the VCM to the microactuator. In terms of the frequency response of the system, there is an anti-resonance between the suspension and microactuator resonance and the resonances can be larger since they can excite each other. This kind of behavior is not surprising, since it is the expected response of a flexible arm where the modes are not widely separated in frequency.

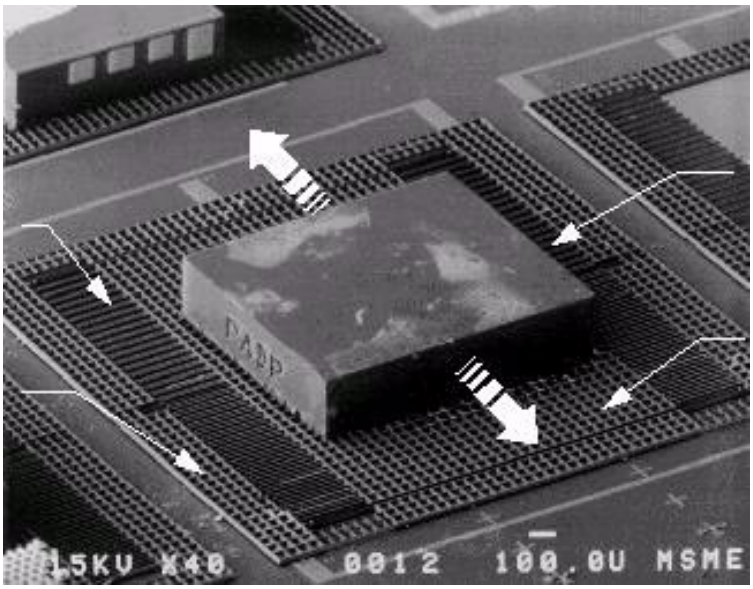


Figure 2.5 *SEM of a slider mounted on D. Horsley's translational microactuator. The large arrows indicate the directions in which the microactuator moves the slider.*

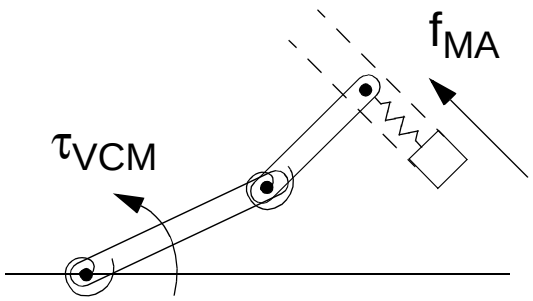


Figure 2.6 *The three-rigid body mixed rotational/translational mass-spring-damper model used after the development of D. Horsley's translational microactuator. Note that the motion of the microactuator is constrained to directions perpendicular to the second (suspension) link.*

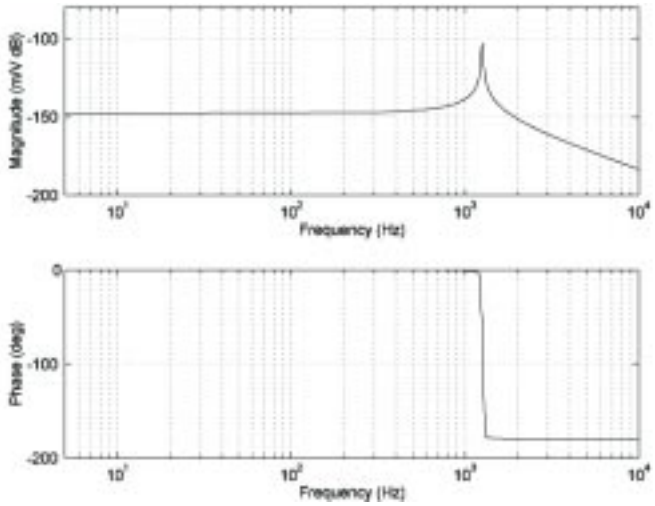
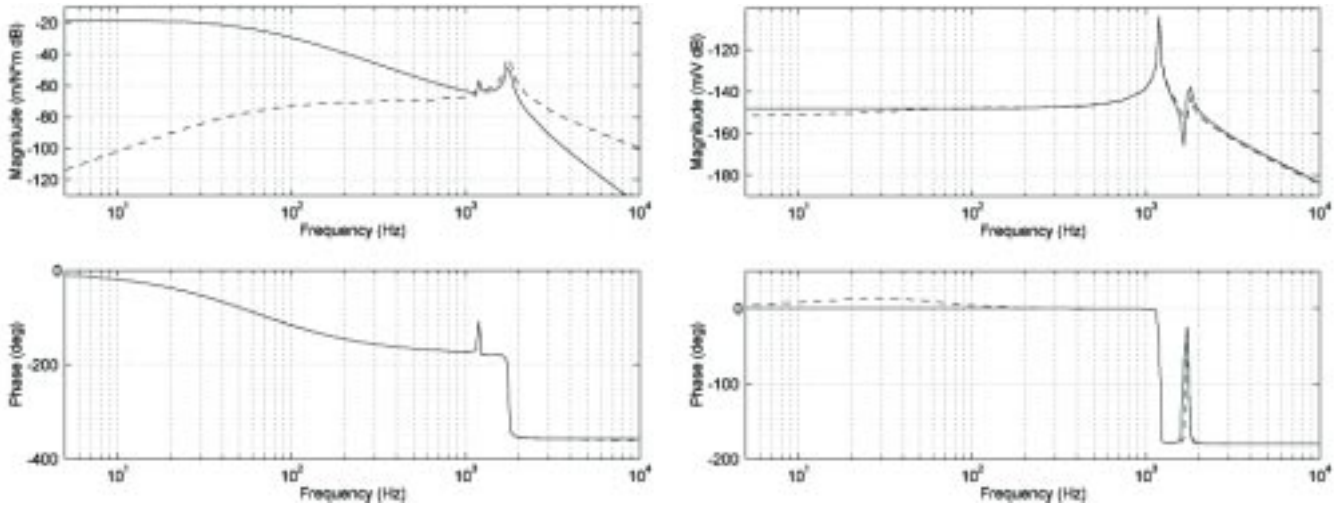


Figure 2.7 *Frequency response of the translational microactuator model used in the 3-mass mixed rotational/translational model.*



(a) *(b)*

Figure 2.8 *The transfer functions in 2 x 2 transfer function matrix for the coupled 3-mass mixed translational/rotational model. (a)* τ_{VCM} *to* x_{VCM} *(solid) and* f_{MA} *to* x_{VCM} *(dashed). (b)* f_{MA} *to RPES (solid) and* τ_{VCM} *to RPES (dashed).*

2.3 The Hutchinson Model

Hutchinson Technology Inc. (HTI), through the National Storage Industry Consortium (NSIC, http://www.nsic.org), provided CML with a model of their Magnum 5 active suspension to test what simulated results CML could achieve on short notice using the mu-synthesis technique. As shown in Figure 2.9, the Magnum suspension has two piezo-electric films at the base of the suspension. By applying different voltages across the piezo-electric strips one strip will expand more than the other, which results in an angular deflection. The Hutchinson technique has both advantages and disadvantages. First, the control

problem is no longer collocated, as it is for the electrostatic MEMS designs, since control is applied at the base of the suspension, but the PES is read at the tip of the suspension. The suspension has many modes and, in addition, the motion of the read/write heads is also coupled to the motion of the VCM through the modes of the piezo-electric films. Second, no relative position sensing of the microactuator is available for the Magnum suspension. On the other hand, the piezo-actuator has a relatively high resonant frequency at about 5 kHz. As will be seen later, measurements of the microactuator's displacement are less important if the bandwidth of the microactuator is well beyond the desired closed-loop bandwidth of 2 kHz. In this case the microactuator is working mainly in the low-frequency, DC gain section of its frequency response, and is less sensitive to perturbations in the model.

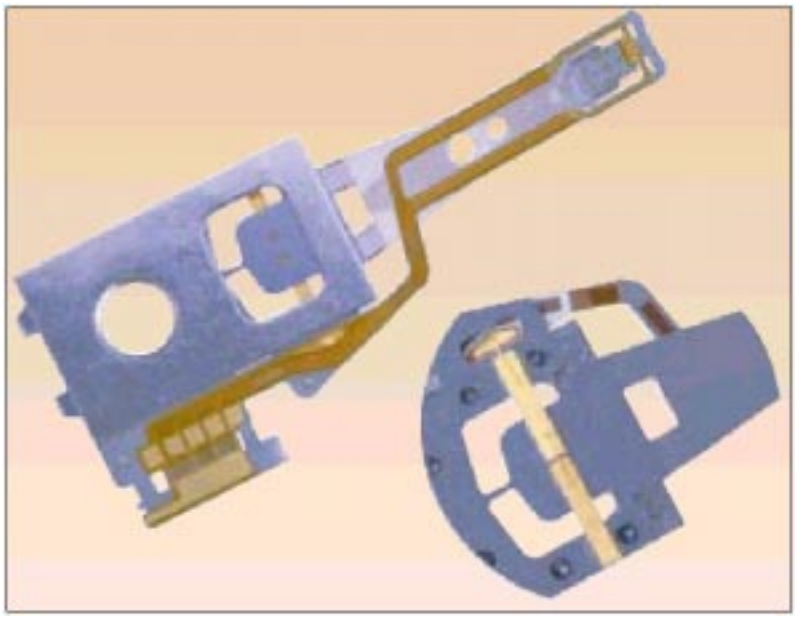


Figure 2.9 *Pictures of Hutchinson Technology Inc.'s active suspension. An integrated suspension is shown at left. At right, a picture of second stage actuator taken from the opposite side reveals the brightly colored piezo elements.*

The block diagram in Figure 2.10 shows the architecture of the HTI model, in which the head displacement due to coupling from the VCM to the active suspension is modeled as a block that filters the head position due to VCM motion. The head displacement due to motion of the active suspension, which, for convenience, will often be referred to as a microactuator in this report, then adds directly to the filtered output of the VCM block. Although this architecture may seem strange, it is what was experimentally determined by HTI. The frequency responses of the blocks in Figure 2.10 are shown in Figure 2.11.

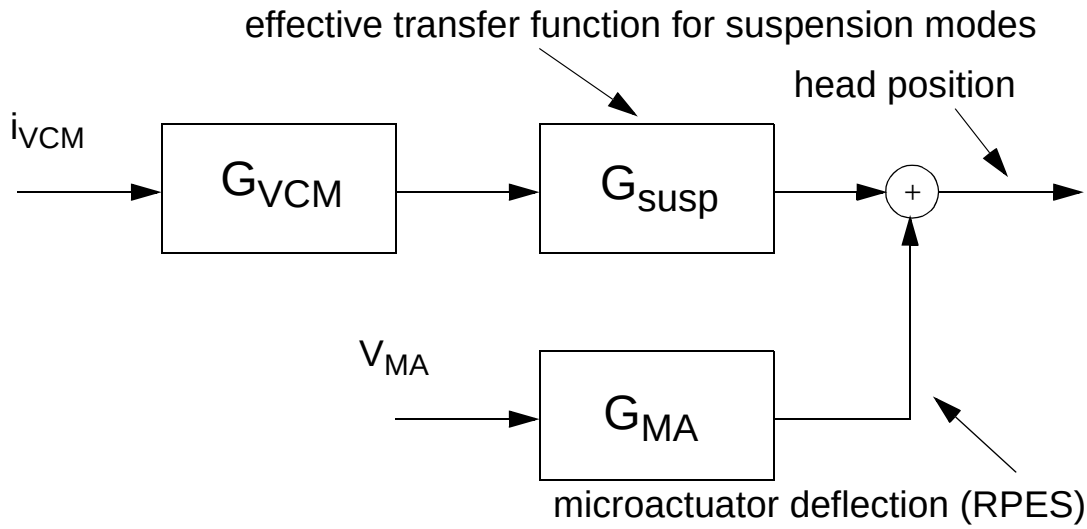
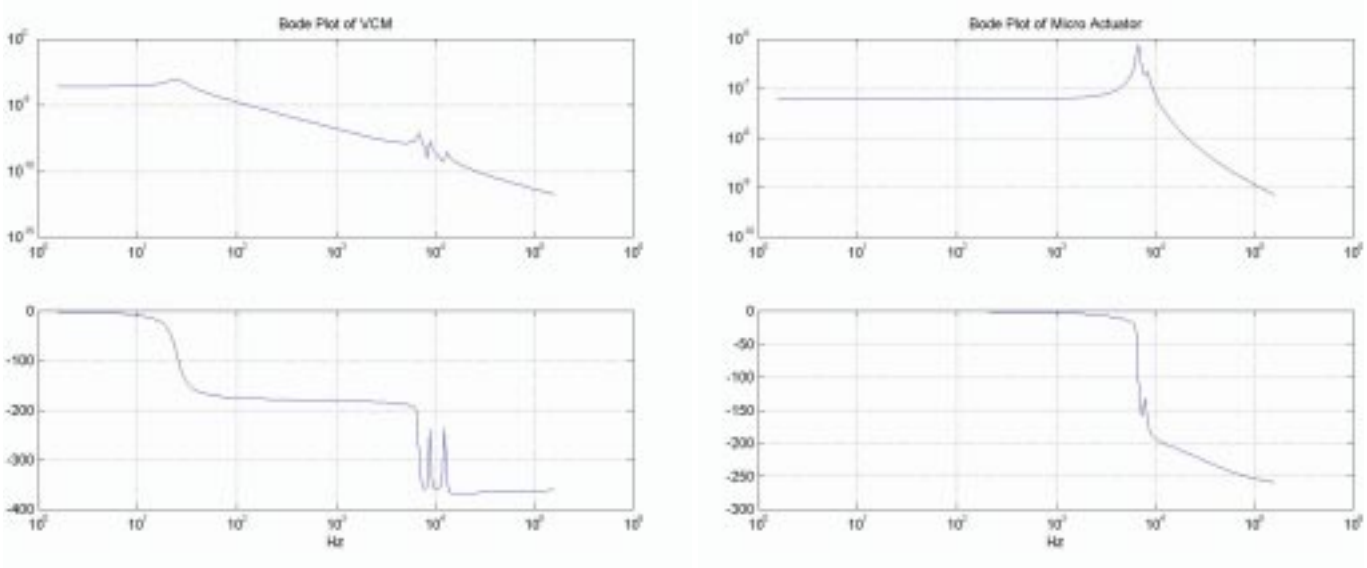


Figure 2.10 *A block diagram showing the architecture of the Hutchinson model. All of the blocks, including G_{susp}, which represents the coupling from the VCM to the active suspension, were determined experimentally by Hutchinson.*



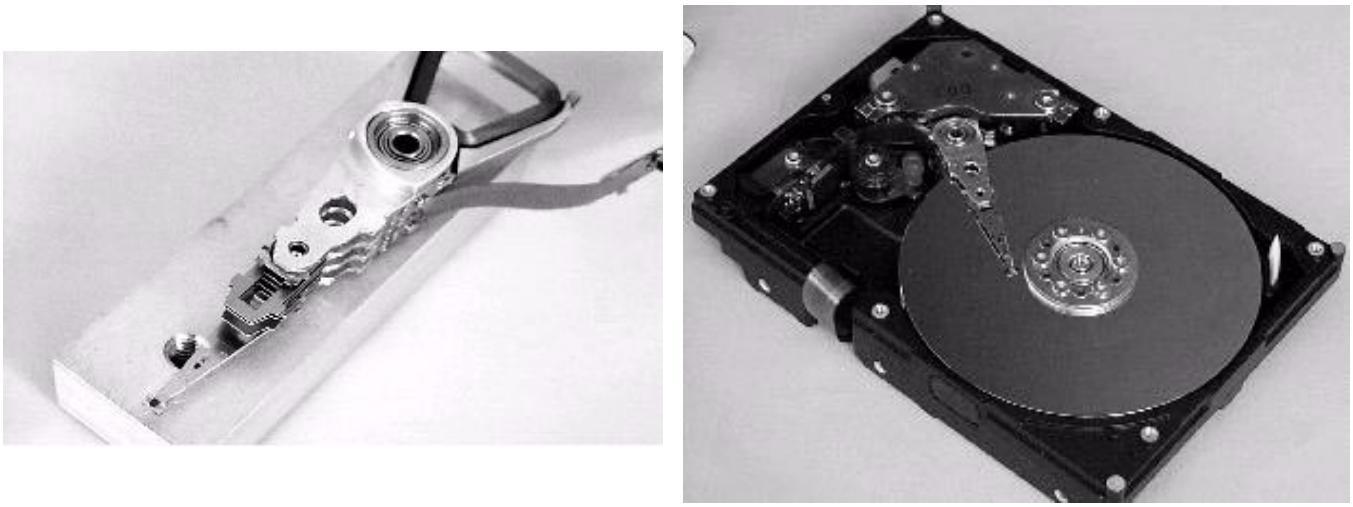
(a) *(b)*

Figure 2.11 *(a) Frequency response of $G_{VCM}\,G_{susp}$ from Figure 2.10. (b) Frequency response of G_{MA} from Figure 2.10.*

2.4 Uncoupled IBM Model

In the uncoupled IBM model, the transfer functions of an IBM Deskstar HDD code-named “Orion” and the IBM area-efficient rotational Invar microactuator (see Figure 2.12) were combined without considering coupling between the systems. Since a model that included any coupling was not provided by IBM, it was decided that the most logical first-cut controller design would use a model without any coupling but

with conservative uncertainties. Hence, in the nominal model, the position of the head is considered to be just the sum of the two separate transfer functions for the position of the microactuator, that is, of the tip of the suspension, and the displacement of the microactuator, which will often be referred to as the relative position error signal (RPES). This architecture is shown in the block diagram in Figure 2.13. The experimental frequency response and the frequency response of the curve-fit to the experimental data are shown in Figure 2.14. The frequency response of the IBM Invar microactuator model is shown in Figure 2.15.



(a) *(b)*

Figure 2.12 *(a) An IBM arm assembly with an integrated microactuator. (b) An IBM DeskStar with an integrated microactuator.*

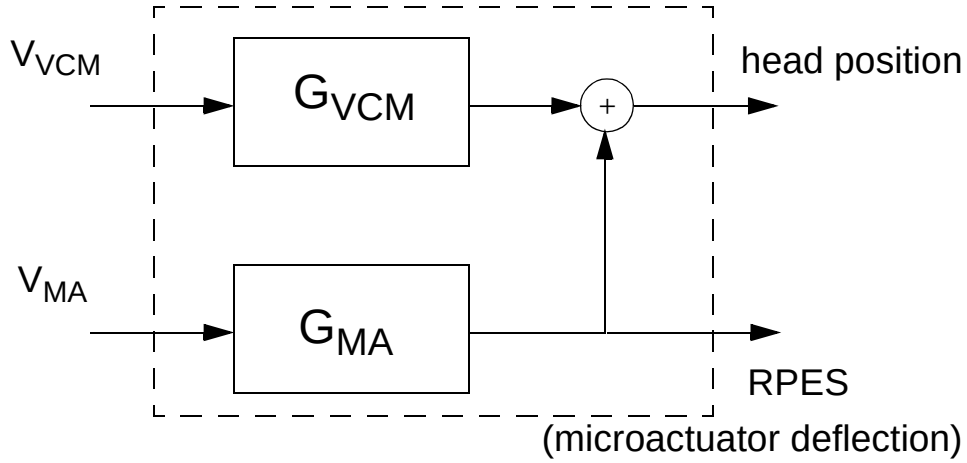
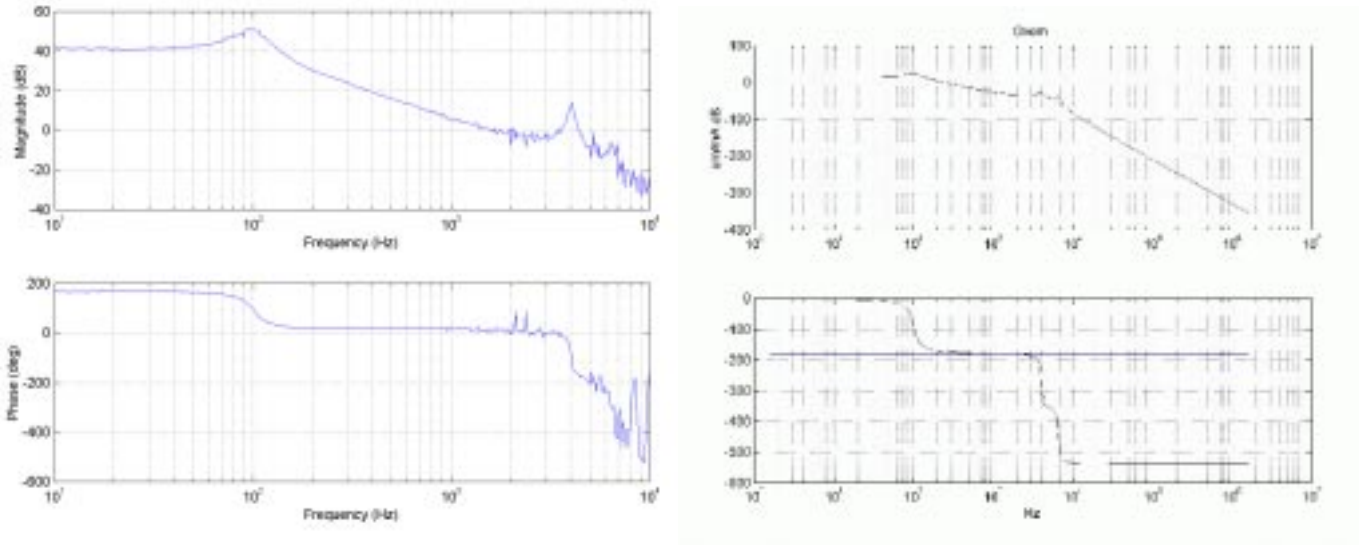


Figure 2.13 *A block diagram showing the architecture of the uncoupled IBM model.*



(a) *(b)*

Figure 2.14 *(a) Experimental frequency response of the IBM VCM model, obtained from an IBM DeskStar drive. (b) Frequency response of the curve-fit to the experimental data.*

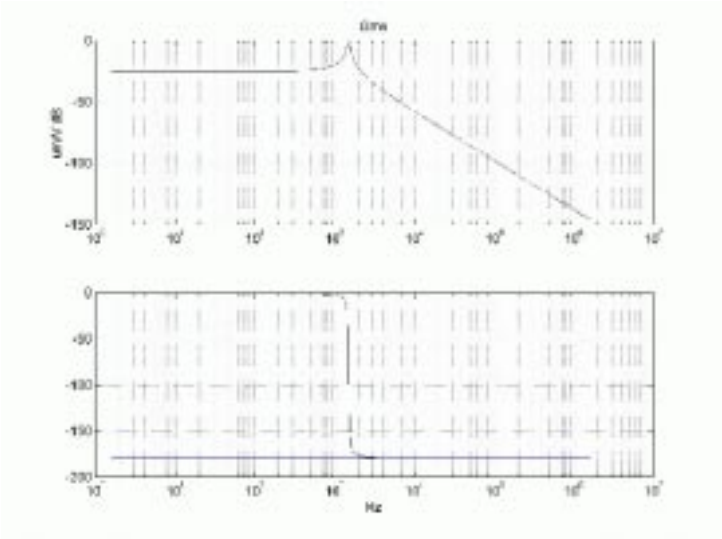


Figure 2.15 *Frequency response of the IBM Invar microactuator model.*

CHAPTER 3

Controller Design

3.1 Why Use Mu-Synthesis?

There is a growing number of MIMO design techniques available to control engineers, however, many techniques are versions of the solutions to the H_2 and H_∞ control problems. To a large extent, deciding what algorithm to use means deciding how to penalize the error signals. For example, if z is the weighted performance errors, then penalizing

$$\int_{t_0}^{T} \|z\| dt, \quad \int_{t_0}^{T} \|z\|^2 dt, \quad \text{or} \quad \sup_{t \in [t_0, T]} \|z\|,$$

that is, the L_1, L_2, or L_∞ norms of z, respectively, will lead to somewhat different controllers. In particular, penalizing the L_2 norm will result in an H_2 optimal controller while penalizing the L_∞ norm will result in an H_∞ optimal controller. Disk drives are stochastic systems and the PES specifications such as the maximum allowable track misregistration (TMR) are typically given as 3σ numbers, so it is not surprising that servo designers are mainly interested in minimum variance techniques. The servo design for a dual-stage disk drive, however, faces some new challenges in terms of uncertain dynamics. There may be significant coupling from the VCM to the microactuator, unmodeled dynamics due to wire bonded signal traces and other artifacts from the assembly process, and variations in the microactuator's resonant frequency. If lithographic misalignments and etching errors cause variations in flexure widths of up to 15% between wafers, then there will be variations in the flexure stiffness of up to 32%, since the flexure stiffness in the compliant direction varies with the square of the flexure width.

There are two possible approaches for handling unmodeled dynamics in the dual-stage disk drive. One approach is to accurately characterize a dual-stage HDD and then design a controller that maximizes performance without allowing for much uncertainty in the model. This method is indeed appropriate for a prototype system than a robust controller design method since it can better assess the ultimate potential of the system. It requires building a dual-stage disk drive and carefully conducting experiments to completely identify the system. However, such a finely-tuned controller could never be used in a commercial drive due to the unit to unit variations resulting from mass production. Another approach is design a controller such that the system will be robust with respect to variations in the model, at a cost of somewhat degraded performance. This is the approach that has been used here. Mu-synthesis has been used to design robust controllers that will perform well for a set of dual-stage disk drives. Using this method, it should be possible to achieve good performance in terms of areal densities and bandwidths by applying the same controller to every dual-stage hard drive that rolls off a production line, despite the variations in the dynamics of the drives. Mu-synthesis, which is essentially an iterative H_∞ controller synthesis, is perhaps the only commercially available means of designing linear time-invariant controllers for uncertain systems by quantifying modeling uncertainties and folding them into the design requirements before the controller synthesis. Other methods, such as H_2, LQG/LTR, and H_∞ only allow the controller designer to analyze the robustness of the design after the controller synthesis. This can result in more design iterations and can make it difficult to change the cost weighting functions to achieve more robustness.

It may also be possible in the near future to do controller designs using a technique similar to mu-synthesis that uses H_2 rather than H_∞ controller synthesis iterations. Mixed H_2/H_∞ control problems and H_2 control with model uncertainty are currently research topics that are receiving much attention.

Also, it should be mentioned that while adaptive control of a dual-stage disk drive may or may not end up being the best solution to the dual-stage disk drive control problem, there are certain obstacles for this technique. Disk drive manufacturers have thus far been unwilling to implement an adaptive controller because of fears that the system will become unstable and also because the computational complexity is usually worse than for a fixed controller. Furthermore, the disk drive industry is highly competitive and the number of months in a drive's design life is continually shrinking. A great deal of testing would be required for adaptive control to earn the industry's trust, so it is not a practical choice for a first-cut design.

Lastly, it should be mentioned that the most promising technique in terms of optimizing performance while tolerating drive to drive variations will probably consist of choosing different fixed controllers for different drives. Provided a MIMO system identification can be done on the drive, a particular controller can be selected that was specifically designed for models close to the model produced by the identification. The model could then be downloaded into memory on the drive's circuit board. Note, however, that this method does not exclude the use of mu-synthesis for designing the individual controllers. Mu-synthesis with small model uncertainties will merely result in a controller that is more limited by performance limitations rather than uncertainties.

3.2 Overview of Mu-Synthesis

This overview is only meant as a brief, qualitative description of mu-synthesis that can help the reader understand the significance of mu and what is going on in the software. For more information on mu-synthesis, consult [15] and [16]. Although they are both are difficult to read and have a large degree of overlap, Zhou's book, *Essentials of Robust Control* [16], is the slightly readable of the two and is recommended. If possible, taking Andy Packard's class on Robust Control, ME234, is also very helpful, if not necessary. The manual for the MathWork's Mu-Analysis and Synthesis Toolbox [17] is very helpful with some of the theorems and concepts and is completely necessary for using the toolbox, which was used extensively for this project. As a side note, Skogestad's book, *Multivariable Feedback Control* [18], is a much easier to read book that is highly recommended for any first-year graduate student studying control theory. Although it does not discuss H_∞ and mu-synthesis in as much detail, it provides a more qualitative overview of these topics, which is what is missing from [15] and [16]. The book also provides a much more complete discussion of multivariable control in general. Lastly, [19] is a good reference for the same reasons and does not have much overlap with [15] and [16].

3.2.1 What is Mu?

Qualitatively, mu, which is the symbol for the structured singular value, is a measure of how big a perturbation to a system needs to be to cause the system to become unstable. Consider the block diagram shown in Figure 3.1, where the transfer function matrix from disturbances, *w*, to weighted performance errors, *z*, can be written as a combination of linear fractional transformations (LFT's) of the plant system interconnection, *P*, the controller, *K*, and some structured uncertainty, Δ. The notation in Figure 3.1 is that y_p is the

output of the system that feeds into the uncertainty block, Δ, and that u_p is the perturbation to the system. In terms of equations, this is written

$$z = F_u(F_l(P,K),\Delta)w = F_l(F_u(P,\Delta),K)w$$

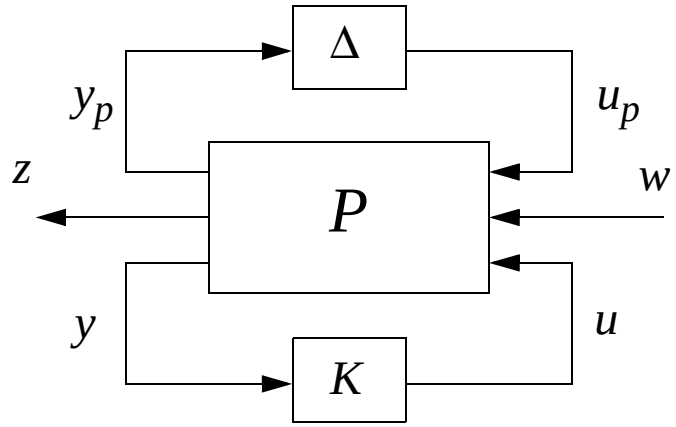


Figure 3.1 *This block diagram appears often in discussions on robust control. The lower LFT shows the feedback connection of the controller while the upper LFT shows uncertainties modeled through a feedback connection. The inputs are the disturbances while the outputs are the weighted error signals.*

where the upper and lower LFT's of a 2 x 2 transfer function matrix *M* with a compatibly-sized transfer function matrix Δ wrapped around it are

$$F_u(M,\Delta) := M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}$$

and

$$F_l(M,\Delta) := M_{11} + M_{12}\Delta(I - M_{22}\Delta)^{-1}M_{21}$$

provided the inverses exist. The Δ here is structured uncertainty; by putting this perturbation matrix Δ in feedback with the plant system interconnection one can create an uncertain plant,

$$\tilde{P} := \{F_u(P,\Delta): \|\Delta\|_\infty<1\} = \{P_{22} + P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12}: \|\Delta\|_\infty<1\}$$

which is really a set of possible plants. Hence P_{22} contains the nominal plant and the rest of P describes how Δ perturbs the plant. Note that the perturbation matrix is always assumed to be normalized to sizes less than one for convenience in the literature and software; frequency weightings that are absorbed into the system interconnection determine both how and to what extent the perturbation modifies the plant. Also, the system interconnection P can always be chosen such that Δ is block diagonal, so

$$\Delta \in \mathbf{\Delta}, \quad \text{where} \quad \mathbf{\Delta} := \{\text{diag}\,[\delta_1 I_{r1},...,\delta_s I_{rs},\Delta_1,...,\Delta_F]:\ |\delta_i(s)| < 1,\ \delta_i(s) \in \text{RH}_\infty,\ \|\Delta_j\|_\infty < 1,\ \Delta_j \in RH_\infty\} \quad ,$$

where RH_∞ is the set of real rational functions bounded everywhere on the imaginary axis and which are analytic in the right half-plane (that is, they have no unstable poles, so they are continuously differentiable in the right half-plane). The fact that Δ is block diagonal is important because it is exploited during the computation of the D-scalings during D-K iteration, which is discussed in the next section. The key point here, however, is that uncertainty that is unstructured at the component level becomes structured uncertainty at the system level, and the mu-synthesis technique exploits this information.

Mathematically speaking, for a complex n x n matrix M, the structured singular value, mu, is defined as

$$\mu_{\mathbf{\Delta}}(M) := \frac{1}{\min\{\bar{\sigma}(\Delta) : \Delta \in \mathbf{\Delta},\ \det(I - M\Delta) = 0\}} \quad ,$$

unless no Δ makes I - MΔ singular,7 in which case μ_{Δ}(M) := 0. If, however, $M := F_l(P,K)$ is the nominal closed-loop system, that is, the bottom loop in Figure 3.1, and M and Δ are compatibly-sized transfer function matrices, then a more specific interpretation may be taken. In this case, mu is the reciprocal of the smallest size perturbation matrix, Δ, for which the loop equations of the simple closed-loop system in Figure 3.2 become singular. Note that when the loop equations for this uncertain closed-loop system, in which the uncertainty is modeled by placing the perturbation block in feedback with the nominal system, become singular stability has been lost.

Naturally, the smaller mu is, the better, since a small mu means that a large perturbation is required to make the system unstable. Typically, mu is required to be less than one because the signals are normalized

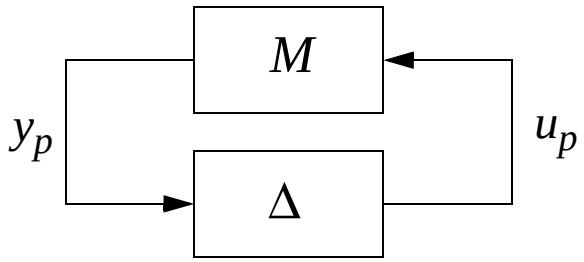


Figure 3.2 *The simple block diagram illustrating the concept behind the definition of the structured singular value. M is the closed-loop system and Δ is the uncertainty. If MΔ becomes of size 1, then I - MΔ becomes singular, that is, the system becomes unstable. In essence, if the loop gain is less than 1, the system must be stable.*

for convenience. The reader may notice that the dimensions of the signals entering and leaving *M* in Figure 3.2 are larger than those of the signals entering and leaving Δ in Figure 3.1, since they include the disturbance vector *w* and the performance error vector *z*. This is because the Δ in Figure 3.2 actually includes a full block on the diagonal that relates *w* to *z*. The explanation for this is that this mu-test is actually a test for both robust stability and robust performance. If P did not have the *w* and *z* as an input-output pair, then the test would only be a robust stability test.

The following discussion explains how robust performance can be tested for using a mu-test, and how Δ should be structured for the test. If $G_p(s)$ is a stable, real-rational, proper transfer function matrix (in practice G_p is M with frequency weightings on *w* and *z* as well as in the uncertainty loop), Δ_P is defined by diagonally augmenting the plant perturbation Δ with Δ_f (which maps *z* to *w*, as in Figure 3.3), and $\|\Delta\|_\infty < 1$, then the loop shown below is well-posed, internally stable, and $\|F_u(G_p,\Delta)\|_\infty \leq 1$ (which means that the performance requirements, which are specified in terms of the magnitudes of frequency weightings, are satisfied) if and only if

$$\sup_{\omega \in R} \mu_{\Delta_p}(G_p(j\omega)) \leq 1 \quad .$$

This idea, which says that a robust performance test is equivalent to a robust stability test with an augmented uncertainty, is stated more precisely as Theorem 10.8 in [16]. The proof of the theorem relies heavily upon a very important result called the main loop theorem. The main loop theorem and an example

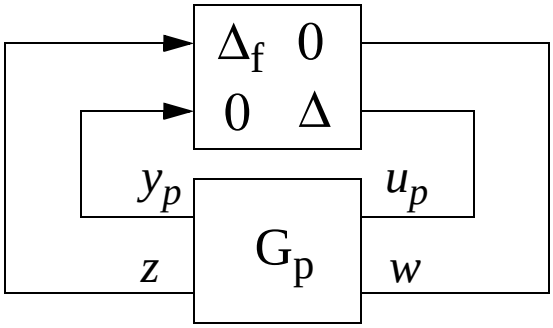


Figure 3.3 *The block diagram for a robust performance test. The Δ block is used to ensure robust stability, while the Δ_f block is used to ensure that the worst case induced 2-norm from w to z is acceptably small (i.e., less than one if the inputs and outputs are normalized).*

showing how it applies to the robust performance of linear systems with uncertainties are presented in [16] on pages 197-199.

3.2.2 D-K Iteration

The only accepted method of mu-synthesis, that is, designing a controller that attempts to minimize the structured singular value of a closed-loop system, is D-K iteration. There are two steps during a single D-K iteration: the D-step, which consists of choosing scaling matrices that decrease mu, and the K-step, which consists of solving an optimal H-infinity control problem. Mathematically, D-K iteration attempts to solve

$$\min_{K \in \boldsymbol{K}_{stab}} \inf_{D,\, D^{-1} \in H_\infty} \left\| DF_l(G,K)D^{-1} \right\|$$

by iteratively searching for K over the set of stabilizing controllers and searching for D over the set of stable, minimum-phase transfer functions.

The objective of the D-scales is to minimize mu. As it turns out, it is usually very difficult to calculate mu but it is much easier to calculate upper and lower bounds on mu. Hence, the effectiveness of a controller will be determined by how small it makes the upper bound on mu. From the definition of mu, it can be shown that $\rho(M) \le \mu_\Delta(M) \le \sigma(M)$, however, it possible to tighten these bounds by multiplying M by transformations that do not affect ρ or σ, but which do affect mu. More specifically, if

$$\boldsymbol{U} = \{ U \in \Delta : UU^* = I_n \}$$

and

$$\mathbf{D} = \left\{ \begin{array}{l} \text{diag}\,[D_1,...,D_S,d_1 I_{m_1},...,d_{F-1} I_{m_{F-1}} 1, I_{m_F}] : \\ \\ D_i \in C^{r_i \times r_i},\ D_i = D_i^* > 0,\ d_j \in R,\ d_j > 0 \end{array} \right\}$$

(so that D and Δ commute, i.e., so that DΔ = ΔD), then tighter bounds may be computed from

$$\max_{U \in \boldsymbol{U}} \rho(UM) = \mu_\Delta(M) \leq \inf_{D \in \boldsymbol{D}} \bar{\sigma}(DMD^{-1}) \ .$$

Unfortunately, the lower bound can have many local maxima, so in practice only a lower bound can be obtained from a search over ***U***. The search for an upper bound is hence more important. The D-scaling that gives the lowest upper bound is found through a point-wise convex optimization in the frequency domain. So, a D(jω) is calculated at every ω, and this data is curve fit with a stable, minimum-phase D(s). The transfer function matrices D and D^{-1} are then absorbed into the generalized plant for the next controller synthesis.

The K-step consists of solving an H-infinity optimal control problem. The symbol $\gamma := \|T_{zw}\|_\infty$, that is, the infinity norm of the transfer function from the disturbances to the performance errors, is often used in discussing H_∞ control problems. An optimal H_∞ controller achieves the minimum γ, γ_{opt}, whereas a suboptimal H_∞ controller does not. Given a choice of γ, solutions to Riccati equations may be used to compute a controller, provided the solutions exist. However, it is generally not possible to know γ_{opt} in advance, so in the Mu-Tools software a bisection algorithm is used to calculate the H_∞ controller for the lowest value of γ for which a solution is found. Using this new controller, D-scales are then recalculated for the new closed-loop system.

Note that because the D-scalings are absorbed into the generalized plant, and the size of the H_∞ controller equals the size of the generalized plant, the size of the controller can become very large in only a few D-K iterations. It is important to try to keep the size of the controller small by (1) using simple frequency weightings, (2) checking to see which uncertainty structure captures the desired effects of the uncertainty most efficiently, (3) eliminating unnecessary states from the nominal plant if the dynamics are drowned out by uncertainty, (4) not setting the maximum number of allowable states too high (e.g., less than 12) when using the auto-prefit (`apf`) option during the D-step in D-K iteration, and, most importantly, (5) using model reduction techniques (e.g., `hankmr`).

3.3 Weights/System Modeling

This section describes the roles of the frequency weightings used in the controller designs and also describes the models used for the frequency weightings. The first, and most important, frequency weighting is that of the runout filter. The weights that penalize the PES and RPES are also important. The later designs assumed disturbance inputs and weights on the controller outputs whereas the earlier designs did not. All the designs included some sort of uncertainty modeling, whether it was multiplicative, additive, or parametric. A block diagram containing all of the frequency weightings used in the designs is shown in Figure 3.4. A brief summary of the weights can be found in Table I and Table II, which give the purpose of

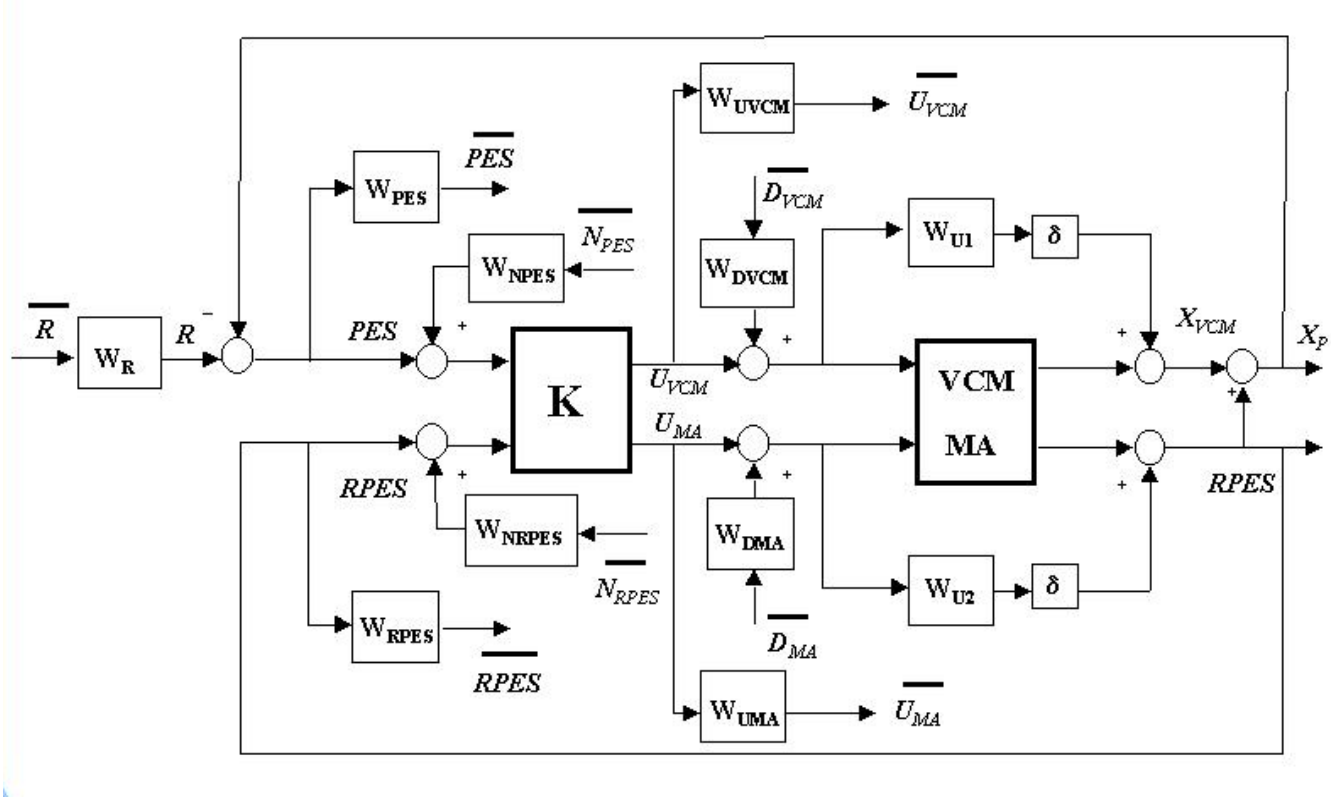


Figure 3.4 *The block diagram showing nearly all of the possible weights considered in the control synthesis method presented in this report. Additive uncertainties are shown, although multiplicative or parametric uncertainties can also be used. A nice feature of the design method is that the weightings are physically meaningful. For example, the inverse of a controller saturation limit can, to a first degree approximation, be used a controller output weighting.*

Table I: Frequency Weights (Rotational and Translational 3-Mass Model Designs)

		Rotational 3-Mass (continuous time)		*Translational 3-Mass (continuous time)*
Weight	*purpose*	*w/RPES*	*w/o RPES*	*w/RPES*
W_R	model runout, control sensitivities	$\dfrac{(20\ \mu m)\left(\frac{35}{800}\right)^2 (s + 2\pi \cdot 800)^2}{(s + 2\pi \cdot 35)^2}$	same	$\dfrac{(20\ \mu m)\left(\frac{52.5}{1200}\right)^2 (s + 2\pi \cdot 1200)^2}{(s + 2\pi \cdot 52.5)^2}$
W_{PES}	limit PES	1/(80 nm)	same	same
W_{RPES}	limit RPES	1/(0.5 μm)	1/(1.4 μm)	1/(1 μm)
W_{UVCM}	prevent VCM saturation	-	-	-
W_{UMA}	prevent MA saturation	-	-	-

Table I: Frequency Weights (Rotational and Translational 3-Mass Model Designs)

		Rotational 3-Mass (continuous time)		*Translational 3-Mass (continuous time)*
Weight	*purpose*	*w/RPES*	*w/o RPES*	*w/RPES*
W_{DVCM}	model VCM disturbances	-	-	-
W_{DMA}	model MA disturbances	-	-	-
W_{NPES}	model PES sensor noise	-	-	-
W_{NRPES}	model RPES sensor noise	10 nm	1 pm	10 nm
W_{U1M}	multiplicative uncertainty on VCM	$\frac{0.05\left(\frac{1500}{75}\right)(s + 2\pi \cdot 75)}{s + 2\pi \cdot 1500}$	same	same
W_{U1A}	additive uncertainty on VCM	-	-	-
W_{U2M}	multiplicative uncertainty on MA	$\frac{0.05\left(\frac{7500}{375}\right)(s + 2\pi \cdot 375)}{s + 2\pi \cdot 7500}$	same	same
W_{U2A}	additive uncertainty on MA	-	-	-

Table II: Frequency Weights (Hutchinson and IBM Model Designs)

	Hutchinson	*IBM*	
Weight	*w/o RPES*	*w/RPES*	*w/o RPES*
W_R	$\frac{(20\ \mu m)\left(\frac{35}{800}\right)^2 (s + 2\pi \cdot 800)^2}{(s + 2\pi \cdot 35)^2}$	see section on runout	see section on runout
W_{PES}	80 nm	100 nm	127 nm
W_{RPES}	2 μm	same	same
W_{UVCM}	1/(2 A)	1/(360 mA)	same
W_{UMA}	1/(100 V)	1/(30 V)	same
W_{DVCM}	0.45 mV	0.058 mA	same
W_{DMA}	23 mV	1.8 V	same
W_{NPES}	10 nm	6.5 nm	same
W_{NRPES}	-	8 nm	same
W_{U1M}	-	-	-
W_{U1A}	$1\times10^{-6}\ \frac{\mu m}{V}$	0.061 μm/mA	$\frac{0.0226s^2 + 260s + 2.28\times10^6}{s^2 + (3.38\times10^4)s + 5.70\times10^8}\ \frac{\mu m}{mA}$
W_{U2M}	$\frac{0.05\left(\frac{10^4}{500}\right)(s + 2\pi \cdot 500)}{s + 2\pi \cdot 10^4}$	-	-
W_{U2A}	-	1.3 nm/V	same

each of the weights and the values used for each of the controller designs. These tables are provided to serve as a quick reference for the contents of this section.

3.3.1 Runout Filter

Before describing the frequency weightings used to model the runout in the controller designs, a brief discussion on runout is in order. Typically, the mechanical runout caused by a spindle shaft is defined as the clearance between the shaft and the bearings that support it. If the shaft is only supported by bearings at one end, as in disk drives, then the runout will be larger at the unsupported end of the shaft, since it is amplified by the length of the shaft and the angle by which the shaft is misaligned at the unsupported end.

As the shaft moves within the bearings and as the shaft bends, the runout changes as a function of time. In disk drives, a slightly different definition is used for the runout. The runout is effectively any variation in the position of the tracks with respect to the read or write head that causes an error signal when the position of the read or write head is fixed. As mentioned in Section 1.2, spindle modes, disk modes, and track eccentricities that are recorded at the time of servo track writing all contribute to the runout. There are two types of runout: repeatable runout (RRO) and nonrepeatable runout (NRRO). RRO dominates the spectrum of the runout and occurs at harmonics of the spindle speed. NRRO, which is due to random motions, is fairly white and serves as the baseline for the spectrum of the runout. For more information on the sources of runout and other PES sources, the reader is referred to [1], [8], and [9]. The are quite sophisticated techniques for dealing with RRO through adaptive feed-forward compensation (see [20], [21], and [22]), and the design of the linear-time invariant (LTI) feedback controller is generally regarded as a separate affair. Hence, in this report, it is assumed that add-on feed-forward RRO compensation schemes are available and RRO compensation is not specifically a goal of the controller design.

The original runout weighting created by Sanjay Aggarwal was a simple, critically-damped second order transfer function with two poles at 35 Hz and two zeros at 600 Hz, as shown in Figure 3.5 [23]. The runout was assumed to be very large at low frequencies to account for the effects of friction in the VCM ball bearings, thus the DC gain is 20 μm. At high frequencies the runout is assumed to be very small, i.e., less that about 40 nm. This is the runout model that was used in the 3-mass rotational design, the 3-mass translational design, and the Hutchinson design.

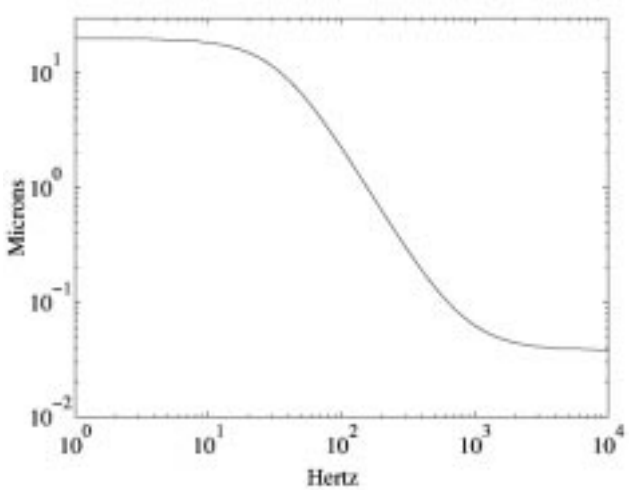


Figure 3.5 *The original runout frequency weighting used in Sanjay Aggarwal's Master's report.*

Besides being a model of the actual runout, the runout weighting is important for another reason. The runout weighting has a large impact on the closed-loop performance of the system. The reason for this is that the inverse of the product of the runout weighting and the performance weighting error gives an upper bound on the sensitivity frequency response (which is also called the error rejection curve).

The runout frequency weighting used in the IBM designs was based on data measured from an IBM DeskStar disk drive that was often referred to by it's internal code name, "Orion." This fifth order frequency weighting is shown with the second order in Figure 3.6 The frequency weighting was constucted

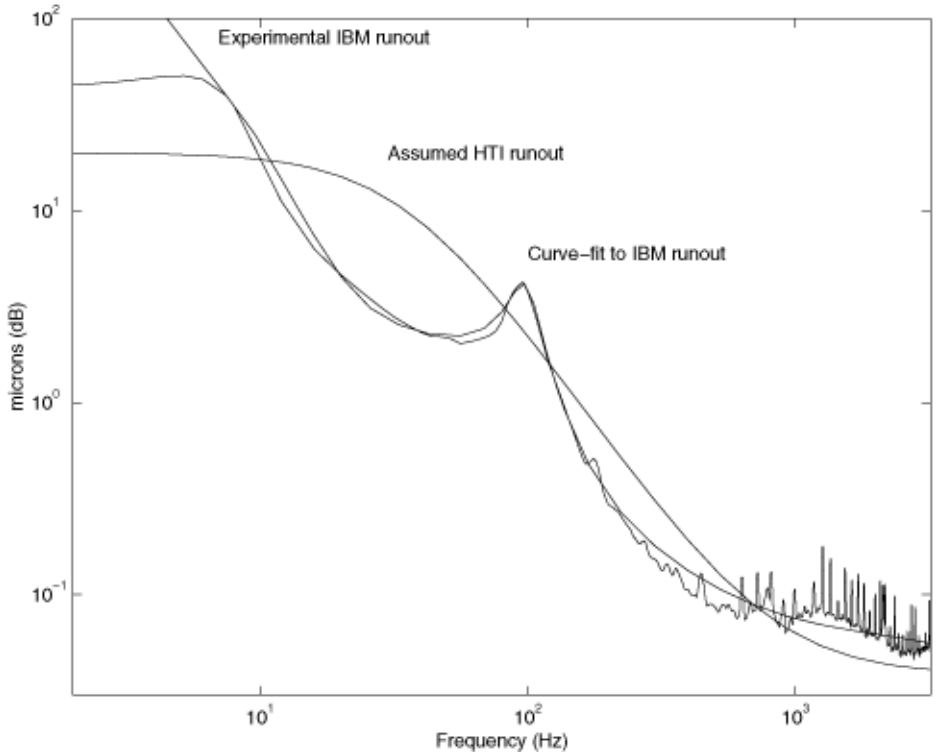
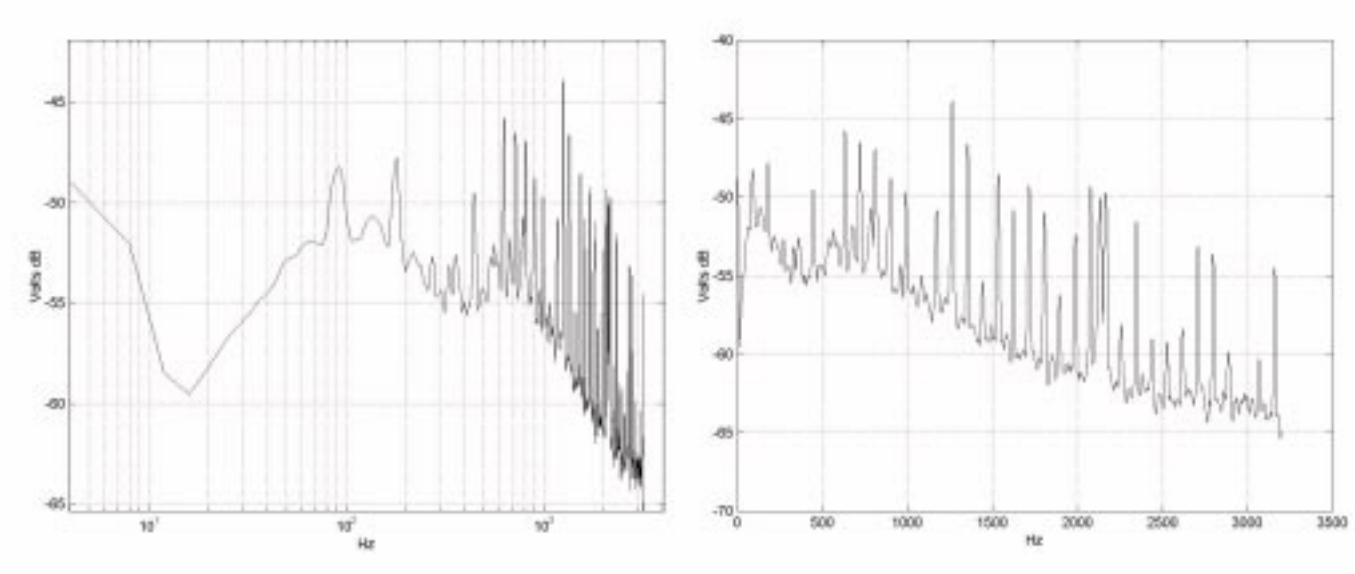


Figure 3.6 *The runout frequency weightings used in the Hutchinson and IBM designs. The weighting used in the Hutchinson design is merely the same weighting as in Figure 3.5. The weighting used in the IBM design was experimentally measured from an IBM DeskStar (internally code named "Orion") disk drive.*

from experimental data, published results, and a little bit of guesswork First, the power spectral density of the PES from the IBM DeskStar drive was measured using a signal analyzer. A monostable multivibrator (a "one-shot") and a TI TMS320C40 DSP evaluation module were used to remove the SAM (servo address marker) from the PES and echo the conditioned signal to the signal analyzer (see [1] for details). This power spectrum is shown in Figure 3.7(a) with a logarithmic frequency axis and in (b) with a linear frequency axis. Although the harmonic distortion is large (there is much more PES at high frequencies

than expected), the spectrum compares well with published results in [8]. Second, the closed-loop transfer



(a) (b)

Figure 3.7 *The power spectral density of the PES measured from the IBM DeskStar drive (plotted versus frequency logarithmically in* ***(a)*** *and linearly in* ***(b)****).*

function from the VCM input to the PES, which is shown in Figure 3.8, was measured using the signal analyzer. Then, the power spectrum of the runout was calculated from a decomposition of the PES power

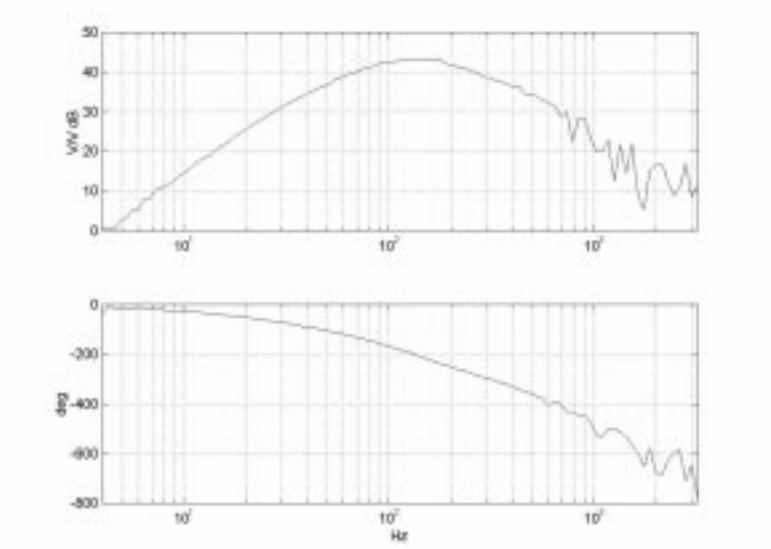


Figure 3.8 *The measured closed-loop transfer function from VCM input to PES (P/(1 + PC), where P is the VCM and C is the controller).*

spectrum,

namely,

$$\Phi_{PES}(\omega) = \left|\frac{1}{1 + P(j\omega)C(j\omega)}\right|^2 \Phi_{nPES}(\omega) + \left|\frac{1}{1 + P(j\omega)C(j\omega)}\right|^2 \Phi_{R}(\omega) + \left|\frac{P(j\omega)}{1 + P(j\omega)C(j\omega)}\right|^2 \Phi_{dVCM}(\omega)$$

where $\Phi_{PES}(\omega)$ is the power spectrum of the PES, $\Phi_{nPES}(\omega)$ is the power spectrum of the sensor noise, $\Phi_{R}(\omega)$ is the power spectrum of the runout, $\Phi_{dVCM}(\omega)$ is the power spectrum of the (input-referred) disturbances to the VCM, $P(j\omega)$ is the frequency response of the VCM, and $C(j\omega)$ is the frequency response of the controller. The values for $\Phi_{nPES}(\omega)$ and $\Phi_{dVCM}(\omega)$ are the same the values for W_{NPES} and W_{DVCM} in Table II. Using those numbers and the frequency response data in the above equation, $\Phi_{R}(\omega)$ was calculated and W_R was then generated by curve-fitting a transfer function to the square root of $\Phi_{R}(\omega)$.

3.3.2 Other Frequency Weightings

Other frequency weightings were chosen based on knowledge of the physical system and on information about disk drives published in Ehrlich's paper on PES decomposition [8]. For example, in the IBM design the microactuator voltage should never exceed 30 V, hence W_{UMA} was chosen as 1/(30 V). Based on estimates in [7], the tracking accuracy for a drive with a track density of 25 ktpi should be about 0.1 μm, hence W_{PES} was chosen as 1/(0.1 μm). The maximum throw of the IBM microactuator is about 2 μm, so W_{RPES} was chosen as 1/(2 μm).

CHAPTER 4

Simulation Results

4.1 Original Rotational 3-Mass Model

4.1.1 Design With RPES Sensing

This section provides an overview of the results produced by Sanjay Aggarwal while he was completing his Master's Degree [23]. Sanjay did a continuous-time controller design for a single-stage disk drive, as well as two controller designs for dual-stage disk drives. One used mu-synthesis to produce two SISO controllers by first designing the VCM controller and then absorbing the VCM controller into the plant for the microactuator controller design. The second method used mu-synthesis to design a MIMO controller in the usual manner. Only the results from the MIMO controller design will be discussed here, since the results of the cascaded SISO approach were inferior to those of the MIMO design, in terms of both performance and efficiency (i.e., the number of states used).

Although the model used for this design is overly simplistic in that the dynamic model is simple, only one disturbance (the runout) is considered, and the uncertainties are rather small, this design is still useful. Also, the controller can not be directly implemented, since it is a continuous-time controller. Nevertheless, the results give an indication of what the upper bound on performance will be for a system such as this. Additionally, the design provides a good starting point, especially since it is generally a better idea to start with a simpler problem and then add complexity than to start with a complex problem. Figure 4.1(a) shows the closed-loop runout to PES sensitivity transfer function. As seen in the figure, the attenuation at low frequencies is very large, so the steady-state error will be good. The bandwidth is also very high, with a -3 dB

cross over of about 2.2 kHz. Similarly, the runout to RPES transfer function, which shows how hard the microactuator is working over various frequency ranges, is presented in Figure 4.1(b). Note that although the transfer function is small at low frequencies, the runout is large at low frequencies so the microactuator is still working somewhat at lower frequencies. If the runout to RPES transfer function were too small, it would be a flaw in the design, since we require reasonable action from the microactuator at low frequencies in order to compensate for VCM pivot friction. The (complementary sensitivity) transfer functions

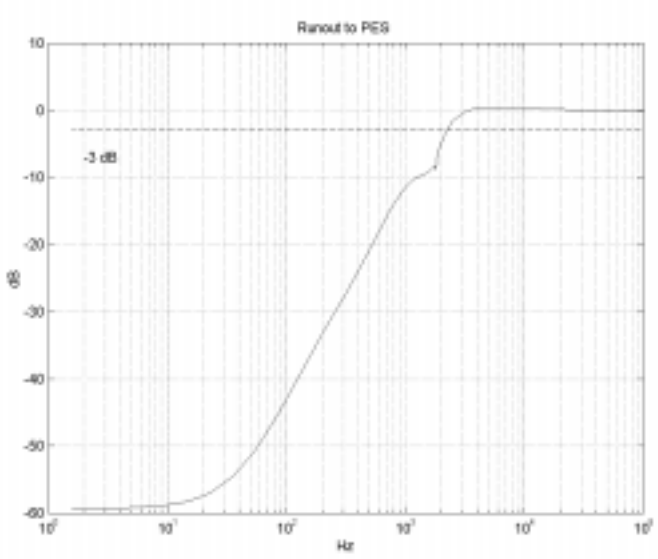


Figure 4.1 (a) *Closed-loop transfer function from runout to PES for the original (continuous-time) 3-mass rotational design with RPES sensing.*

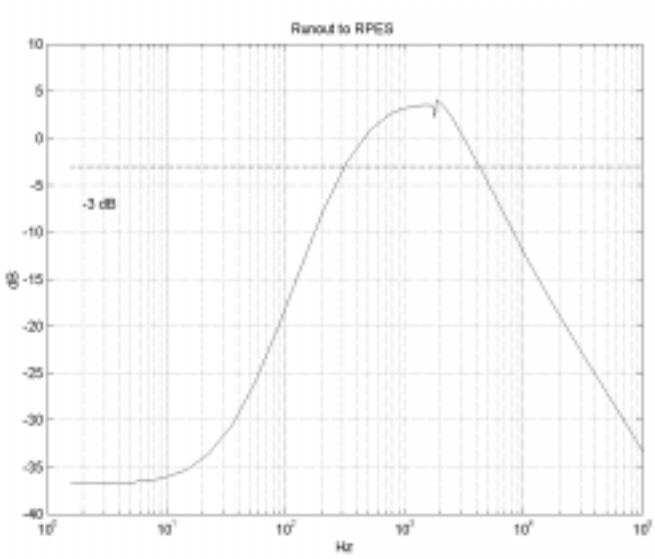


Figure 4.1 (b) *Closed-loop transfer function from runout to RPES for the original (continuous-time) 3-mass rotational design with RPES sensing.*

from runout to head position (x_p) and from runout to the position of the tip of the suspension (x_{VCM}) are shown in Figure 4.2(a) and Figure 4.2(b). Figure 4.2(a) shows that the bandwidth of the dual-stage system is quite high (4.1 kHz if measured as the -3 dB cross over of this complementary sensitivity, which is probably the more standard definition of bandwidth although it is perhaps less meaningful), while Figure 4.2(b) shows that the bandwidth of the VCM is about 960 Hz. The response of the system to a unit step in the runout, which should be equivalent to a one track seek in the absence of disturbances, is shown in Figure 4.3. The response shown very quick and smooth settling to within 10% of a track in about only 0.25 μsec. From Figure 4.2(b) and Figure 4.3 it is apparent that the VCM bandwidth is considerably higher than is desirable.

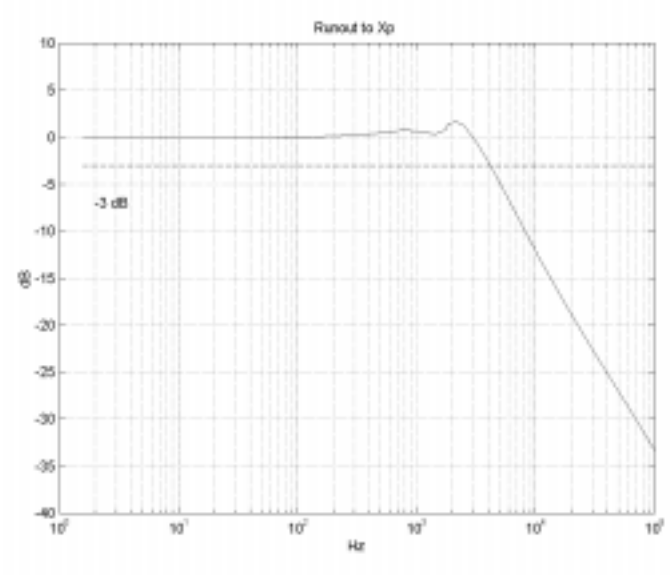


Figure 4.2 (a) *Closed-loop transfer function from runout to head position (x_p) for the 3-mass rotational design with RPES sensing.*

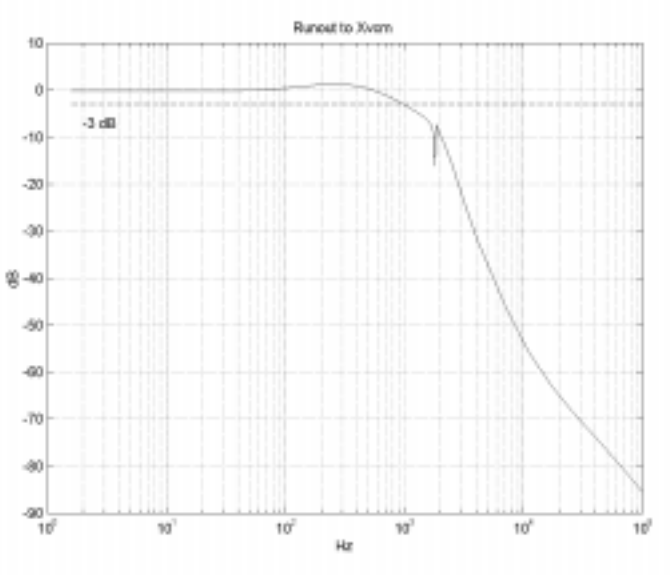


Figure 4.2 (b) *Closed-loop transfer function from runout to the suspension tip position (x_{VCM}) for the 3-mass rotational design with RPES sensing.*

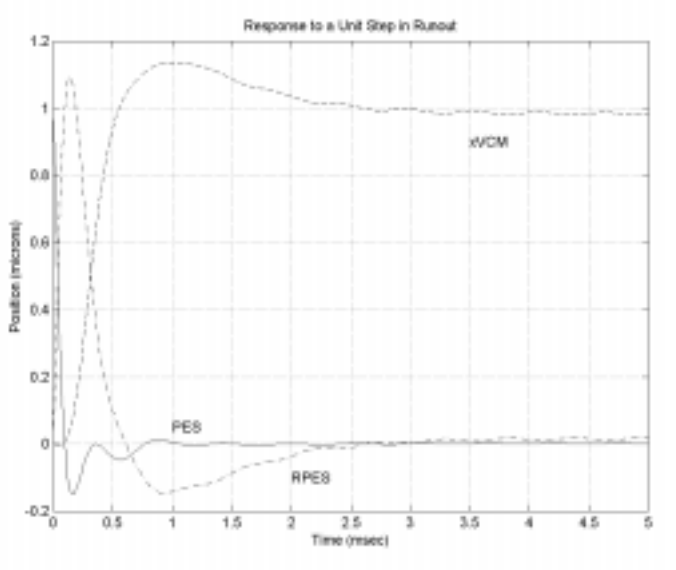


Figure 4.3 *Closed-loop responses of the PES, the suspension tip position (x_{VCM}), and the RPES for the 3-mass rotational design with RPES sensing.*

4.2 Translational 3-Mass Model

4.2.1 Design With RPES Sensing

The design for the mixed rotational/translational model is essentially a repeat of the design for the 3-mass rotational model except a different dual-stage actuator model. The design is again in continuous-time and the same uncertainty and runout weights are used. The main purpose of the design is to test the effects on performance of using a translational microactuator instead of a rotational microactuator. The presentation of this particular case study will also be used to illustrate the effects of model reduction. The 3-mass translational design is a 9^{th} order controller that was obtained by model reducing a 40^{th} order controller. For each frequency response plot, the result for the full order controller is plotted in a dash-dot line with the result for the reduced order controller plotted in a solid line. The transfer functions from runout to PES and from runout to RPES are shown in Figure 4.4(a) and Figure 4.4(b). Immediately one can see that the

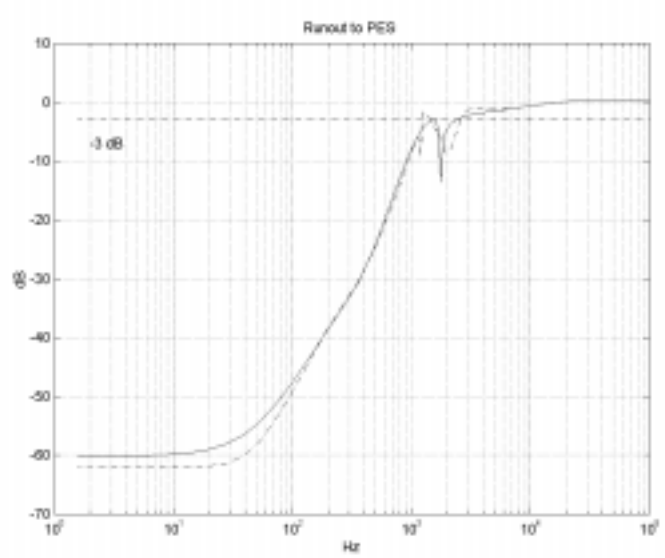


Figure 4.4 (a) *Closed-loop transfer function from runout to PES for the (continuous-time) 3-mass translational design with RPES sensing using the full order (dash-dot) and reduced order (solid) controllers.*

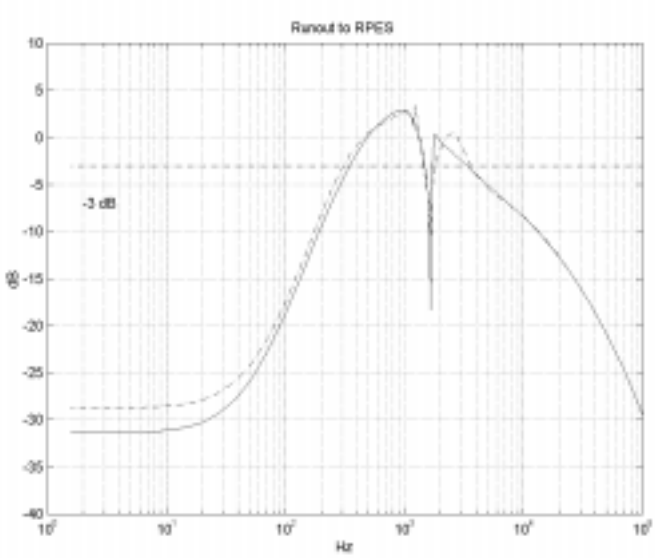


Figure 4.4 (b) *Closed-loop transfer function from runout to RPES for the (continuous-time) 3-mass translational design with RPES sensing using the full order (dash-dot) and reduced order (solid) controllers.*

shapes of the sensitivities are more complex at the resonances, which are located near the cross over frequencies. This is attributed to an increase in the coupling from the VCM to the microactuator. One also notices that the shape of the frequency response has changed dramatically due to model reduction. Specifically, a much sharper notch appears in the sensitivities for the closed-loop system with the model reduced

controller. The runout to head position (x_p) and the runout to suspension tip position (x_{VCM}) are shown in Figure 4.5(a) and Figure 4.5(b). One can see that the bandwidth of the VCM is still quite high and, in fact, the VCM contributes significantly to the compensation for the coupling between the VCM and the micro-actuator at about 1.7 kHz. Generally the frequency responses of the unreduced and reduced models will

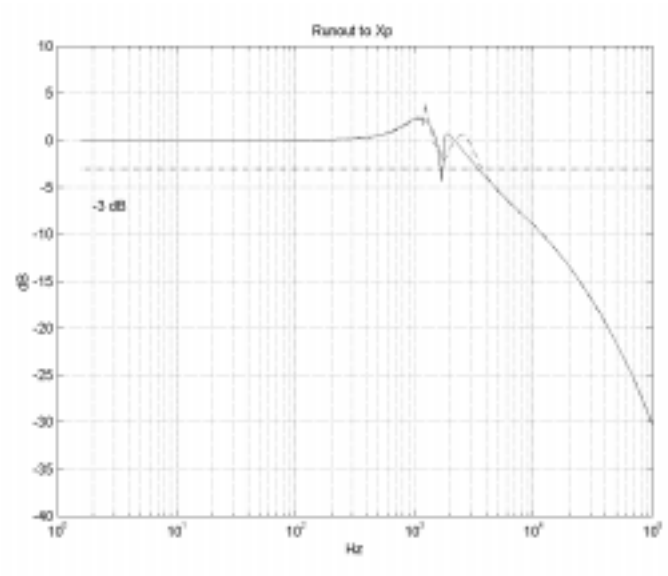


Figure 4.5 (a) *Closed-loop transfer function from runout to head position (x_p) for the 3-mass rotational design with RPES sensing using the full order (dash-dot) and reduced order (solid) controllers.*

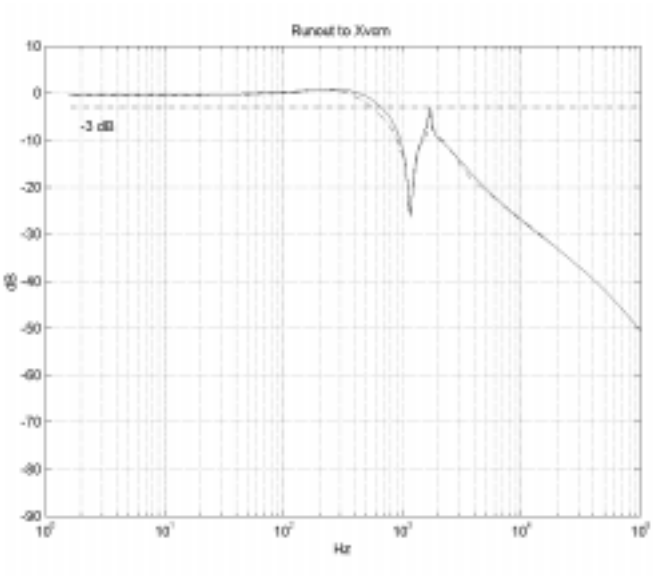


Figure 4.5 (b) *Closed-loop transfer function from runout to the suspension tip position (x_{VCM}) for the 3-mass rotational design with RPES sensing using the full order (dash-dot) and reduced order (solid) controllers.*

match well, until the number of states requested for the reduced order model becomes so small that the success of the model reduction attempts become erratic. Indeed, the 9th order result presented here falls into that category, however, it does yield robust performance when used to close the loop. The mu plots for the uncertain closed-loop systems using the full and reduced order controllers are shown in Figure 4.6. There is degradation of the margin by which robust performance is achieved since the peak value is larger for the system with the reduced order controller, but the results are still acceptable. The plot illustrates a typical strategy: design the system so that the frequency response μ-test is a little conservative so that some degradation due to model reduction can be tolerated. Lastly, the responses of the two closed-loop systems to a unit step in the runout are shown in Figure 4.7(a) and Figure 4.7(b). Again, the dynamics of the two closed-loop systems are remarkably different.

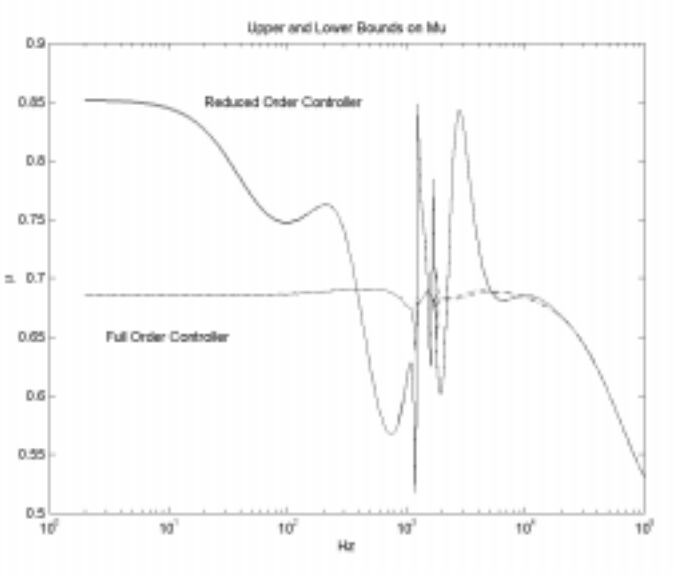


Figure 4.6 *The structured singular value (μ) plots for the uncertain closed-loop systems using the full order (dash-dot) and reduced order (solid) controllers.*

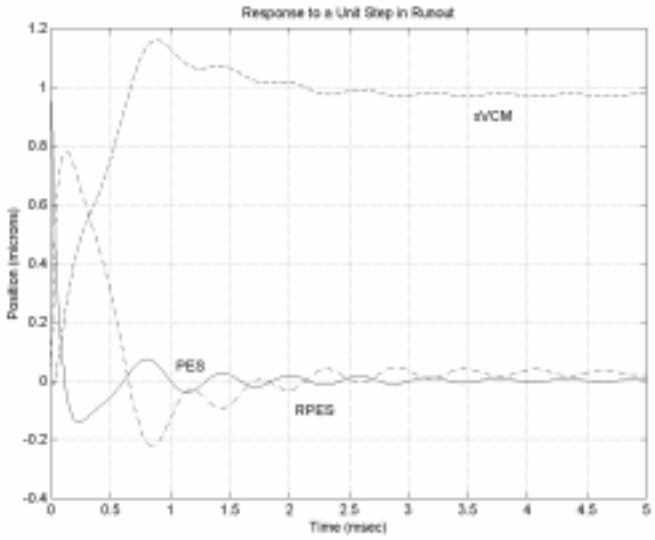


Figure 4.7 (a) *Closed-loop responses of the PES, the suspension tip position (x_{VCM}), and the RPES for the 3-mass rotational design with RPES sensing using the full order controller.*

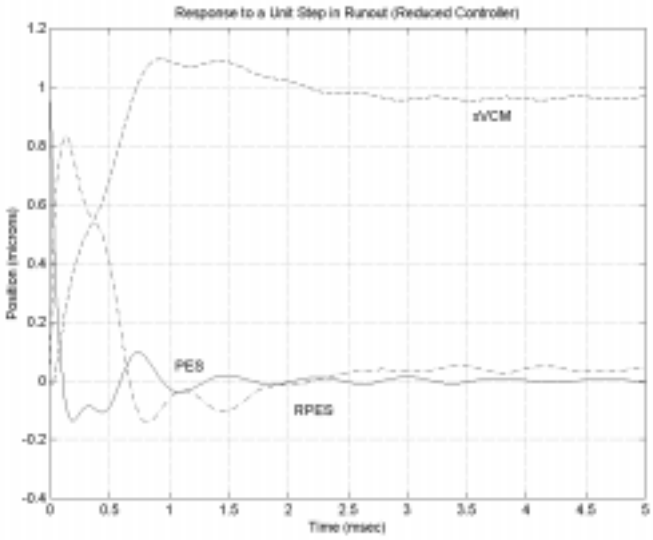


Figure 4.7 (b) *Closed-loop responses of the PES, the suspension tip position (x_{VCM}), and the RPES for the 3-mass rotational design with RPES sensing using the reduced order controller.*

4.3 IBM

4.3.1 Design With RPES Sensing

This section discusses the results from the solution of the mu-synthesis problem posed above when the plant is the 2-input, 2-output system comprised of the IBM VCM model and the IBM microactuator model. The IBM designs were done in discrete-time with an assumed sampling rate of 25 kHz. The results show that the intuitive, physically-based weights of Section 3 lead to a sensible solution that performs robustly. Figure 4.8(a) and Figure 4.1(b) show the closed-loop transfer functions from runout to PES and

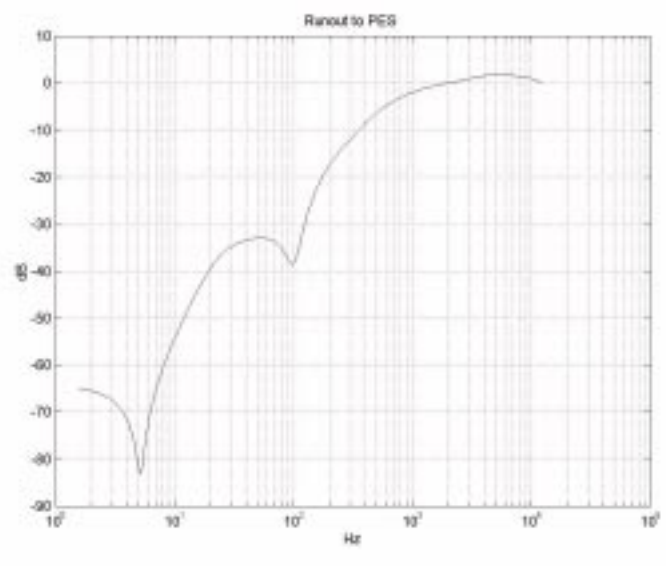


Figure 4.8 (a) *Closed-loop transfer function from runout to PES for the IBM design with RPES sensing.*

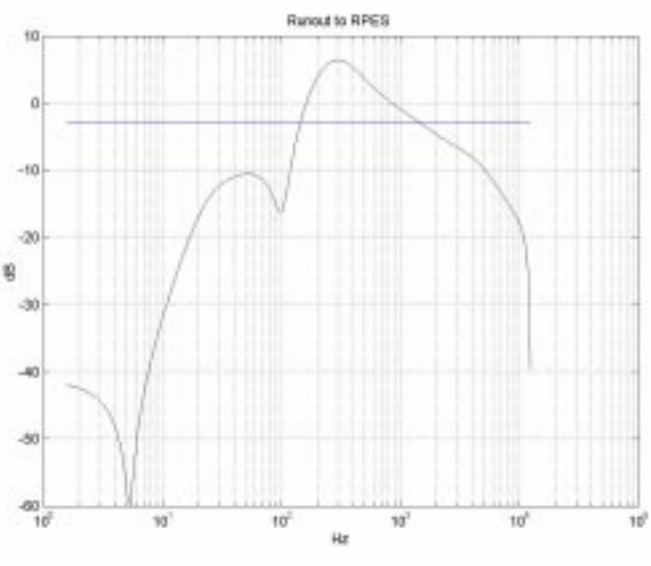


Figure 4.1 (b) *Closed-loop transfer function from runout to RPES for the IBM design with RPES sensing.*

RPES, respectively. Note that the low frequency runout rejection is very good in both the PES and the RPES, as is demanded by the upper bounds placed on the sensitivity transfer functions by the W_R, W_{PES} and W_{RPES} weights. The microactuator has excellent steady-state tracking and the tracking of the VCM is good enough to keep the microactuator centered very well. As indicated by Figure 4.1(b), the microactuator is relatively unexcited by the spectral content in the runout below 50 Hz. The microactuator becomes more active in the 50-100 Hz range, which should improve low frequency tracking significantly, since the VCM bearings have significant non-linear friction behavior in this frequency range. From the runout to head position transfer function in Figure 4.9(a), the -3 dB closed-loop bandwidth of the system is 1.2 kHz,

while from Figure 4.2(b) the -3 dB closed-loop bandwidth of the VCM is only 400 Hz. Note also that the

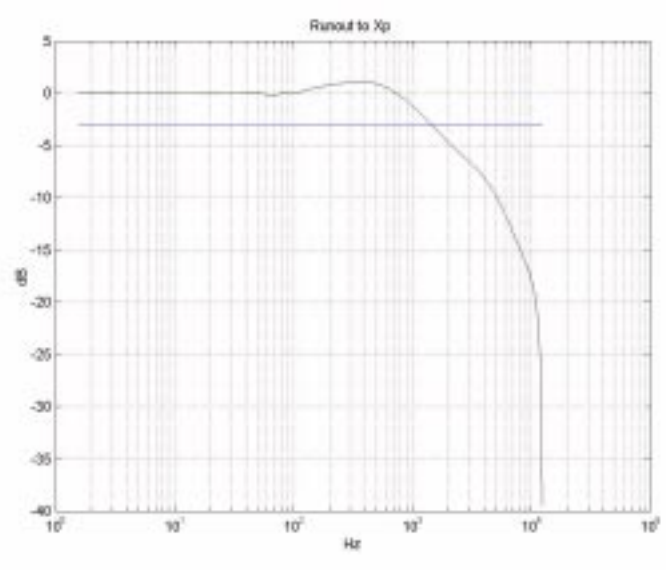


Figure 4.9 (a) *Closed-loop transfer function from runout to head position (x_p) for the IBM design with RPES sensing.*

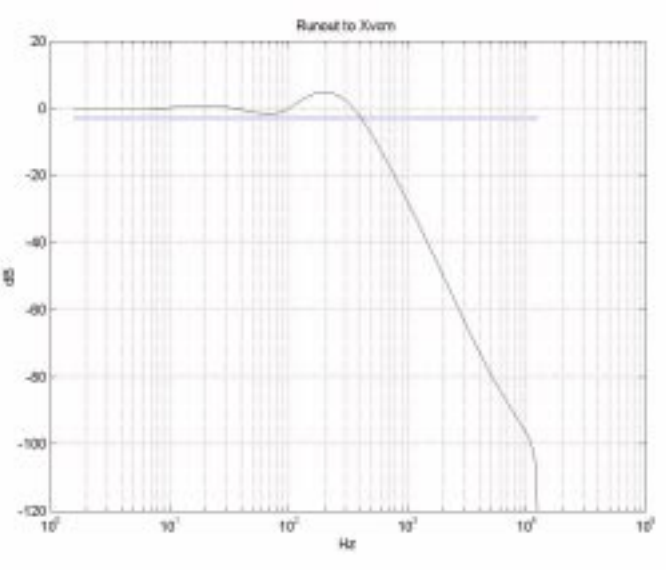


Figure 4.2 (b) *Closed-loop transfer function from runout to the suspension tip position (x_{VCM}) for the IBM design with RPES sensing.*

RPES is actually larger than the runout in the range of 120-830 Hz, since it must be able to track the VCM's errors as well as the runout. The step response in Figure 4.10 illustrates this behavior.

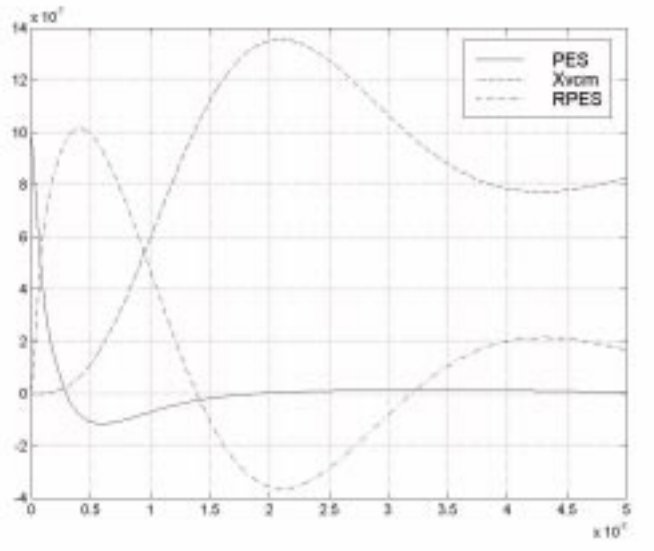


Figure 4.10 *Closed-loop responses of the PES, the suspension tip position (x_{VCM}), and the RPES for the IBM design with RPES sensing.*

4.3.2 Design Without RPES Sensing

Naturally, a good design without RPES sensing should recover the performance and robustness of the design with RPES sensing. Intuitively, the degree of recovery depends upon the level of importance of the RPES signal. Indeed, this is the observed trend. Performance is degraded in order to achieve the same level of robustness. As will be seen later, however, the results can be analyzed to inspire efficient SISO dual-stage controller design. The closed-loop bandwidth of the design without RPES sensing is close to that of the design with RPES sensing (see Figure 4.11(a)), however, the disturbance rejection is suffers at high frequency. It is quite clear from the figure that the very uncertain microactuator resonance is an uncontrolled mode that is notched out. The behavior is illustrated well by the step response in Figure 4.13. If the uncertainty in the microactuator is reduced, the performance of design with RPES sensing can of course be recovered.

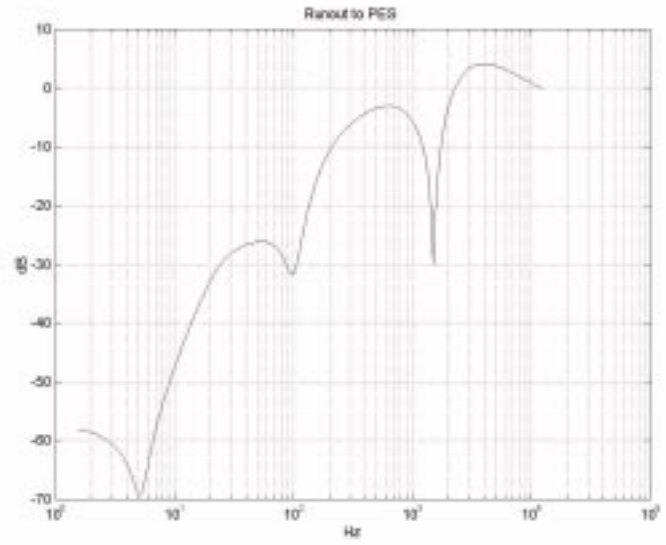


Figure 4.11 (a) *Closed-loop transfer function from runout to PES for the IBM design without RPES sensing.*

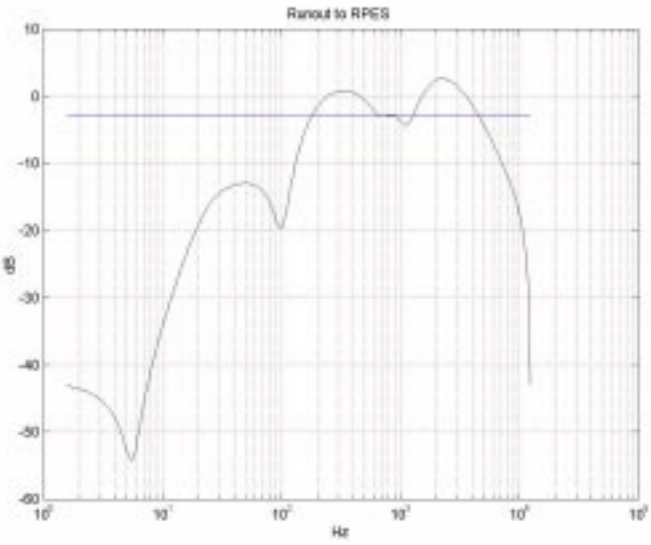


Figure 4.4 (b) *Closed-loop transfer function from runout to RPES for the IBM design without RPES sensing.*

Some interesting observations may be made about the results of the design without RPES sensing. Ignoring disturbances other than the runout and assuming a fairly uncoupled system, the block diagram for the system using the SIMO controller can be drawn as in Figure 4.14(a), which can then be manipulated into Figure 4.14(b). In the diagram there is a new, fictitious signal labeled PES_{VCM}, which is what the PES would be if it were measured from the track-centerline to the microactuator pivot (i.e., the PES plus the

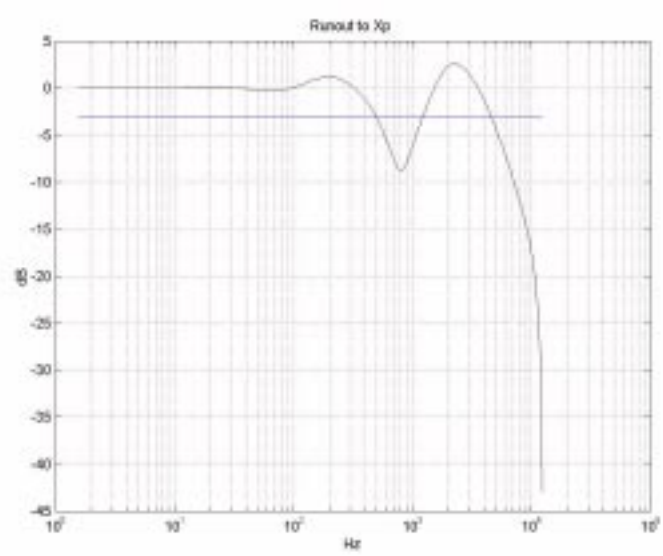


Figure 4.12 (a) *Closed-loop transfer function from runout to head position (x_p) for the IBM design without RPES sensing.*

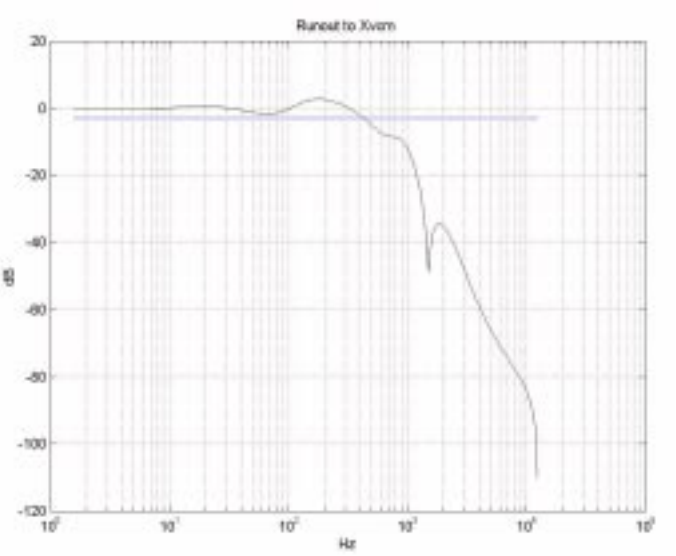


Figure 4.5 (b) *Closed-loop transfer function from runout to the suspension tip position (x_{VCM}) for the IBM design without RPES sensing.*

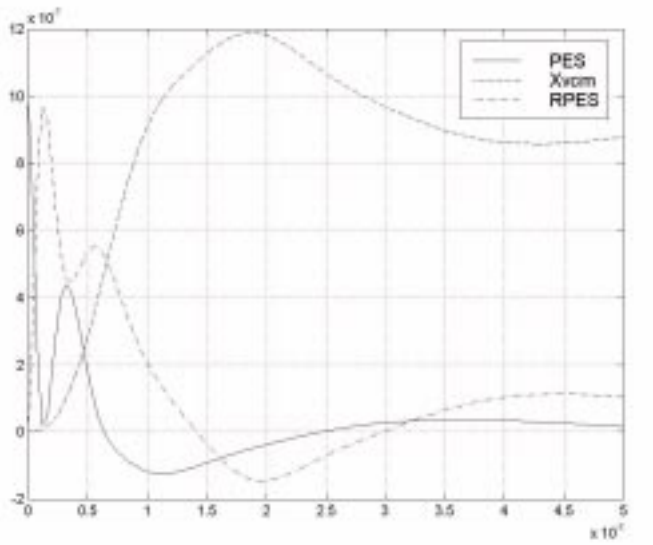
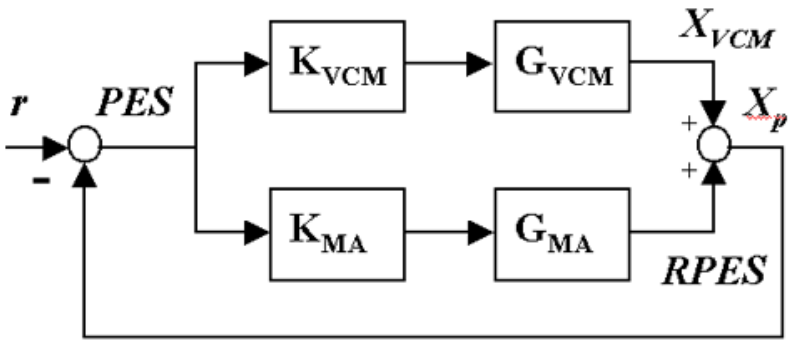


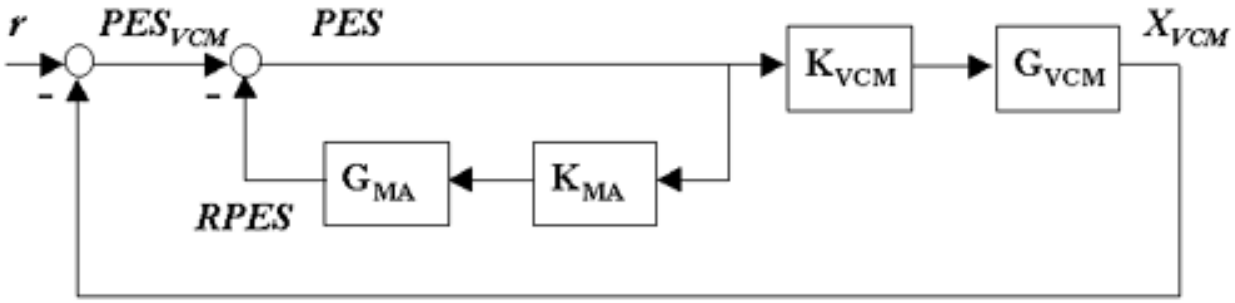
Figure 4.13 *Closed-loop responses of the PES, the suspension tip position (x_{VCM}), and the RPES for the IBM design without RPES sensing.*

RPES). The microactuator feedback loop filters the PES_{VCM} to produce the PES, and as seen in Figure 4.15(a), the filter is high-pass. The job of this high-pass filter is to attenuate PES_{VCM} up to as high a frequency as is possible so that the bandwidth can be extended. Beyond just extending the bandwidth of

the servo system, which is critical for compensating for high-frequency NRRO (which tends to be loosely defined in the literature as an output-referred disturbance composed of the effects of disk flutter, head windage, and servo-loop amplification of noise), the disturbance rejection curve is also pushed down significantly at low frequencies. So, in addition to bypassing the stiction problem, the microactuator eases the design of the entire servo system. For example, the requirements of a feedforward RRO compensation scheme are greatly reduced because the sensitivity is already pushed down significantly at the most important (low frequency) harmonics of the spindle motor. If this high-pass filter is considered to be included in K_{VCM}, it is apparent that the real open-loop bandwidth of the VCM can be measured from the PES_{VCM} to X_{VCM} transfer function. Figure 4.8(b) shows that the open-loop VCM bandwidth is only about 400 Hz.



(a)



(b)

Figure 4.14 (a) *A simplified block diagram for a dual-stage system in which all the weights are ignored and the system is assumed to be decoupled. The VCM and microactuator work in parallel here.* ***(b)*** *By redrawing the block diagram so that the RPES sums after the runout (r), the microactuator and VCM can be viewed as working in cascade. For purposes of analysis, the block diagram makes more sense.*

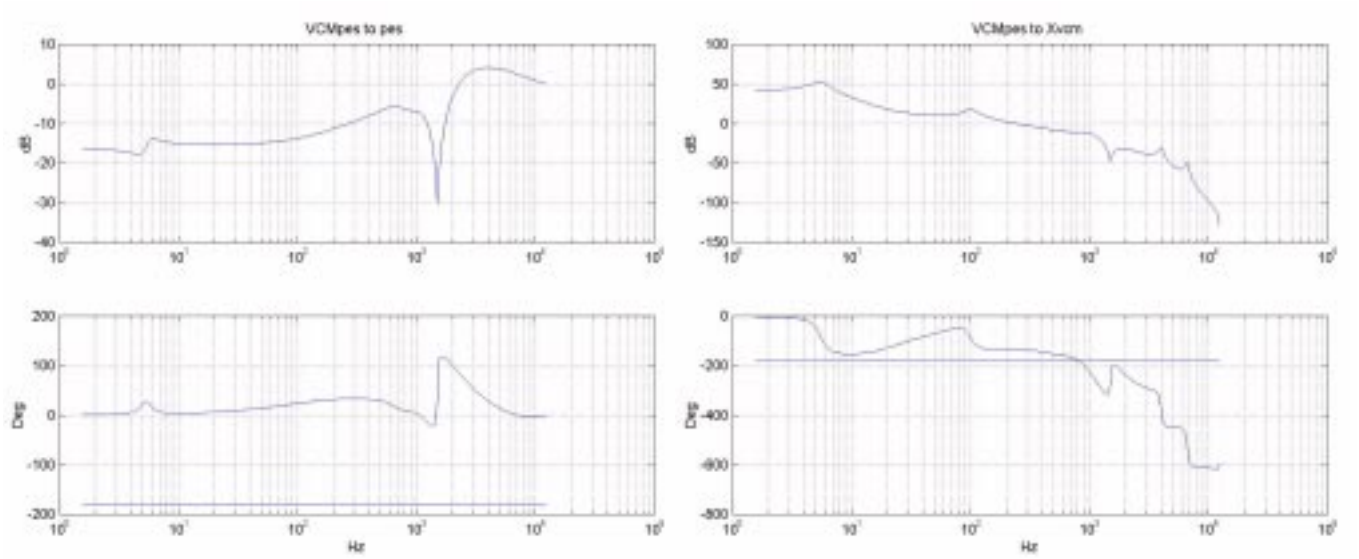


Figure 4.15 ***(a)*** *The high-pass filter-like transfer function from the distance between the suspension tip and the track center line (PES_{VCM}) to the PES for the IBM design without RPES sensing.*

Figure 4.8 ***(b)*** *Open-loop transfer function from PES_{VCM} to the suspension tip position (x_{VCM}) for the IBM design without RPES sensing. Note the low cross-over frequency.*

4.4 Hutchinson

4.4.1 Design Without RPES Sensing

The Hutchinson design is most easily compared to the IBM design without RPES sensing, since the Hutchinson Technology, Inc. (HTI) active suspension lacks the ability to measure its deflection. Although the sampling rate of 20 kHz assumed in this discrete-time design is lower than the 25 kHz sampling rate used in the IBM designs, the results are more impressive. This is due primarily to the fact that the first torsion mode of the Magnum 5 active suspension is at a very high frequency of about 7 kHz. In the IBM design the resonance of the microactuator is at 1.5 kHz, which is very close to the desired closed-loop bandwidth. A second reason why the HTI design looks better is that the HTI design was done assuming the same fictitious runout spectrum as in the design for the original rotational 3-mass model. The experimentally-based runout model used in the IBM designs is more dynamic, and hence presents a more challenging problem. The closed-loop runout to PES and runout to RPES sensitivities are shown in Figure 4.16(a) and Figure 4.9(b). The response of the system to a one micron step in the runout is shown in Figure 4.18. From the runout to PES sensitivity plot one can see that the low frequency attenuation of the PES is about 58 dB and the closed-loop bandwidth is about 1.3 kHz. Also, the step response shows quick settling (about 0.6 ms for a one-track seek) and very good frequency separation between the VCM and the microactuator.

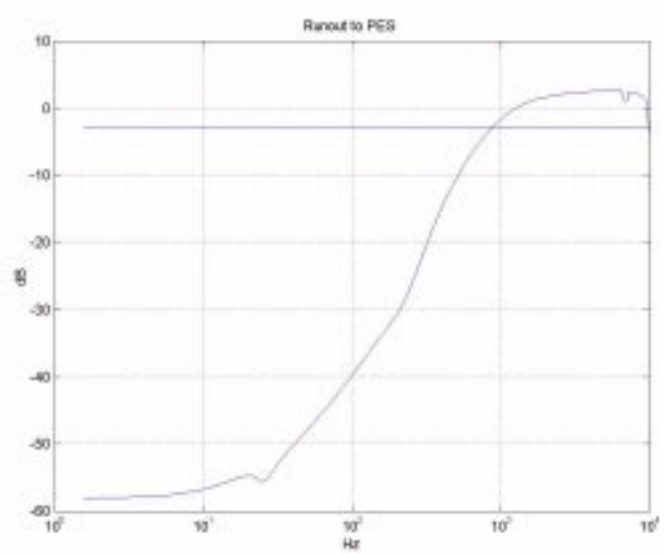


Figure 4.16 (a) *Closed-loop transfer function from runout to PES for the Hutchinson design.*

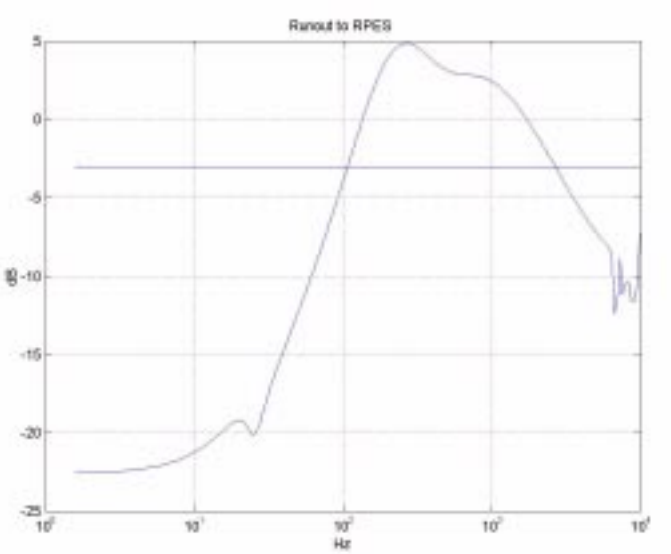


Figure 4.9 (b) *Closed-loop transfer function from runout to RPES for the Hutchinson design.*

Figure 4.17 (a) *Closed-loop transfer function from runout to head position (x_p) for the Hutchinson design.*

Figure 4.10 (b) *Closed-loop transfer function from runout to the suspension tip position (x_{VCM}) for the Hutchinson design.*

The same SISO analysis discussed in section 4.3 for the IBM design without RPES sensing can be applied to the HTI design. In fact, in this case the results are even clearer. Figure 4.19(a) and Figure 4.12(b) show the frequency responses of the transfer functions from PES_{VCM} to PES and PES_{VCM} to X_{VCM}. Figure 4.19(a) shows that the microactuator feedback loop illustrated in Figure 4.14(b) clearly looks like a

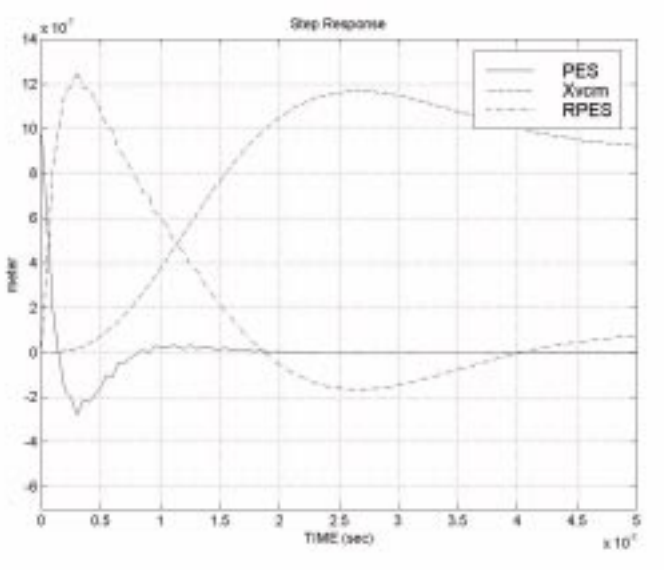


Figure 4.18 *Closed-loop responses of the PES, the suspension tip position (x_{VCM}), and the RPES for the Hutchinson design.*

high-pass filter, with low frequency attenuation of about 35 dB and a cross over of about 1.3 kHz, which is the closed-loop bandwidth. Figure 4.12(b) shows that the VCM bandwidth is less than 200 Hz, implying that cheaper, less accurate VCMs could be used. Essentially, the VCM design would be dominated by seek requirements and tracking requirements would be taken care of by the microactuator.

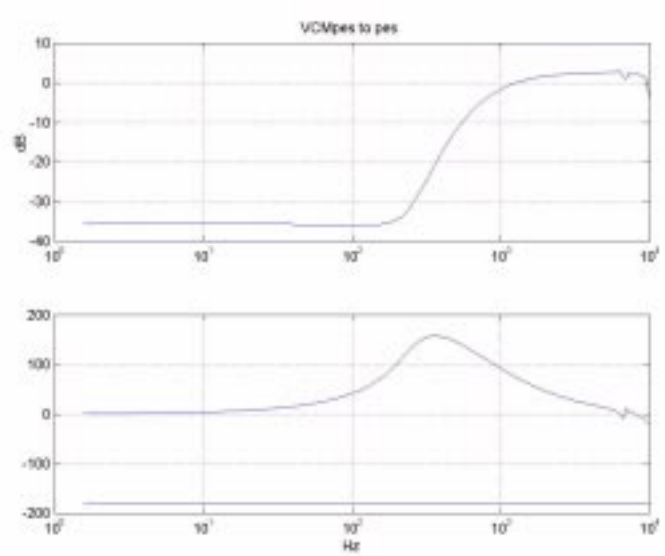


Figure 4.19 (a) *The high-pass filter-like transfer function from the distance between the suspension tip and the track center line (PES_{VCM}) to the PES for the Hutchinson design.*

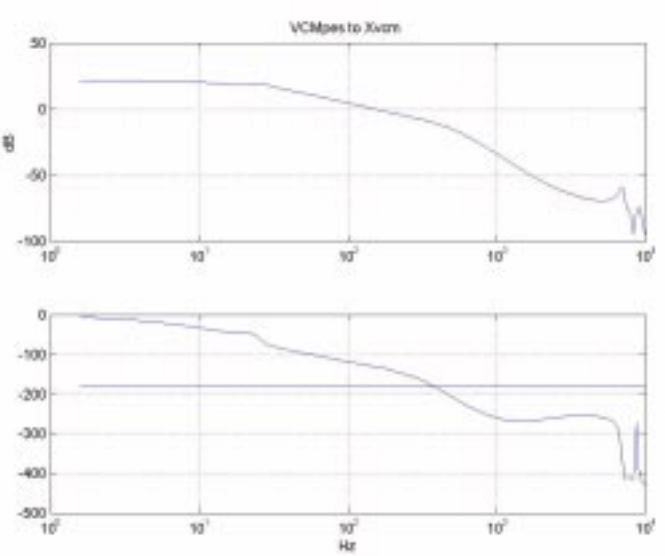


Figure 4.12 (b) *Open-loop transfer function from PES_{VCM} to the suspension tip position (x_{VCM}) for the Hutchinson design. Again, the cross-over frequency is very low, implying that a lower performance, cheaper VCM can be used.*

CHAPTER 5

Automation of Controller Synthesis: DSTAGE

5.1 What is DStage?

DStage is a Matlab program written by Dan Hernandez and is a tool for quickly designing robust controllers for dual-stage servo systems. The methodology of the controller design is essentially the same as is presented in this report. DStage was written using Matlab version 5.1 and requires the Matlab Mu-Analysis and Synthesis Toolbox, which is used heavily for the controller synthesis. A copy of DStage is on the CD containing this report, and a copy should be floating around 5121 Etcheverry on a floppy disk or on a hard drive. Also, DStage is available over the web at http://canaima.berkeley.edu/research/nsic/? provided the password which allows access to the dstage.zip file is known. The development of DStage was funded by NSIC (the National Storage Industry Consortium, http://www.nsic.org).

5.2 Program Features

DStage is, essentially, a wrapper for the Mu-Analysis and Synthesis Toolbox that allows anyone with limited control experience to use the advanced method of mu-synthesis for controller design for dual-stage servo systems. DStage does not contain new algorithms. Rather, it strings together the functionalities of the Mu-Analysis and Synthesis Toolbox (“Mu-Tools“) specifically for the dual-stage servo problem. The basic features that DStage offers are: problem definition, controller synthesis, controller analysis, and model

reduction. These basic features are accessed through three different DStage windows: the Design Window, the Analysis Window, and the Model Reduction Window. The first two of the basic features, problem definition and controller synthesis, are reached through the Design Window. Controller analysis and model reduction are reached through the Analysis Window and the Model Reduction Window, respectively. The following description of the program's features will also serve as a tour of the program. Hence the features will be described in the order in which they would naturally be encountered.

5.2.1 The Design Window

To start DStage, start Matlab and go to the Dstage\Program\ directory and type dstage (or if the Program directory is in Matlab's path, just type dstage). The DStage window will appear with two buttons on it, one labeled "Start a new session" and the other labeled "Load an existing .mat file". Loading an existing .mat file allows the user to continue a previously started session. Choosing to start a new session brings up a file dialog box for choosing a dual-stage HDD model file, as shown in Figure 5.1. The model file should be a Matlab function with no inputs and with a Mu-Tools system and an optional scaling choice as its outputs. The structure of the required dual-stage HDD system appears behind the file dialog box in the DStage window. Sample model files (mkGibm.m, Hutch.m, hutch_svcm.m, mkGrot.m, mkGtrans.m) can be found in the \Dstage\Models\ directory. The scaling choice can be either "conmuscl" or "ctprescl" and chooses between two functions for doing state transformations and input/output scalings to improve the numerical conditioning of the systems prior to doing the controller synthesis. The default choice is "ctprescl." Although DStage has these functions to help with the scaling of the system (which are very important in MIMO systems since they affect the spread of the singular values), the importance of using properly sized units for the model cannot be stressed enough. Often, any further scaling steps will be unnecessary if the units are carefully chosen and sometimes post-processing scalings on the system like `conmuscl` and `ctprescl` will not be enough.

Upon selecting a model file, a dialog box appears asking whether the design is to be done in continuous time or discrete time. If discrete time is selected, a second dialog box will appear, prompting the user for a sample time. Three buttons then appear in the DStage window, a "Design" button, an "Analysis" button, and a "Model Reduction" button. Clicking on the "Design" button will bring the user to the Design Window, with the other buttons operating similarly. This screen will appear every time the user goes between the Design, Analysis, and Model Reduction Windows.

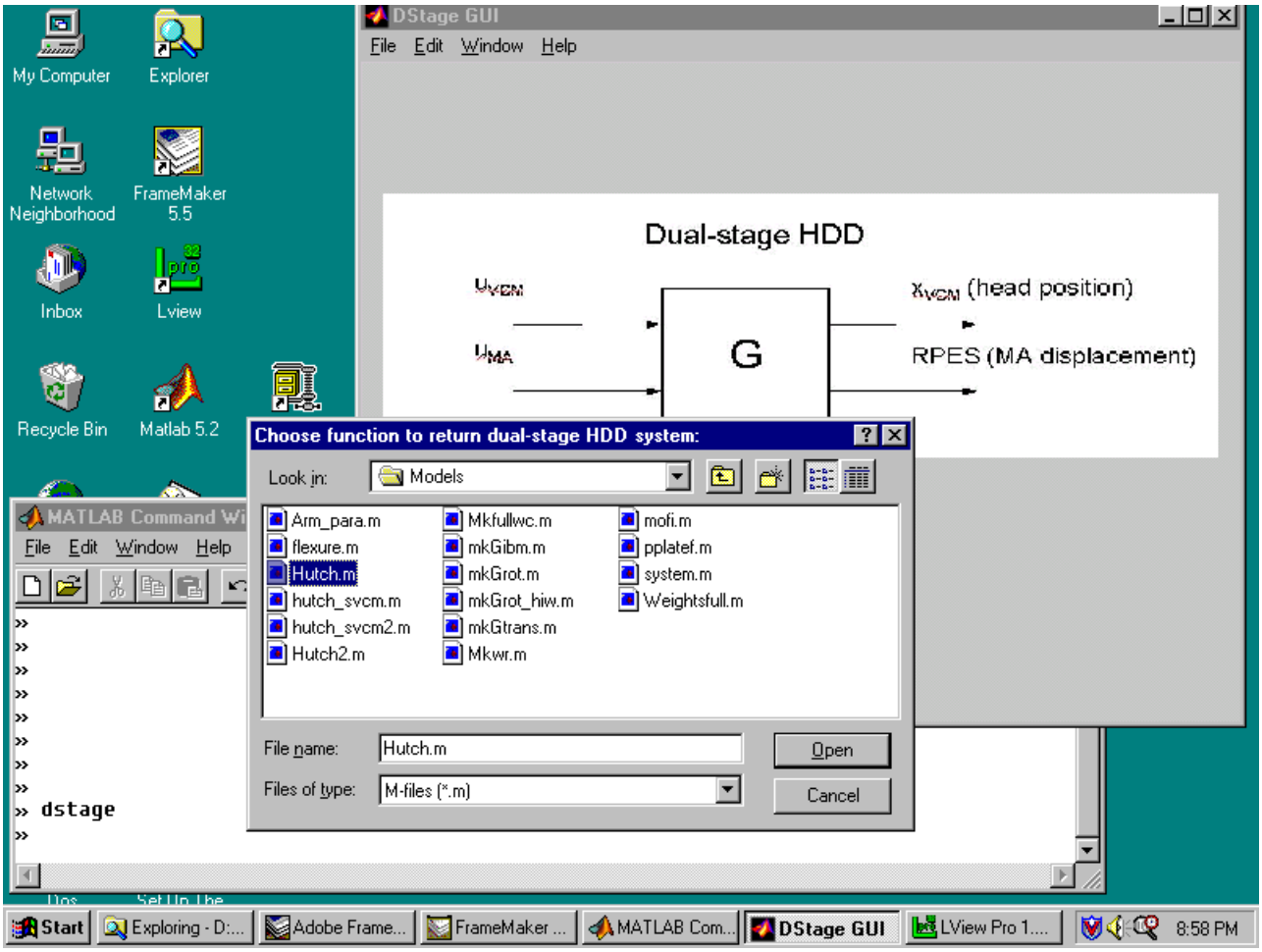
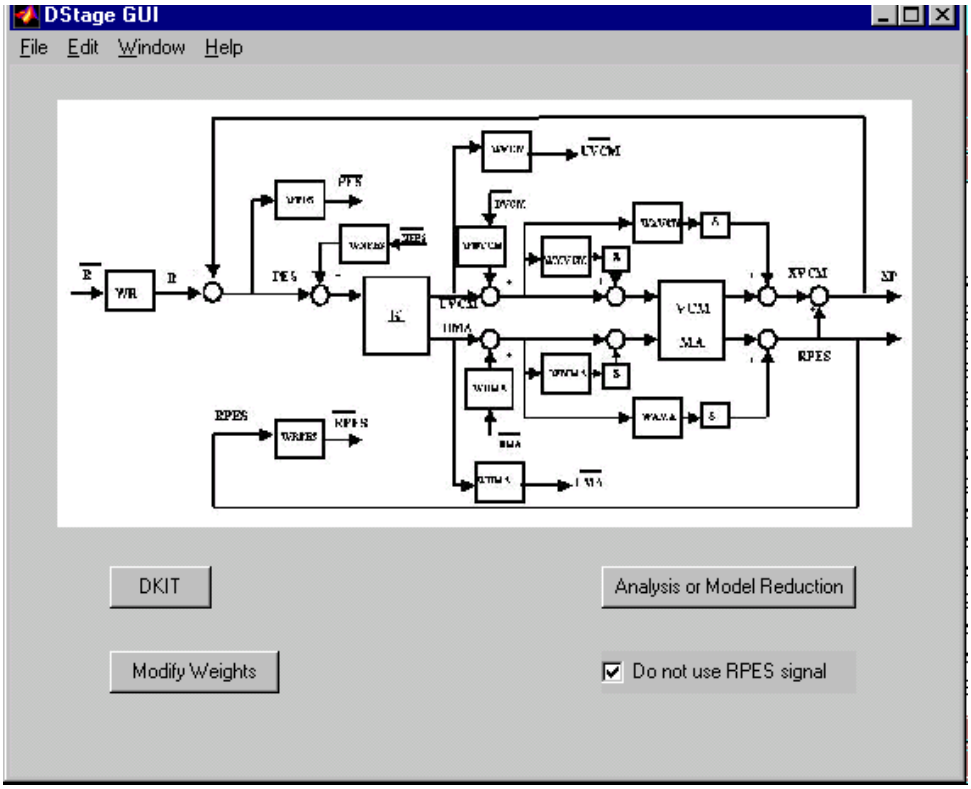
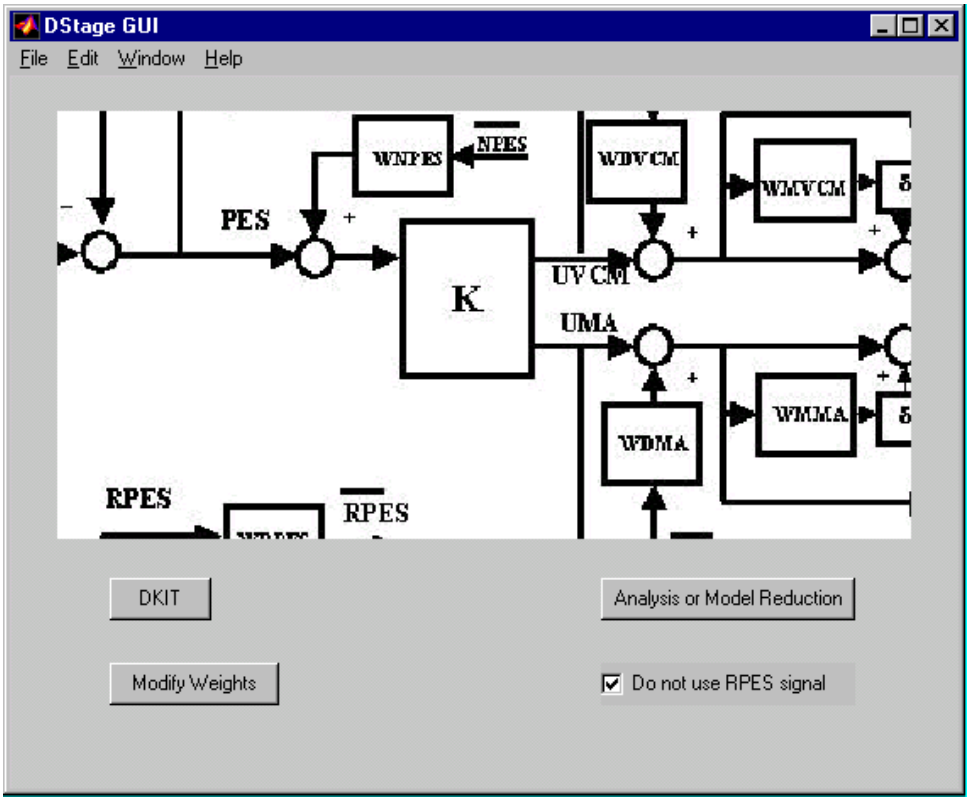


Figure 5.1 *The dialog box that prompts the user to select a dual-stage HDD model file. The model file must return a Mu-Tools system matrix with inputs u_{VCM} and u_{MA} and outputs x_p (the head position) and RPES as shown in the block diagram behind the dialog box.*

The Design Window is shown in Figure 5.2(a). Most of the window is occupied by a block diagram of the system used in the mu-synthesis design, including all of the possible weighting functions. This block diagram is almost the same as that shown in Figure 3.4 of section 3.1.3 in this report, except that in Figure 3.4 only additive uncertainties are shown and some of the names are different (e.g., W_{U1A} is used for the additive VCM uncertainty instead of W_{AVCM}). The block diagram is rather hard to read in the window, so to see what the blocks are, the user can either adjust the size of the window or zoom in on parts of the block diagram using the mouse, as shown in Figure 5.2(b). The “Do not use RPES” checkbox can be toggled to turn off/on the RPES sensing capability. The “Modify Weights”, “DKIT”, and “Analysis and Model Reduction” buttons are so the user can modify the weights before running the D-K iterations, run the D-K iterations, and exiting to the Analysis or Model Reduction Windows, respectively.



(a)



(b)

Figure 5.2 ***(a)*** *The Design Window, where weights can be modified and D-K iterations can be run at the touch of a button.* ***(b)*** *An illustration of the ability to zoom in on the block diagram, which is necessary due to the small sizes of the features and the poor resolution of the image display.*

5.2.2 Modifying Weights

Clicking on “Modify Weights” brings up a list box and selecting a weight brings up a screen for modifying the weight in the DStage window, as in Figure 5.3 for the modification of the W_R weight. The frequency response of the weight (which is specified by a continuous time transfer function) is plotted in the top half of the window. The transfer function for the weight may be selected as being a constant, a 1^{st} order transfer function, a 2^{nd} order transfer function, or a transfer function returned by a function file (see for example mkWr.m in the \Dstage\Models\ directory). Slider and text box controls are provided for easily adjusting the constant, 1^{st} order, and 2^{nd} order weights. Also, many of the weights may be turned off by clicking on a checkbox in the lower right hand corner of the window. The displays for modifying the W_R, W_{PES}, W_{RPES}, W_{AVCM}, and W_{AMA} weights are special in that they display more information. The displays for W_R and W_{PES} show the approximate upper bound on the runout to PES sensitivity discussed in section 3.1. The display for W_{RPES} shows the approximate upper bound on the runout to RPES sensitivity. Lastly, the displays for W_{AVCM} and W_{AMA} (see Figure 5.4) show the additive uncertainties with the plant frequency responses, which is useful because the intersection of the additive uncertainty with the plant’s magnitude response tends to clearly mark where the bandwidth will be cut-off.

At this point, a few comments should be made about how the weights are implemented. When the weights are second and third order transfer functions that are not loaded from a file, `ModWts.m` creates them as zero-pole systems with repeated poles and zeros. This is, in fact, a flaw that was carried over from the initial work of Sanjay Aggarwal, in which he used a runout filter with repeated poles and zeros. Although it was unknown to me then, the use of systems with repeated eigenvalues in controller synthesis methods that use state-space forms will generally cause numerical problems, since they rely on Schur forms that become relatively inaccurate when there are repeated eigenvalues. A better choice for the structure of the weights would be a butterworth filter, in which the poles are evenly spaced (in the s-plane) along a semi-circle corresponding to a specific frequency (see the Appendices in [24]).

5.2.3 D-K Iterations

After setting up the model and weighting functions, controller synthesis using D-K iterations can be performed at the click of a button. Clicking on “DKIT” starts the D-K iterations and displays many of the intermediate results, including data from the gamma iterations, singular value plots, D-scale fittings, mu-

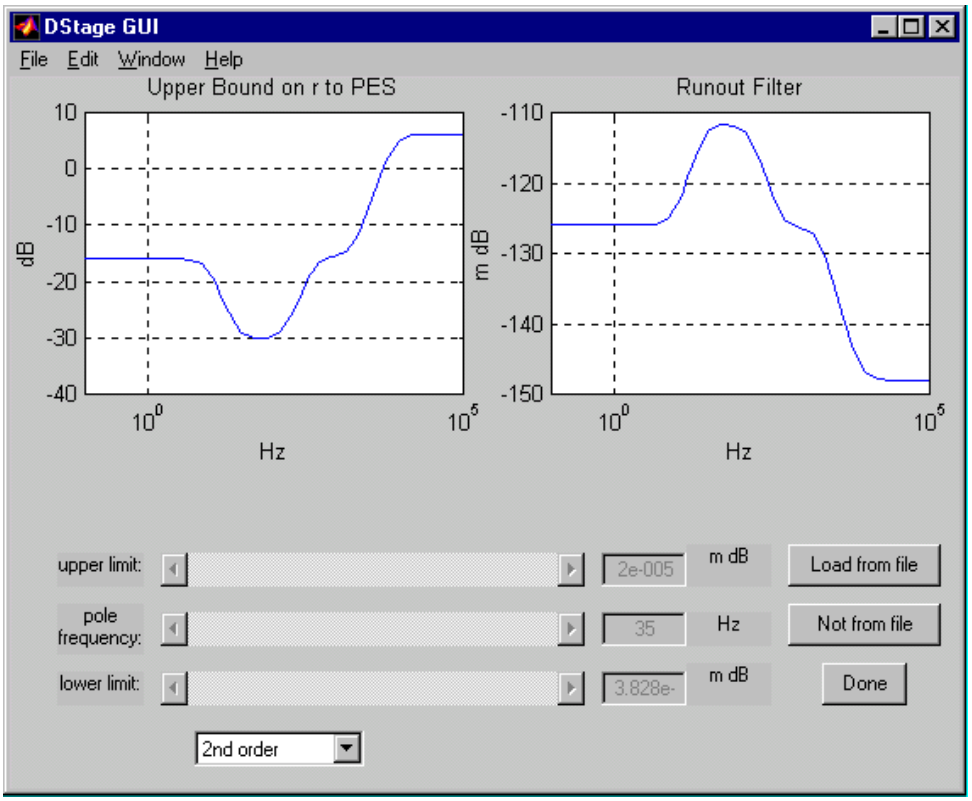


Figure 5.3 *The window for modifying the runout filter, W_R. The user can select from 0^{th}, 1^{st}, or 2^{nd} order filters that are easily adjusted using the sliders and text boxes. Alternatively, the filter can be loaded from a file. Note that two plots are shown, the one on the right shows the frequency response of the runout filter while the one on the left shows its affect on the approximate upper bound on the runout to PES sensitivity.*

plots, and iteration summaries (see Figure 5.5 for an example). For information on what all of these intermediate results mean, see section 3.1 or references [2], [3], and [4]. Upon finishing the D-K iterations, DStage returns to the Design Window.

5.2.4 The Analysis Window

After generating controllers, the user can go to the Analysis Window (see Figure 5.6) to evaluate the performance of the closed-loop system. The controllers from the last D-K iteration in the current session will be listed in the “Controller:” listbox. Alternatively, the user may load the controllers from a previous session by clicking on the “Change File” button (the current file is listed just below this button). The Analysis Window allows for quick and easy assessments and comparisons of the performances of various designs by plotting several results. Simply select the controller to be used, the plot type, and the figure in which the plot will be displayed, and click the “Plot” button. Table III is a summary of the available plots. Note that

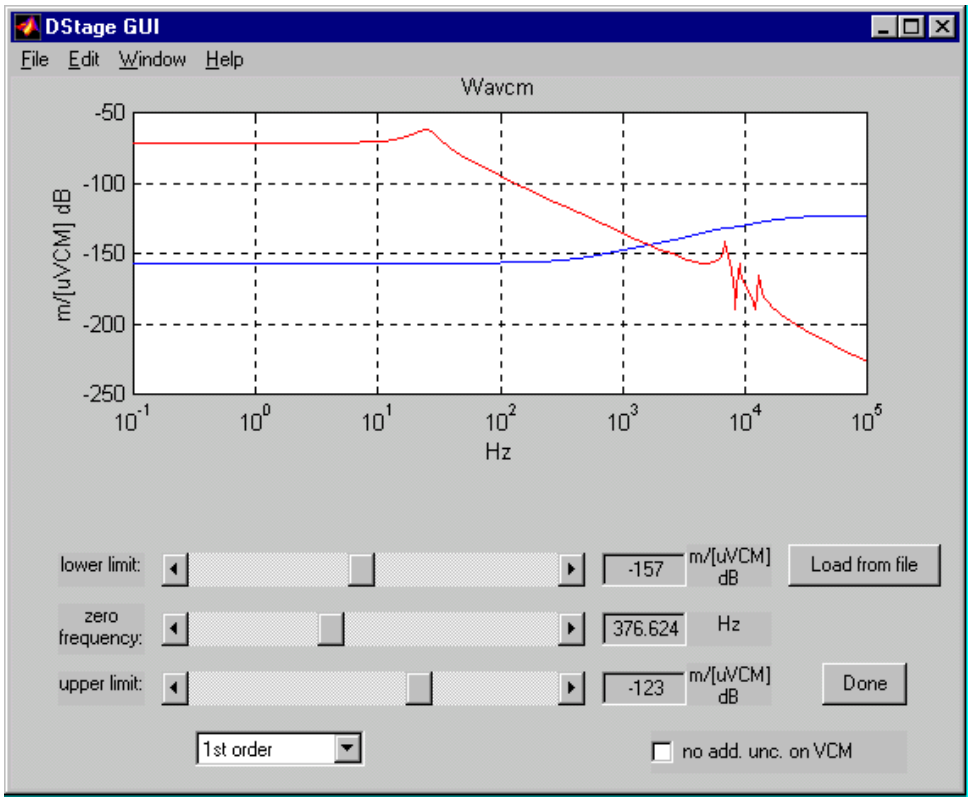


Figure 5.4 *The window for modifying the additive uncertainty weight W_{AVCM}. Again, the user can select from 0^{th}, 1^{st}, or 2^{nd} order filters load a transfer function from a file. The plot shows both the uncertainty weighting and the VCM model. Since the plot is logarithmic, the intersection of the two curves serves as a very good approximation to the maximum closed-loop bandwidth of the actuator.*

Table III: Plot Types Available In the Analysis Window

Plot Type	*Transfer Function/Signal*	*Plot Type*	*Transfer Function/Signal*
Mu Plot	-	*System and Weights*	Plant transfer functions
Open-loop Frequency Response	PES to xhead		Weights used in design
	PES to xVCM	*Step Response*	Position response to step in runout
	PES to RPES		uVCM response to step in runout
Closed-loop Frequency Response	runout to PES		uMA response to step in runout
	runout to xVCM	*Controller Bode Plots*	2 (SIMO) or 4 (MIMO) controller transfer functions
	runout to RPES	*Performance Plots*	PES
	runout to xhead		RPES
	runout to uVCM		uMA

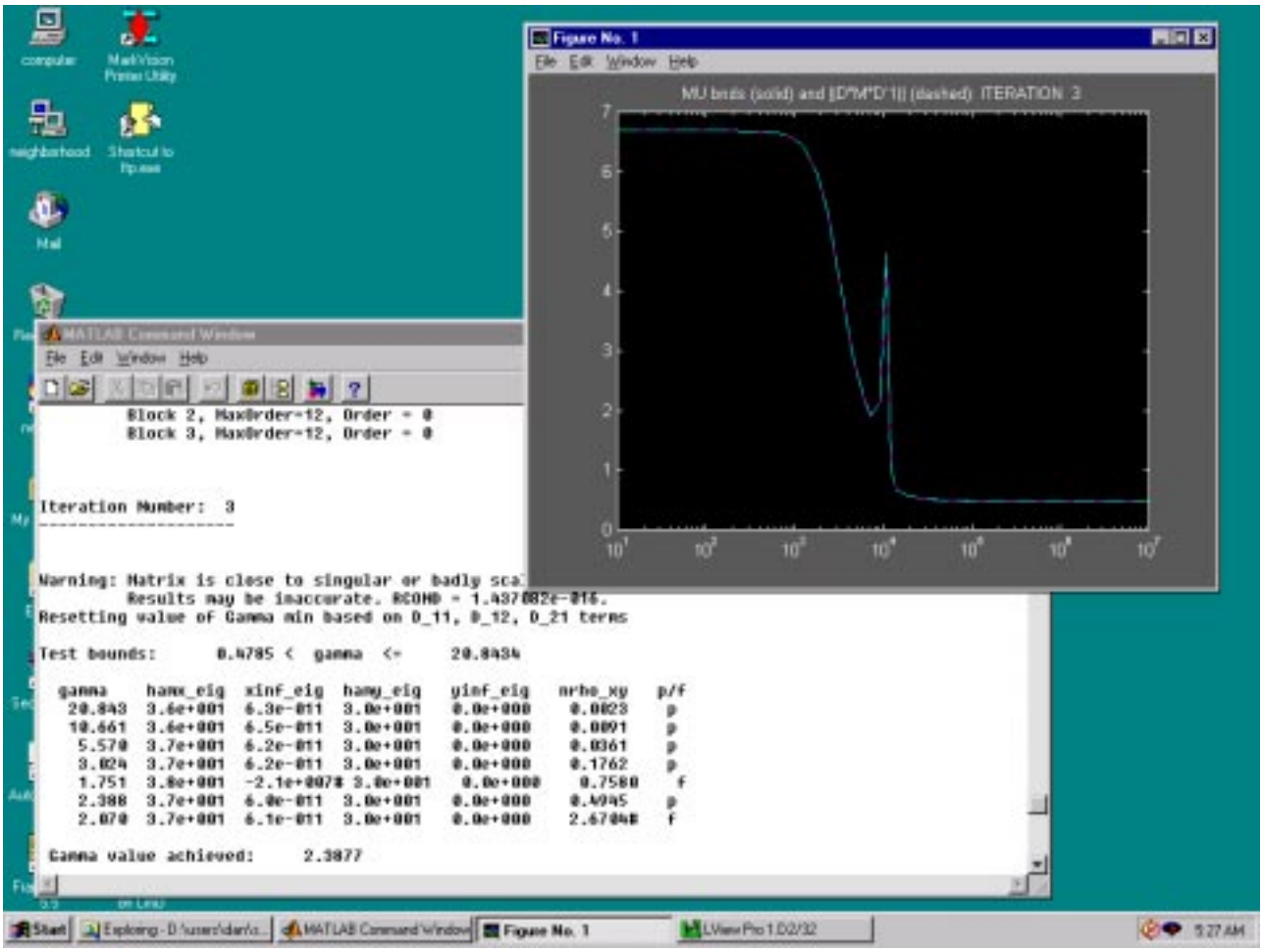


Figure 5.5 *The intermediate results such as singular value plots, mu plots, and gamma iteration results appear during the automated D-K iterations. This controller synthesis uses the Mu-Analysis and Synthesis Toolbox.*

Table III: Plot Types Available In the Analysis Window

Plot Type	*Transfer Function/Signal*	*Plot Type*	*Transfer Function/Signal*
	runout to uMA	*Pole-Zero Map*	-

the performance plots referred to in Table III are the normalized runout to the normalized output, and should always be less than one, especially under nominal conditions.

5.2.5 The Model Reduction Window

When it is decided that a design successfully meets the desired robustness, performance, and aesthetic criteria, the user can go to the Model Reduction Window to reduce the size of the controller. The Model Reduction Window, which is shown in Figure 5.7, is very similar to the Analysis Window in appearance. Again, the upper left hand corner shows the current file and the upper right hand corner lists the controllers

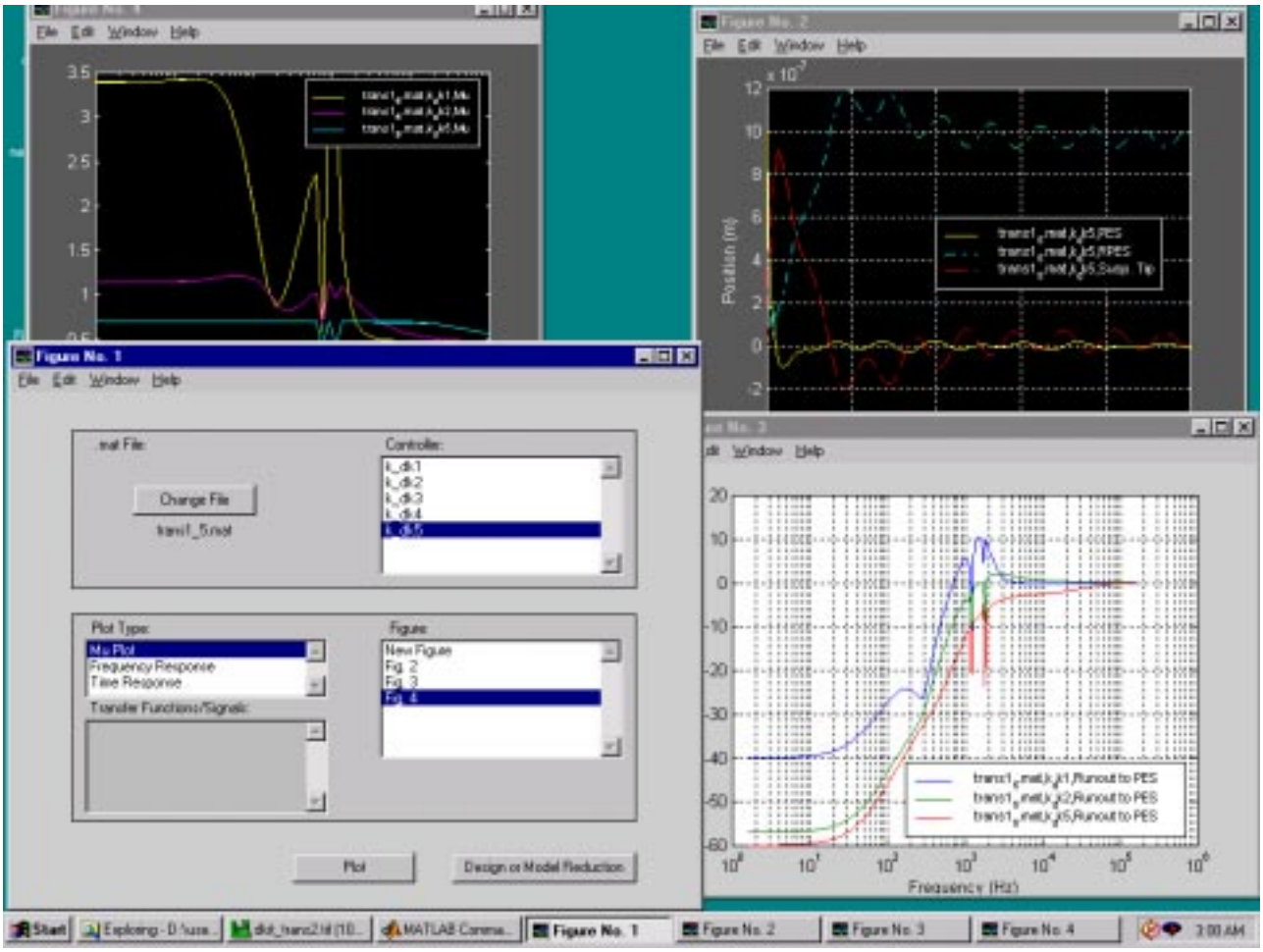


Figure 5.6 *Many useful results can easily be plotted and compared from the Analysis Window, which is shown in the bottom left hand corner of this screen capture.*

available. The minor changes are that the Model Reduction Window allows the user to save results to a file (a "*" will appear after the name of the current file if the user has produced model reduced controllers that have not been saved to a file) and that the "Controller:" listbox has a button below it for removing controllers from the list. This "Remove Controller" button exists because the number of controllers in a file could become very large without it. In the lower half of the Model Reduction Window the user can choose whether or not to plot the controller Bode plots and/or the closed-loop mu plots and can initiate the model reduction. There is also a listbox for selecting the model reduction method, however, the "Balanced Realization/Hankel Norm" selection is the only method that will do anything. The other selections are unconnected, which reflects that this software was a work in progress. Fortunately, the "Balanced Realization/Hankel Norm" selection, which uses the Mu-Tools command hankmr to do an optimal Hankel norm approximation of the controller, is the method that works the best. Upon clicking on "Try MR", the user will be prompted for a range for the number of states over which model reduction will be attempted.

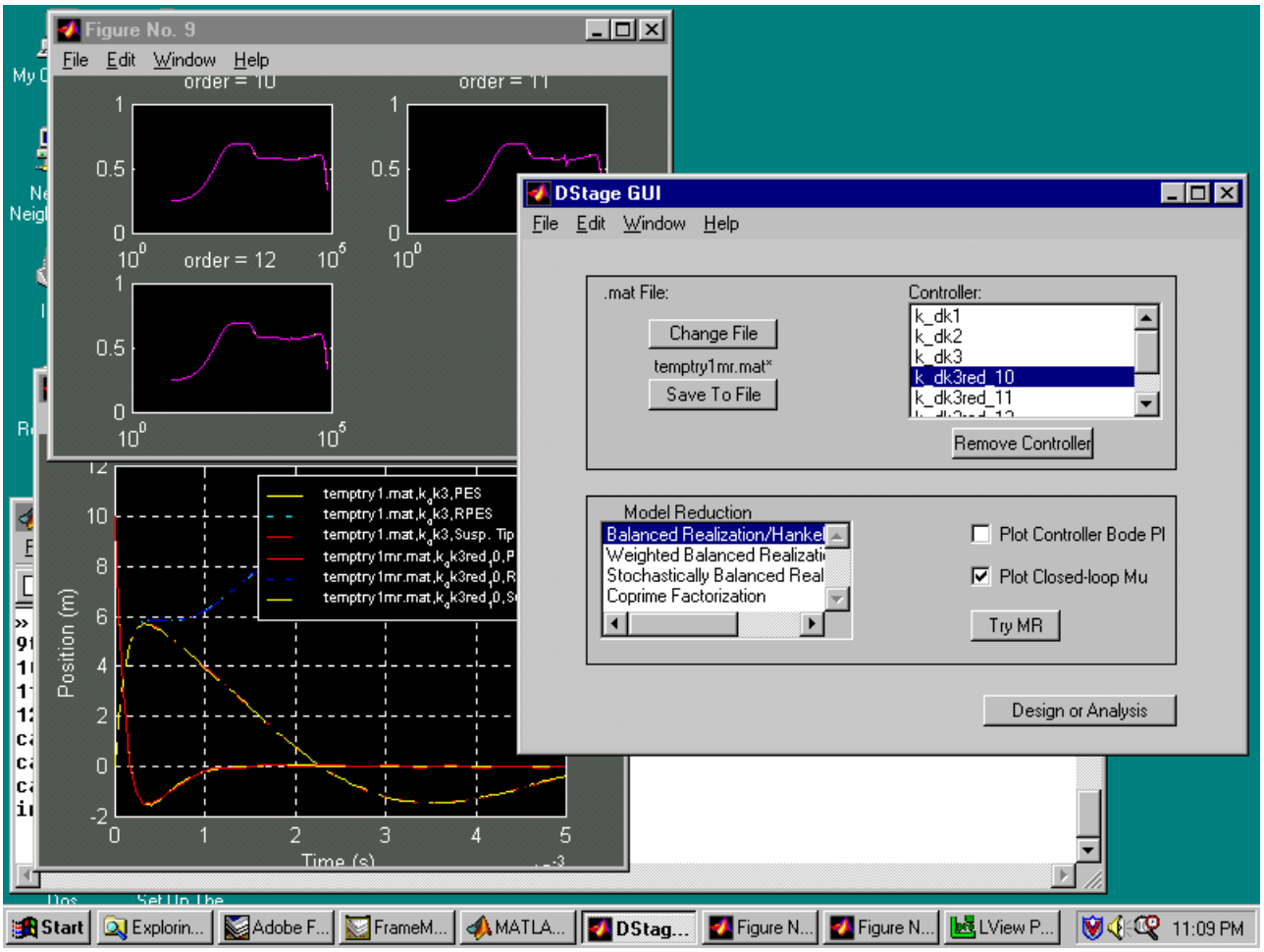


Figure 5.7 *The Model Reduction Window can be used quick and painless model reduction of any of the controllers in the current session and can also load controllers from previous sessions that were saved to files. Currently, only the model reduction method available is the optimal Hankel norm approximation of a balanced realization. Fortunately, of the methods listed in the window, this method works the best.*

DStage will then attempt to model reduce the selected controller for each number of states in the range. For each attempt, the stability or instability of the reduced order controller will be reported in the Matlab Command Window, and the controller bode plots and mu-plots will be plotted if those options were selected. Note that if the original controller was itself unstable, DStage will issue an error and model reduction will not be attempted. The suggested procedure for model reduction is to run the model reduction algorithm without plotting anything, and then rerunning the model reduction with plotting the Bode or mu plots for only the orders that produced stable reduced controllers. This saves time and avoids generating a multitude of useless plots. Note that while the mu-plots best answer the question of whether or not the model reduction was successful, the Bode plots are useful for quickly determining how well the model reduction is doing. If the Bode plots differ substantially, then the mu plots will, almost surely, have peak values greater

than one.

As mentioned above, the method of model reduction that is used here consists of converting the discrete-time system to continuous-time using a bilinear (Tustin) transformation (since it preserves the H_∞-norm), doing a balanced realization of the system, getting the optimal Hankel norm approximation of order k of the balanced system, and converting the system back to discrete-time using the inverse bilinear transformation. If the system to be reduced is unstable (although you probably don't want to implement an unstable controller, even if it does stabilize the plant, you might need to do this if, say, you have a model obtained through system identification that has marginally unstable modes), then the system should be decomposed into stable and unstable parts and model reduction should be performed on the stable subsystem only. This decomposition can be done for continuous-time systems using `sdecomp` in the μ-Analysis and Synthesis Toolbox. The function can also be easily modified to decompose discrete-time systems. Unfortunately, the toolbox does not have very many model reduction techniques for discrete-time systems. If it is desired to make improvements in this area, then a model reduction technique based on balanced realizations for discrete-time systems should be used. There are several such algorithms in the literature that can be directly inserted into software code (e.g., see Safonov and Chiang, 1989). Also, a fairly good discussion of some techniques based on balanced realizations (e.g., balanced truncation, balanced residualization, and frequency weighted balanced reduction) can be found in [18].

5.3 Program Architecture

DStage was written using the standard Matlab techniques for writing a graphical user interface (GUI). The GUIDE (GUI Development Environment), which is new for Matlab 5, was not used. GUIDE mainly speeds up GUI layout, and can make complicated GUI designs very difficult by hiding important commands behind icons. It also appears to be difficult to build dynamic GUIs using GUIDE. Matlab makes it easy to write and maintain event-driven GUIs by looking after mouse and keyboard events. It is up to the author of the GUI to decide what happens when an event occurs, and this is done through the use "callback functions." For example, if the callback function for a button were disp('hi'), then 'hi' would be printed in the Command Window every time the button is clicked. Of course, in this example it doesn't make sense to use the term callback function. The reason the term is used is because it is typical to write re-entrant code. Although a program could be written so that each element of the GUI (e.g., a button, a slider, a text box, etc.) called a different function, a huge number of small files would be required. Hence, it makes

sense to have one (or a few) files and to just pass control flags. Lastly, it should be mentioned that since variables cease to exist outside of functions, data is stored in the figure in which the GUI is displayed. Each figure has a userdata variable, so data that needs to be preserved between calls can be stored as fields in a structure. In fact, all of the information that DStage preserves between windows and saves and loads to files are within this userdata structure that is passed from window to window through the callback functions.

There are three main files in DStage and several other files that perform specific subtasks. The main files are Dstage.m, Analysis.m, and MR.m. Although the code in these files may seem intimidating at first due to the length of the files, their roles are not very complicated. The truth is that there is a substantial amount of overhead involved in maintaining a presentable, seamless GUI that has all of the desired functionalities. Even something as simple as keeping track of how many lines are plotted on a figure, what controllers they used, and what their line colors are requires a fair amount of overhead. The main file is DStage.m, which takes care of initialization, load a model, handles the Design Window (including displaying the block diagram and running the D-K iterations), and handles the window that serves as a bridge between the Design, Analysis, and Model Reduction Windows. DStage.m calls ModWts.m to adjust or load the weighting functions. The bridge between the various windows consists of using another file for the callback function. For example, when the button labeled “Model Reduction” is clicked, since the callback function is MR, DStage.m will call MR.m. After control is passed to MR.m, it will clear the figure and redraw it as the Model Reduction Window. The file Analysis.m only looks after the Analysis Window and similarly the file MR.m only looks after the Model Reduction Window.

CONCLUSIONS

Given that disk drives are remarkably complex devices with many sources of tracking errors, the use of a second actuation stage appears to be an effective way to compensate for all of the components of TMR at once. The case studies in this report have shown that bandwidths of a couple of kilohertz can be expected in a dual-stage disk drive using the current technology. This bandwidth should be high enough to allow drives with track densities in excess of 25 ktpi, with preliminary analysis indicating that track densities upward of 100 ktpi are possible. This is much larger than the track densities of about 6 ktpi in today's high-end laptop drives, which have the highest areal densities. The practicality of such a solution will, of course, depend upon economics. If the second stage actuator can be batch produced and easily integrated with the rest of the drive, then dual-stage disk drives should still be quite cost competitive.

The use of mu-synthesis has also been shown in this report to be an effective way to design high-performance, robust controllers. This type of controller design is especially useful for designing controllers for prototype drives, where the dynamics of the system may be very complicated and not completely understood. The controller design procedure, like many modern design techniques, uses frequency weightings to describe performance requirements and a combination of frequency weightings and parametric uncertainty to describe the norm-bounded uncertainties. This technique could also be very useful in production drives, since there is uncertainty in the plant model due to unit-to-unit drive variations caused by the manufacturing process. A software tool, DStage, has also been written in Matlab to assist the design and analysis of μ-controllers for dual-stage disk drives. This program, which interacts with the user almost entirely through a GUI, uses the μ-Analysis and Synthesis Toolbox for Matlab to generate and analyze controller designs.

Although a large amount of work has already been done on this topic, there are still many areas of research

to pursue. The refinement and implementation of controllers using these techniques are, of course, necessary to continue the life and usefulness of the project. A comparison of these H_∞-based controllers with minimum variance controllers, in terms of both robustness and performance in a stochastic environment, might be a good starting point. It would also be useful for the designer to develop a better disk drive simulator, that includes nonlinearities as well as accurate characterizations of the disturbances entering into the system, including runout. It is likely that both implementation of the controller and the creation of a better simulator will require a stronger interaction with members of the disk drive community to be successful. Other areas of potential research include the design of a mixed H_2/H_∞ controller, or the design of a multi-rate controller in which the microactuator's RPES could be sample at a much faster rate than the PES. The mixed H_2/H_∞ controller problem has been solved for the state-feedback and SISO output feedback cases, but has not, to the best of the author's knowledge, been solved for the MIMO output-feedback case. The multi-rate H_2 and H_∞ problems have been solved using the lifting technique in combination with various other methods, including the use of periodic Riccati equations. There is still a great deal of work that can be done in this area. Again, strong interaction with the industry will make the work both more enjoyable and more profitable.

REFERENCES

[1] Satinderpall (Sat) Pannu, “Adaptive Feedforward Servo for Disk Drives,” Doctoral Dissertation, University of California at Berkeley, 1998.

[2] G. Franklin, J. Powel, M. Workman, *Digital Control of Dynamic Systems,* 3rd ed., Addison-Wesley Longman, Inc., 1998.

[3] K. Ashar, *Magnetic Disk Drive Technology: Heads, Media, Channel, Interfaces and Integration*, IEEE Press, 1997.

[4] C. D. Mee, E. Daniel, *Magnetic Recording Technology*, 2nd ed., IEEE and McGraw-Hill, 1996.

[5] C. D. Mee, E. Daniel, *Magnetic Storage Handbook*, 2nd ed., IEEE and McGraw-Hill, 1997.

[6] A. Sacks, W. Messner, “MR Head Effects on PES Generation: Simulation and Experiment”, *IEEE Transactions on Magnetics*, v. 32, no. 3, pp. 1773-1778, May 1996.

[7] L.S. Fan, H. Ottesen, T. Reiley, R. Wood, “Magnetic Recording Head Positioning at Very High Track Densities Using a Microactuator-Based, Two-Stage Servo System,” *IEEE Trans. of Industrial Electronics*, v. 42, no. 3, pp. 222-233, June 1995.

[8] R. Ehrlich, D. Curran, “Major HDD TMR Sources, and Projected Scaling with TPI,” *IEEE Trans. on Magnetics*, v. 35, no. 2 (pt. 1), pp. 885-891, March 1999 (presented at the Second Asia-Pacific Magnetic Recording Conference (APMRC '98)).

[9] D. Abramovitch, T. Hurst, D. Henze, “An Overview of the PES Pareto Method for Decomposing Baseline Noise Sources in Hard Disk Position Error Signals,” *IEEE Trans. on Magnetics*, v. 34, no. 1, pp. 17-23, January 1998.

[10] J. McAllister, “Disk Flutter: Causes and Potential Cures,” *Data Storage*, v. 4, no. 6, pp. 29-34, May/June 1997.

[11]E. Grochowski, R. Hoyt, “Future Trends in Hard Disk Drives,” *IEEE Trans. on Magnetics*, v. 32, no. 3, pp. 1850-1854, May 1996.

[12]J. Davis, “Beyond the Superparamagnetic Limit II: Far-field Recording,” *Data Storage*, Feb. 1996, pp. 33-36 (also available at http://www.datastorage.com).

[13]G. Knight, “Beyond the Superparamagnetic Limit I: Near-field Recording,” *Data Storage*, Feb. 1998, pp. 23-30 (also available at http://www.datastorage.com).

[14]David Horsley, “Microfabricated Electrostatic Actuators for Magnetic Disk Drives,” Doctoral Dissertation, University of California at Berkeley, 1997.

[15]J. Doyle, K. Zhou, K. Glover, *Robust and Optimal Control*, Prentice-Hall, Inc., 1996.

[16]K. Zhou, J. Doyle, *Essentials of Robust Control*, Prentice-Hall, Inc., 1998.

[17]G. Balas, J. Doyle, K. Glover, A. Packard, R. Smith, μ-*Analysis and Synthesis Toolbox User’s Guide*, MUSYN, Inc., and The MathWorks, Inc., 1995.

[18]S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control*, John Wiley & Sons Ltd, 1996.

[19]J. Doyle, B. Francis, A. Tannenbaum, *Feedback Control Theory*, Macmillan Publishing Company, 1992.

[20]A. Sacks, M. Bodson, W. Messner, “Advanced Methods for Repeatable Runout Compensation,” *IEEE Trans. on Magnetics*, v. 31, no. 2, pp. 1031-1036, March 1995.

[21]S. Weerasooriya, S., J.L. Zhang, T.S. Low, “Efficient Implementation of Adaptive Feedforward Runout Cancellation in a Disk Drive,” *IEEE Trans. on Magnetics*, v. 32, no. 5, pt. 1, pp. 3920-3922, Sept. 1996.

[22]C. Kempf, W. Messner, M. Tomizuka, R. Horowitz, “Comparison of Four Discrete-Time Repetitive Control Algorithms,” *IEEE Control Systems*, v. 13, pp. 48-54, December 1993.

[23]Sanjay Aggarwal, “Design and Control of Micro-actuators for High Density Disk Drives,” Master’s Report, University of California at Berkeley, 1997.

[24]A. Oppenheim, R. Schafer, *Discrete-Time Signal Processing, Prentice-Hall*, 1990.

[25]M. Safonov, R. Chiang, “A Schur Method for Balanced Truncation Model Reduction,” *IEEE Trans. on Automatic Control*, v. 34, pp. 729-733, 1989.