Physical Modeling of Probe-Based Storage

Tara M. Madhyastha* Katherine Pu Yang†

Abstract

Magnetic disks may be reaching physical performance limits due to the superparamagnetic effect. To close the performance gap between processors and storage, researchers are exploring a variety of new storage technologies [17]. Among these new technologies, probe-based micro-electrical mechanical systems (MEMS) magnetic storage arrays are attractive [3]. Probe-based storage is dense and highly parallel. It uses rectilinear motion in contrast to rotating media. Commercial devices are expected within the next several years.

The wide range of possible architectures and the unique performance characteristics of probe-based storage require that standard file system algorithms for disks, including scheduling and layout, must be revisited to determine their efficiency domain. Because these devices do not yet exist, analysis of system performance depends on simulation models. At this early stage of development, models that bridge the gap between the physics of the device and its performance characteristics can provide important feedback to both hardware and software designers.

This paper compares results from three models of probe-based storage that convey successively more accurate descriptions of the underlying physics. We conclude that the physical accuracy of the model has a significant impact on the predicted performance under real workloads.

1 Introduction

For over thirty years, magnetic disks have dominated secondary storage. During this time, increases in the speeds and capacities of other systems components and the I/O demands of these components have forced storage systems researchers to develop methods to keep

*Department of Computer Science, University of California Santa Cruz, 1156 High Street, Santa Cruz, CA 95064

†Cisco Systems, San Jose, CA 95134

pace. Although disk performance has improved, traditional hard drive technology may be nearing physical performance limits. To narrow the I/O performance gap, researchers are examining a variety of new storage technologies based on holography [7, 12, 13], probe-based AFM storage [4, 9], and probe-based MEMS magnetic storage [2, 3].

Among these new possibilities, probe-based storage is attractive. Probe-based micro-electrical mechanical systems (MEMS) storage arrays are fabricated using the IC fabrication process. Probe-based storage is dense, highly parallel, and uses rectilinear two-dimensional motion (in contrast to rotating media). Commercially viable devices are expected within the next several years.

The wide range of possible architectures and the unique performance characteristics of probe-based storage require that standard file system algorithms for disks, including scheduling and layout, must be revisited to determine their efficiency domain. Because these devices do not yet exist, analysis of system performance must depend on simulation models. At this early stage of development, models that bridge the gap between the physics of the device and its performance characteristics provide important feedback to hardware and software designers.

This paper compares results from three models of probe-based storage that convey successively more accurate descriptions of the underlying physics. We have implemented these models within the Pantheon [18] simulator and compared them using I/O traces with different workload characteristics. We conclude that the physical accuracy of the model has a significant impact on the predicted performance under real workloads. In particular, we conclude that a model based on classical optimal control theory is more accurate than previously proposed models [16, 6] and can be used to optimize architectural parameters for specific I/O workloads.

The remainder of this paper is organized as follows. We briefly survey related work in §2. We describe the architecture of a probe-based storage device in §3, and briefly describe data layout in §4. We describe a previously proposed model [6] and introduce the spring model and optimal control models in §5. We present the results of our experimental evaluation in §6, and conclude in §7 with implications for future work.

2 Background

The storage density that can be achieved using longitudinal magnetic recording is physically limited by the superparamagnetic effect. As bits become smaller, materials lose their ability to hold a charge at a given temperature. A promising way to achieve higher densities is by using an array of tips moved by micro-electrical mechanical machinery, rather than a single read/write head.

Many research centers are now researching and developing probe-based storage devices. The Carnegie Mellon Center for Highly Integrated Information and Storage Systems (CHI^2PS^2 [3]) is developing non-volatile, rewritable, low-cost, IC-based mass storage devices. An important aspect of the design is integration of storage and processing in a single chip.

The probe-based storage system described in §3 is based upon the CHI^2PS^2 design, but other nanoscale technologies are being developed. Despont *et al.* [4] and Mamin *et al.* [9] discuss a technology that uses tiny indentations, made in a polymer layer by atomic force microscopy (AFM) tips, to represent stored bits that can be read by the same tip that wrote them. The Atomic Resolution Storage (ARS) project at Hewlett-Packard Labs is investigating an array of atomic scale tips. Although these devices differ in how they physically record bits, they share an important characteristic: each uses a sparse tip array, driven by micromoters in the x and y dimensions, to read and write data.

Accurate models of system components, particularly performance-limiting components, are important for system design. Because probe-based storage devices do not yet exist, analysis of system performance depends on simulation models. Disk models have been in use as long as disks, and more sophisticated models simulate detailed features such as controller effects, caching, data layout, and head movement [15, 5]. As probe-based storage systems evolve, so does the interest in modeling them. Schlosser and Griffin *et al.* [16, 6] model the data seek and access dynamics for a probe-based (MEMS) storage system using first-order mechanics. They allow for non-constant acceleration by discretizing the movement of the sled. In contrast, the approach used here directly models the low-level physical characteristics of probe-based storage that lead to non-constant acceleration.

3 Probe-Based Storage Architecture

Probe-based magnetic storage systems consist of a substrate of magnetic media suspended over an array of many read/write tips, as shown in Figure 1. These tips are etched out of silicon using a process that is compatible with standard IC lithography. Depending upon the design, each tip can individually access data at a rate of approximately 100–200 Kbit/second. The tip array is sparse, enabling each tip to manipulate a rectangle of bits while the sled moves above it (one such rectangle is highlighted in Figure 1). Because the data rate for each tip is low, many tips are usually arranged in arrays, and bits are accessed in parallel to achieve higher overall bandwidth.

The media sled is suspended above the tip substrate by silicon beams that act as springs, and moved by forces generated by lateral resonant microactuators, as shown in Figure 2. In this figure, the shaded parts move and the unshaded parts are stationary. Electric forces

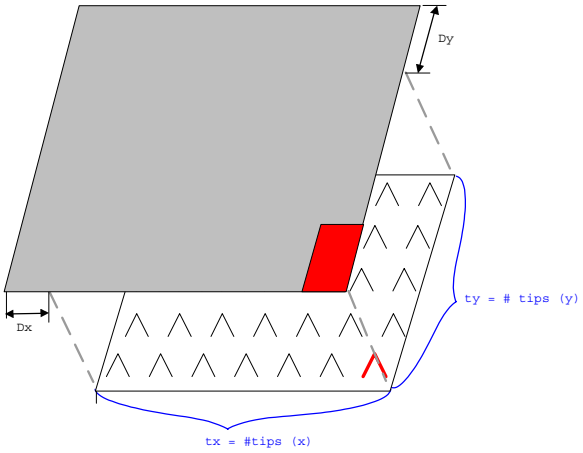


Figure 1: Probe-based storage device.

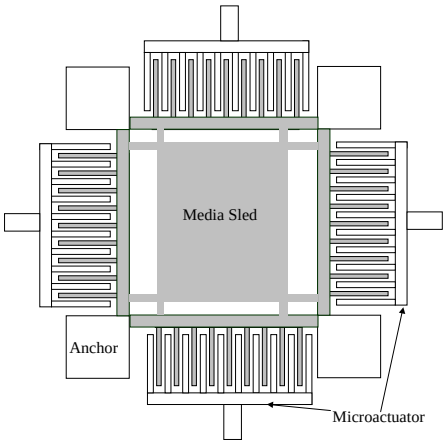


Figure 2: Sleds and microactuators.

applied to the fingers of the microactuator combs exert electrostatic forces on the sled that cause it to move in the x and y direction, overcoming the forces exerted by the anchors and beams that keep it in place. In mechanical terms, there are three forces that affect sled movement: the electromotive force produced by the actuator, the restoring forces from the springs, and the damping force from the air.

To service a read or write request, the sled first repositions itself so that the tip array can access the required data. This movement is called *seek time*. We describe three models for seek time in §5. Once positioned, the sled reads or writes the data by moving over it with a constant velocity. The *read/write time* is a function of the sled velocity and the data layout, and is modeled separately.

Our model parameters are based on a design for a probe-based storage device by Carley [3], further described by Schlosser *et al.* [16, 6]. We consider an array of 100 by 100 tips, with one media sled of 1 cm^2, and a media density of 50 nm per bit. Only one tip row may be active at a time, and each tip accesses data at 200 Kbit/sec. Many other architectural configurations are possible for probe-based storage. For example, one might vary the tip configuration, number of sleds, media density, and so on. Our model describes the motion of one sled and is easily parameterized to handle other multiple sled configurations.

4 Data Layout

Sectors are ordered on conventional rotating disks to maximize throughput for large, sequential requests. We can do the same for probe-based storage. Figure 3 shows a sample data layout scheme for a probe-based storage device with four tips. Figures 3a–d show

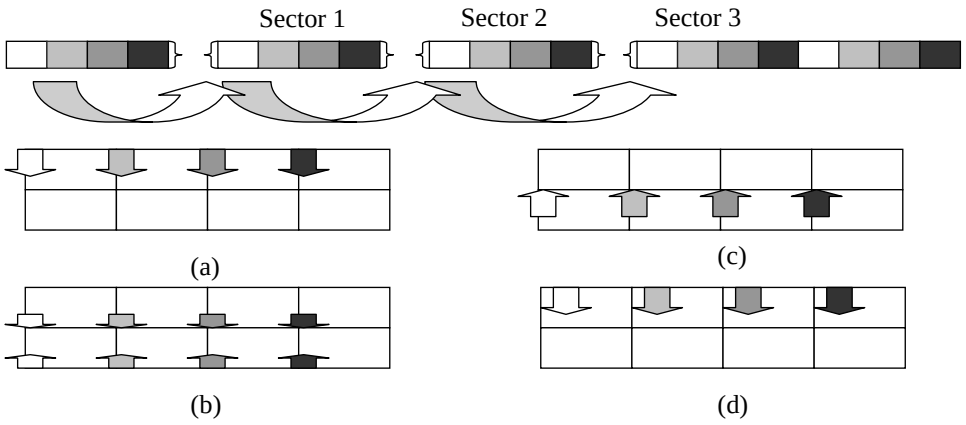


Figure 3: Mapping disk sectors to probe-based storage.

how sectors 0–3 are accessed. This data access pattern is based upon two assumptions: (1) only one row of tips are active simultaneously, and (2) tips are capable of reading/writing in both the $+y$ and $-y$ directions but data must be read in the direction it was written.

We map disk sectors to probe-based storage as follows. The bits of each sector are striped on the media, so the first bit is accessed by the first tip, the second by the second tip, etc. Thus, the sector is read/written by all the tips of one row in parallel, while the sled moves above it in the $+y$ direction (the origin is the upper left hand corner of the device). Figure 3a illustrates this movement. After one row of tips accesses data along the y-axis, the next tip row is activated, and the sled reverses direction. Figure 3b shows that sector 1 is split between two tip rows, and Figure 3c shows how sector 2 is subsequently accessed in the reverse direction to sector 0. Only after all the rows of tips finish accessing a column of bits will the sled move to the next column, as illustrated by Figure 3d.

5 Models for Probe-Based Storage

To make accurate decisions about data placement and scheduling, we need a good model for the seek time of a probe-based storage device. We examine three models that rely on macroscopic physical principles.

5.1 The Unconstrained Sled Model

We first describe the the model proposed in a generalized form by Griffin *et al.* [6]. We describe the motion of the sled in terms of its position, velocity (or distance traveled per unit time), and acceleration (rate of change in velocity per unit time). The variables for

these quantities are x_0 and x_f (initial and final positions), v_0 (initial velocity), v_f (final velocity), a (acceleration), and t (time).

The unconstrained model assumes we move from x_0 to x_f by accelerating with maximum possible acceleration to the midpoint, decelerating with the same magnitude to x_f. Seeks in x also incur an additional *settling time*, t_{settle}, for the sled oscillations to dampen. Seeks in y do not incur this extra cost because the sled does not come to a full stop. The seek time t_{seek} in x, assuming an initial and final velocity of zero, is

$$t_{seek} = 2\sqrt{\frac{x_f - x_0}{a}} + t_{settle} \tag{1}$$

In practice, and as our results show, this is an aggressive assumption that does not accurately approximate the fully constrained system of Figure 1. Acceleration is not a constant; it increases and decreases with time. In their generalized form of this model, Griffin *et al.* [6] allow for piecewise discontinuous acceleration to accommodate physical reality, but these values of acceleration must be specified as input, rather than determined by the dynamics of the actual physical mechanism.

Because the springs that move the sled are basically independent in each dimension, we treat motion in the x and y dimensions independently; the seek time is then calculated as the maximum seek in each dimension. In the simplified model, the equations are identical, and y_f and y_0 may be substituted for x_f and x_0 in (1), with a settle time of zero, to estimate the seek time in the y dimension.

We have also considered a generalization of the unconstrained model's seek time calculation with nonzero initial and final velocities; seek time increases or decreases slightly depending on initial and final position because the sled might reverse motion once or twice. However, we wish to focus on intrinsic differences of the models, so we consider only the simplified version of all models here. A full derivation of the equations to calculate seek time using the unconstrained sled model may be found in [8].

5.2 The Spring Model

The unconstrained model, while useful for understanding some of the characteristics of probe-based storage, omits important features of the underlying architecture. For this reason we consider a more realistic model, which is based upon the mechanics involved in the sled movement and takes into account restoring, damping, and external forces. The mechanics of sled movement are

$$m\ddot{x} + \lambda\dot{x} + kx = F(t). \tag{2}$$

Equation 2, the well known equation describing the damped oscillator, expresses Newton's Second Law of Motion for the sled: x is the position of the sled at time t, m is the mass of the sled, λ is the damping coefficient, k is the coefficient of elasticity of the restoring force (spring constant), and $F(t)$ is the external force created by the microactuators.

The spring model is characterized by the assumption that the external force $F(t)$ is constant and has magnitude precisely chosen so that the equilibrium position of the sled is the desired final position, viz.

$$x^* = \frac{F}{k} \tag{3}$$

For comparison with the unconstrained model, we assume further that the initial sled velocity is zero, that the initial position is $x_0 = 0$, and that the total distance traveled is x^*. With these assumptions, the general solution of (2) is

$$\begin{aligned} x_t &= C_3 e^{-rt} \cos \omega t + C_4 e^{-rt} \sin \omega t + \frac{F}{k} \qquad (4) \\ C_3 &= -\frac{F}{k} \\ C_4 &= \frac{rC_3}{\omega} \end{aligned}$$

In (4), the inverse relaxation time r and oscillation frequency ω are, respectively,

$$r = \frac{\lambda}{2m} \tag{5}$$

$$\omega = \sqrt{\frac{k}{m} - \frac{\lambda^2}{4m^2}} \tag{6}$$

Equation 4 calculates the position of x at time t. We solve for the time numerically by calculating the dynamics of x and imposing a tolerance on the oscillation as the sled settles. In other words, when the position falls within a specific range, the seek is complete.

5.3 The Optimal Control Model

Although (4) represents the relationships between the sled mass and the forces that operate on it, this model does not describe an important feature of the mechanical system. The spring model uses a constant force just large enough to move the sled the required distance.

Here, we examine the optimal solution of this problem. The force varies over time, and either we are applying the maximum force or we are not. This is called *bang-bang control*, and it is the same kind of control that we use in the basic model, except that now our system has more physical parameters.

We consider the time-optimal control solution of (2) using bang-bang control in (7). Here, K is a positive gain constant, and $u(t)$ is the control.

$$m\ddot{x} + \lambda\dot{x} + kx = Ku(t) \tag{7}$$

Let us set $\lambda = 2\alpha$ and $k = \alpha^2 + \omega^2$.

$$x_1 = \frac{\omega x}{K},\ x_2 = \frac{1}{K}(\alpha x + \dot{x}).$$

We define

$$\mathbf{Z} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},\ \mathbf{M} = \begin{pmatrix} -\alpha & \omega \\ -\omega & -\alpha \end{pmatrix},\ \mathbf{s} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Then (7) becomes

$$\frac{d\mathbf{Z}}{dt} = \mathbf{M}\,\mathbf{Z} + X(t)\mathbf{s} \tag{8}$$

The solution to (8) is derived in [1] and is provided in (9). These equations are transcendental, so we cannot eliminate t; however, we can calculate the time-optimal control as a function of the state $(\frac{\omega^2+\alpha^2}{\omega}x_1, \frac{\omega^2+\alpha^2}{\omega}x_2)$ with the knowledge that the control must switch between the values $\Delta = \pm 1$ and that it cannot remain constant for more than π/ω units of time.

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = e^{-at} \begin{bmatrix} cos\omega t & sin\omega t \\ -sin\omega t & cos\omega t \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \frac{\Delta}{\alpha^2+\omega^2} \begin{bmatrix} \alpha e^{-\alpha t} sin\omega t - \omega + \omega e^{-\alpha t} cos\omega t \\ -\omega e^{-\alpha t} sin\omega t - \alpha + \alpha e^{-\alpha t} cos\omega t \end{bmatrix} \tag{9}$$

We solve (9) numerically similarly to the spring model, by calculating the position as a function of time and changing the control according to the position relative to the switch curve.

Parameter	Description	Value
m	mass	0.1, **0.2**, 0.3 g
F	external force	(0,10,20, . . . , **230**, . . . 500)$\times 10^{-4}$ N
k	spring coefficient	300, **500**, 700 N/m
λ	damping coefficient	0.445, 0.482, **0.626**, 0.743, 0.763 kg/s
ω	resonant frequency	**220** Hz
a	acceleration	**115** m/s^2
t_{tol}	tolerance	10, **25**, 40 nm
t_{settle}	settle time	**0.7** ms

Table 1: Simulation parameter values. Default values are typeset in boldface.

5.4 Transfer Time

In §5.1–§5.3, we have described three models for seek time. This is only one component of total access time.

As we described in §4, the sled moves in the $\pm y$ direction to read the data, activating the next row of tips at every direction change ($T_{turnaround}$). When all tip rows have finished accessing a column of bits, the sled moves to the next column (T_{xmove}). The actual read/write time is a function of the velocity of the sled, and is given by T_{access}. Thus, when x columns are accessed and there are y direction changes, or turnarounds, transfer time is

$$T_{transfer} = T_{access} + x \times T_{xmove} + y \times T_{turnaround} \tag{10}$$

6 Model Comparison

The three models we have described in §5 represent the mechanics of the underlying physical system with increasing accuracy. The unconstrained sled model ignores the effects of damping and the restoring force. The spring model incorporates these parameters, but does not optimally apply the external force. To realize the full potential of a given hardware configuration, we derived the optimal control model. Here we explore the differences between these models and their sensitivity to different parameters.

6.1 Parameter Selection

The unconstrained sled model and both the spring and optimal control models use different parameters, which complicates their comparison. The unconstrained sled model calculates

t_{seek} directly as a function of x^*, a, and t_{settle}. The numerical solution for t_{seek} using the spring model and the optimal control models uses x^*, force F, sled mass m, resonant frequency ω, damping factor λ, spring constant k, and tolerance t_{tol}. In short, the unconstrained sled model approximates the dynamics of the system with acceleration a and a fixed settle time t_{settle}.

However, the two sets of parameters are related by (3), (6), and the relationship between force, mass, and acceleration given in (11).

$$F = ma \tag{11}$$

We selected parameters for our simulations, shown in Table 1, based on values in the literature [16, 10, 11], the theoretical relationships between these parameters, and the physical features of the probe-based storage system. Our parameter selections are chosen to make the two models directly comparable.

First, the sled mass m is approximated based on the size of the sled and the density of polysilicon (2.3 g/cm^3). Assuming a 1 cm $\times$ 1 cm $\times$ 1 mm sled, m is 0.23 g. We study the effect of several reasonable values for m.

Griffin *et al.* [6] use an acceleration a for the probe-based storage device we are modeling of approximately 115 m/s^2. Thus, we can use Equation 11 to calculate a consistent value of F (230×10^{-4}N). For completeness, we vary F in our experiments from 0 N to a maximum of 500×10^{-4} N.

We estimate the spring constant k from the sled's final equilibrium equation (3). Based on the architectural parameters we specified in §3, the maximum length that a sled can move is 100μm. Thus, we can determine reasonable values for k. Assuming F is 230×10^{-4} N and x^* is 50μm, k is 460 N/m. This is one reasonable estimate; however, we determine the range of values used in this experiment by seeing how they affect the seek time. We wish to limit the seek time to several milliseconds at most. If k becomes too small compared to the external force F, the sled will spend a lot of time oscillating and the seek time will be too long. On the other hand, if k is very large, it will take more force to move the sled. Therefore, we vary k in the experiments from 300, 500, to 700 N/m.

Finally, we calculate λ through the following two methods. First, according to (6), λ is calculated from the values of m, k, and ω. Griffin *et al.* [6] state that ω is 220 Hz in their first generation model, which is consistent with a settle time of approximately 0.7 ms. The same frequency of 220 Hz is used for the two models. Second, we can look at the discriminant of Equation 2 when the external force $F = 0$ (i.e., free oscillation). With small damping, i.e., $\lambda^2 - 4mk < 0$, then given $m = 2 \times 10^{-4}$ kg and $k = 500$ N/m, λ should be less than 0.632 kg/s. Both methods suggest that λ should be around 0.6 kg/s or smaller, with the upper limit depending upon m and k. Therefore, for all experiments, λ is calculated from Equation 6.

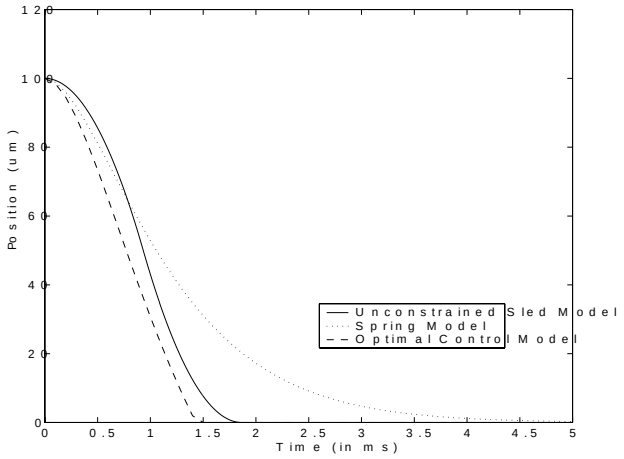


Figure 4: Model dynamics.

6.2 Model Characteristics

Figure 4 shows the position of the sled (x) over time from the initial position 100 μm to the final position of 0 μm (representing the maximum range of movement in this proposed architecture), where the parameters are set to their default values (typeset in boldface in Table 1). The optimal control model calculates the shortest trajectory, the unconstrained sled model approximates this path, and the spring model calculates the longest seek time.

A key difference in these models may be understood by examining the sled acceleration over time as shown in Figure 5. In the unconstrained sled model, acceleration assumes one of two values, $\pm a$. In the spring model, acceleration varies continuously to reflect the behavior of the sled as a constant force is applied to counteract damping and restoring forces. Note that the damping is close to critical damping, so oscillation is minimal. The optimal control model applies maximum force to the sled, switching it at the perfect moment to drive it to the destination in minimum time.

Figure 6a plots the seek time in x estimated by the two models against the number of 512 byte sectors traversed in multiples of 10,000, beginning at sector 0. Behavior is consistent with model dynamics described in Figure 4; the spring model is overly conservative, the unconstrained sled model much less conservative, and the optimal control model is naturally a lower bound on what can be achieved with the physical description of the system. The unconstrained sled model overestimates optimal seek time by approximately a factor of two. Figure 6b is a detail of the time to seek from sector 0 to sectors 3000–4000 for the unconstrained sled model and the optimal control model. The periodicity of seek times is a function of data layout; recall from §4 that the sled moves in the $\pm y$ direction, switching

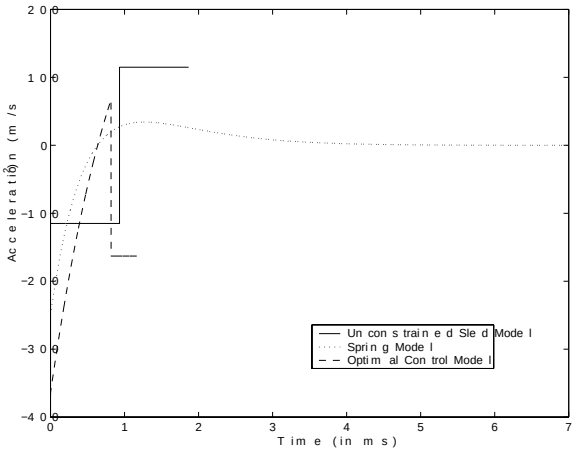


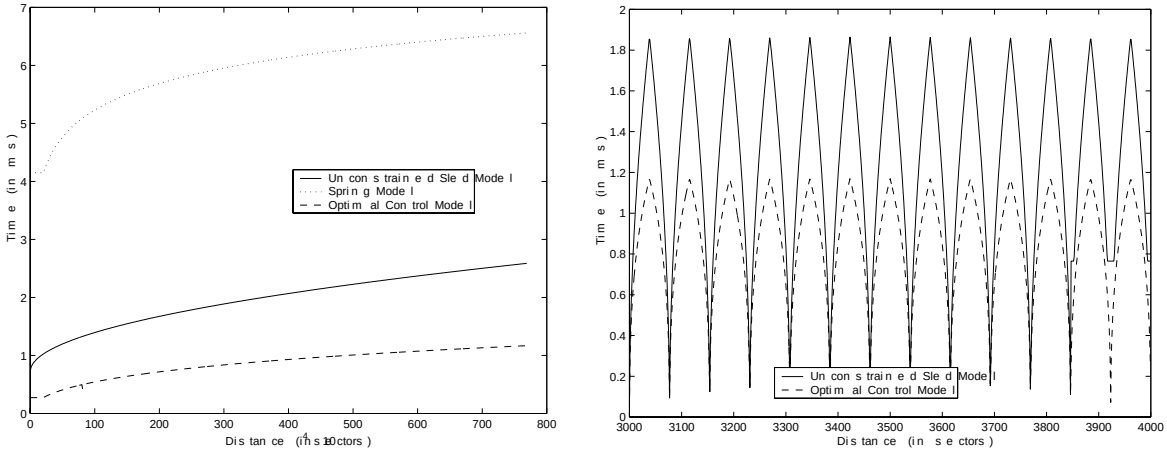
Figure 5: Acceleration over time.

tips at each direction change. Seeks within the first 3846 sectors lie within the first column, and require no movement in the x direction. When an x seek is required, the unconstrained sled model includes the settle time t_{settle}, which is 0.7ms. Both the spring and optimal control models calculate t_{settle} implicitly as a function of the system parameters.

We illustrate the impact of the sled mass m on the seek time calculated by the unconstrained sled model and the optimal control model in Figure 7. In all models, if all other parameters are fixed, the heavier the sled, the greater the seek time.

In the optimal control and spring models, we can directly represent changing values of λ and k. The impact of different values of the spring constant k upon seek time in the spring model is shown in Figure 8. The parameter λ must increase when k increases to maintain a constant resonant frequency (ω) of 220 Hz, according to (6). When k=700 N, the maximum distance the sled can move is less than 100 μm because the maximum force is 500 N. Intuitively, as we make the spring more resilient and increase the damping factor, we can expect seek times to decrease; the sled will come to a stop sooner.

Independent of the model used, seek times are small relative to transfer times. Figure 9 shows the time to read randomly distributed sectors of different sizes. Transfer time dominates seek time predicted by the unconstrained sled and optimal control models for a sector size of 4KB. This tradeoff is important to understand to choose appropriate transfer sizes and striping factors for probe-based storage arrays.



(a) (b)

Figure 6: Seek times using different models.

6.3 Experimental Results

To study the impact of more detailed physical modeling on an actual workload, we have implemented these models within the Pantheon [18] simulator and tested them against real workloads. The workloads we have chosen come from two different Hewlett-Packard systems, **snake** and **cello**, and are described in [14]. Snake is a file server for an HP-UX cluster, and cello is a timesharing system. From snake we examine the traffic to the /usr1 partition, and from cello we examine the traffic to the news disk. These workloads were chosen because they represent a range of sequentiality; 4% of accesses on cello news are sequential, whereas 38% of accesses on snake /usr1 are sequential. Because the original disks have smaller capacities than the probe-based storage device we are describing, we modified the trace to represent a seek pattern that scanned the entire device by scaling the sector size without increasing the transfer time. The result is an approximation to what might happen if the file system policies used in these systems were used directly on a probe-based storage device of similar capacity.

Figure 10a and 10b show the cumulative distribution function of response times estimated using the different models for the cello news and snake /usr1 workload, respectively. All requests are serviced in the order they arrive. We can see that all models have different behavior for real workloads, although the optimal control and spring models have similar, step-like shapes. The unconstrained sled model is not a close approximation to the optimal control model. As expected, the response times for the more sequential workload, snake /usr1, are generally lower than that of the cello news disk.

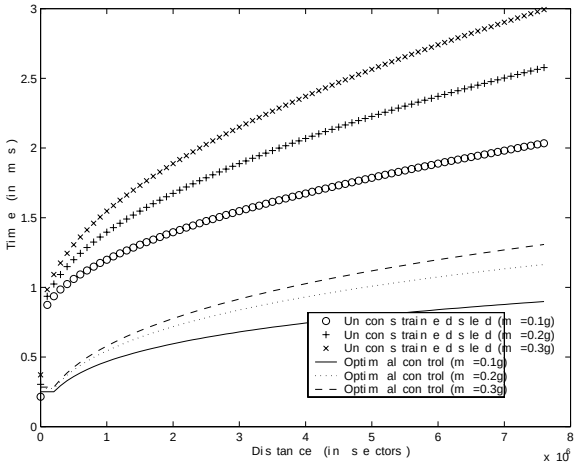


Figure 7: Changing the sled mass.

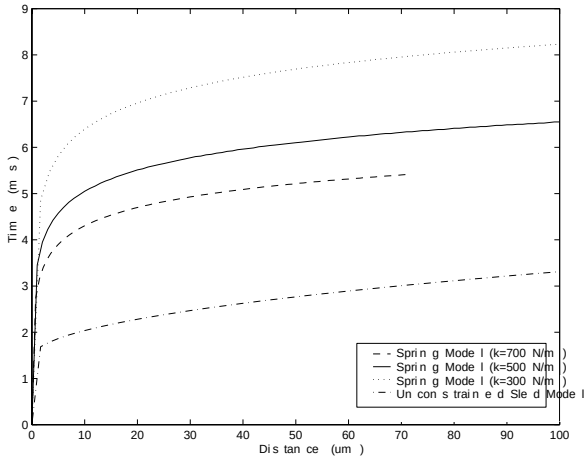


Figure 8: Changing the spring constant.

6.4 Summary

We have compared the dynamics and seek times of the unconstrained sled model, the spring model, and the optimal control model. The spring model and optimal control model more closely represent underlying mechanics of probe-based storage and have more parameters specifically describing those mechanics. With consistent physical parameters, seek times estimated by the spring model are an upper bound and the optimal control model provides a lower bound. The unconstrained sled model is an approximation to the lower bound that uses a single parameter, acceleration, to describe the more complicated spring system.

7 Conclusions

Probe-based storage technologies share the common characteristics of being dense, highly parallel, and using rectilinear two-dimensional motion. Models that bridge the gap between the physics of the device and its performance characteristics provide important feedback to hardware and software designers. We have proposed two new models for seek time for probe-based storage: the spring model and the optimal control model, and compared them with the unconstrained sled model [6]. These models convey successively more accurate descriptions of the underlying physics of the device. We conclude that the physical accuracy of the model has a significant impact on the predicted performance of real workloads. Furthermore, the optimal control model and spring model give upper and lower bounds on the I/O performance, and may be used to optimize architectural parameters for specific I/O

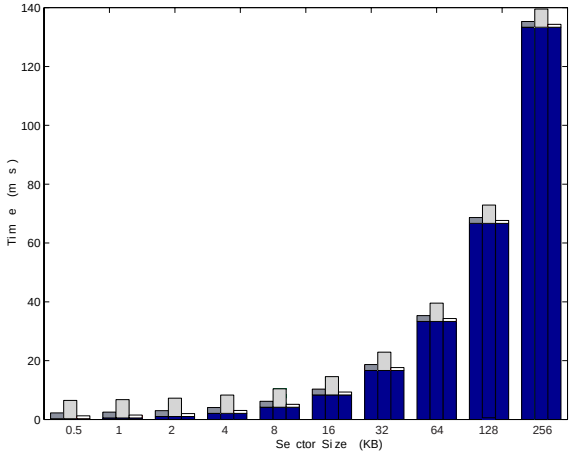


Figure 9: Time to read randomly distributed sectors of different sizes.

workloads.

We are currently extending the optimal control model to accommodate a broader range of parameters, including nonzero initial and final velocities and overdamped systems.

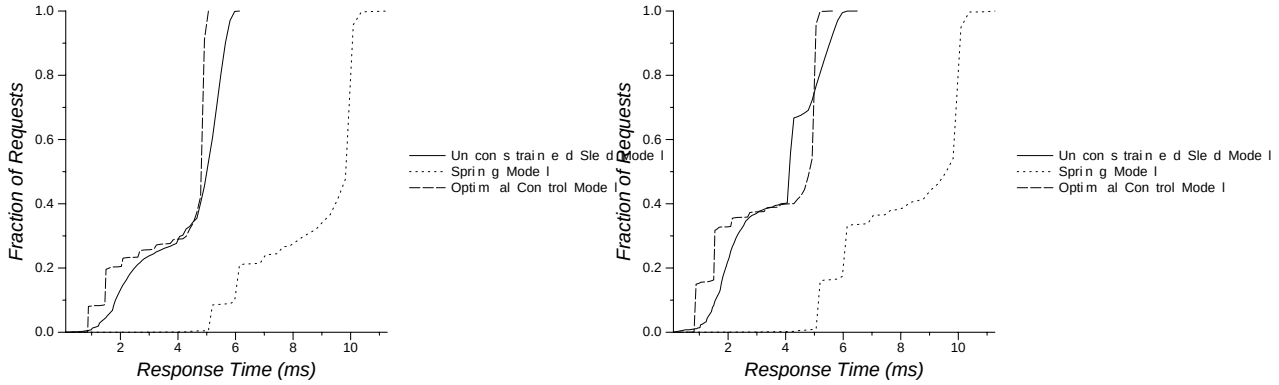


Figure 10: Seek times for (a) cello news, and (b) snake /usr1 using different models.

Acknowledgements

We gratefully acknowledge Miriam Sivan-Zimet for creating and testing the layout modules used in the Pantheon simulator for this research. We thank Ali Shakouri for his help with the physics of the spring model, and Joel Yellin for helpful discussions of the control theory aspects of this work.

References

[1] M. Athans and P. L. Falb. *Optimal Control: An Introduction to the Theory and Its Applications*. McGraw-Hill, 1966.

[2] C. Brown. Microprobes promises a new memory option. *E.E. Times*, 1998.

[3] L. R. Carley. www.chips.ece.cmu.edu, 1999.

[4] M. Despont, J.Brugger, U. Drechsler, U.Durig, W. Haberle, M. Lutwyche, H. Rothuzen, R. Stutz, R. Widmer, H. Rohrer, G. Binnig, and P. Vettiger. VLSI-NEMS chips for AFM data storage. In *Technical Digest. IEEE International MEMS 99 Conference. Twelfth IEEE International Conference on Micro Electro Mechanical Systems*, 1999.

[5] G. R. Ganger, B. L. Worthington, and Y. N. Patt. The disksim simulation environment version 1.0 reference manua l. Technical Report CSE-TR-358-98, University of Michigan, Ann Arbor, Feb. 1998.

[6] J. L. Griffin, S. W. Schlosser, G. R. Ganger, and D. F. Nagle. Modeling and performance of MEMS-based storage devices. In *Proceedings of ACM SIGMETRICS 2000*, 2000.

[7] G.T.Sincerbox and ed. Selected papers on holographic storage, vol.ms 95 of SPIE milestone series. In *Internal Society for Optical Engineering*, 1994.

[8] T. M. Madhyastha, P. Yang, and M. Sivan-Zimet. Physical modeling of probe-based storage. Technical report, University of California, Santa Cruz, April 2000.

[9] H. J. Mamin, B. D. Terris, L. S. Fan, S. Hoen, R. C. Barrett, and D. Rugar. High-density data storage using proximal probe techniques. In *IBM Journal of Research and Development*, 1995.

[10] M. Mehregany, W. H. Ko, A. S. Dewa, and C. C. Liu. Introduction to MEMS systems and the multiuser mems processes. Technical report, Electronics Design Center, Department of Electrical Engineering & Applied Physics, Case Western Reserve University, 1993.

[11] T. Mukherjee, S. Lyer, and G. K. Fedder. Optimization-based synthesis of microresonators. In *Sensors and Actuators A*, 1998.

[12] D. Psaltis and G.W.Burr. Holographic data storage. In *IEEE Computer*, 1998.

[13] S. Redfield and J. Willenbring. Holostore technology for higher levels of memory hierarchy. In *Eleventh IEEE Symposium on Mass Storage Systems*, 1991.

[14] C. Ruemmler and J. Wilkes. Unix disk access patterns. In *Proc. of the Winter'93 USENIX Conference*, pages 405–420, 1993.

[15] C. Ruemmler and J. Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27(3):17–28, Mar. 1994.

[16] S. Schlosser, J. Griffen, D. Nagle, and G. Ganger. Filling the memory access gap: A case for on-chip magnetic storage. Technical Report CMU-CS-99-174, Carnegie Mellon University, 1999.

[17] J. W. Toigo. Avoiding a data crunch. *Scientific American*, 282(5):58–74, May 2000.

[18] J. Wilkes. The Pantheon storage-system simulator. Technical Report HPL-SSP-95-14, Storage Systems Program, Computer Systems Laboratory, Hewlett-Packard Laboratories, Palo Alto, CA, May 1996.