Dynamic Simulation as a Design Tool for a Microactuator Array

Dan Reznik* Stan Brown John Canny
{dreznik,swb,jfc}@cs.berkeley.edu
Dept. of EECS, UC-Berkeley, Berkeley, CA 94720 USA

***Abstract*—**
We use dynamic simulation to optimize the design of an existing micro-electromechanical (MEM) device, called the Manipulation Chip (M-Chip). This device contains an excess of 10,000 moving actuators, called resonators, which oscillate torsionally at a few kHz. Parts dropped on the chip's surface are conveyed towards a unique direction. Given the enormous number of moving parts, it is impractical to attempt to measure the device's (or part's) dynamic state during a manipulation task. Yet, knowing this information is crucial for redesign and optimization. We make use of a powerful dynamic simulation tool, called "Impulse", to generate synthetic measurements over a range of experiments. From these results, we suggest redesign options which debug existing problems and improve the feed rate. The array is found to behave similar to a viscous spring-loaded conveyor belt; most of its energy is spent on driving the part vertically, calling for a more efficient design.

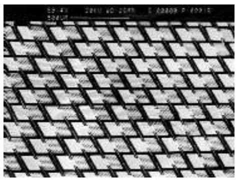
I. Introduction

A recent trend in robotics has been the design and fabrication of micro-electromechanical (MEMs) active surfaces intended for part manipulation at the sub-mm scale. Böhringer et al. [1] have fabricated one such device called the *M-Chip* ("M" stands for manipulation), shown in Figure 1(a). This device consists of an array of approx. 10,000 micro-actuators, called *resonators*, tiled over a few square-cm of silicon substrate. Resonators are rectangular slabs of silicon supported a few μm above the substrate by torsional *rods*. Electrostatic actuation cause resonators to oscillate about the rods at a few kHz. One end of the resonator is equipped with a *ridge* of several vertical *poles*. The asymmetric design generates anisotropic impact forces when a small *part* is placed over the array, inducing a motion bias towards a specific direction [1].

In one version [1] of the device, called the *unidirectional array*, resonators are all oriented towards a single direction and tiled in interleaved fashion, as shown by the electron micrograph in Figure 1(b). This configuration acts as a microscale parts feeder for light objects dropped on its surface – these are transferred from one end of the array to the other at a constant speed ν called the *feed rate*.

*Support provided in part by NSF Grant FD93-19412.
[1] Another version of the device, called the *squeeze array*, consists of two opposing unidirectional arrays connected to each other along a center line. This device has been proposed for sensorless part orientation [2].

(a)



(b)

Fig. 1. (a) A photo of Böhringer's hand holding his M-Chip, containing approx. 10,000 resonators. (b) Electron micrograph of a portion of the array showing the interleaved tiling of resonators.

One of the main difficulties in testing and re-engineering the M-Chip is obtaining meaningful experimental data given the small dimensions and the sheer number of on-chip moving actuators. The fact that the dynamic state of the device and/or part is practically inaccessible makes it difficult for one to identify design changes which could lead to better performance. This type of problem is analogous to that faced by a semiconductors designer who needs detailed measurements on the operation of a new type of device with the intent of improving its performance. Such measurements will be often too time consuming and/or noisy, so an alternative is simulation tools such as FEM or SPICE. Here we propose a similar solution. Our contribution is to utilize a powerful dynamic simulation tool called *Impulse* [3] to extract detailed, noise-free dynamic information from the array over a variety of experiments. In designing dynamic simulations for the M-Chip, we have as principal goals:

1. Design verification and debugging: does the device correctly feed parts placed on its surface and at which rate? If not, can problems be identified and solutions proposed which are successful at least in simulation?

2. Performance optimization: for a working prototype, are there changes in the design which could lead to better performance?

3. Device modeling and critique: can the chaotic part-array interactions be reduced to a simplified model which captures the bulk behavior of the device's dynamics?

These questions are addressed in the following sections.

A. *Related Work*

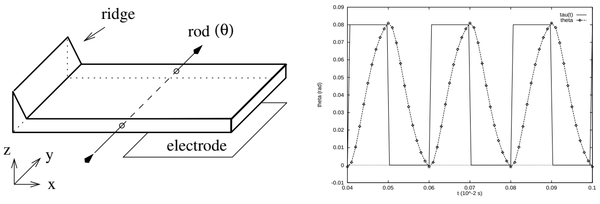
One way to study part motion is to abstract the array into a motion field. This simplification has been used to obtain qualitative results on part reorientation for a version of the M-Chip called the squeeze array [2]. However, no attempt has yet been made to simulate part-array interactions quantitatively at the detailed mechanical level, owing mainly to the lack of efficient, accurate dynamic simulation tools. The Impulse tool [3], recently made available to us, overcomes these problems by providing a model for collision detection and resolution, which is particularly well suited for the types of rigid vertex-face interactions likely to occur between the part and the resonators.

Previous work done using Impulse to collect statistical data to characterize a complex mechanical process was done in the context of estimating pose statistics for polyhedral parts dropped from random orientations on a flat surface [4]. From a design optimization standpoint, the work closest to ours is that of Berkowitz and Canny [5, 6] who used Impulse to optimize the design of a passive parts-orienting device. From a modeling standpoint, Boothroyd [7] has worked on abstracting the performance of vibratory bowl feeders with respect to part mass, part geometry, friction, and oscillating frequency of the device. A related and emerging line of work involves simulation-based optimization of part shapes targeted for thermodynamic self- assembly [8, 9].

The remainder of this paper is organized as follows: in Section II the simulation setup is explained. In Section III we describe how both the jamming and double bounce problems were eliminated, and also performance studies which identify an optimal resonator shape. In Section IV we develop an abstract model for the device and study the energetics of part motion. Section V concludes with a summary and directions for future work.

II. Simulation Setup

In modeling the M-Chip with Impulse, the focus was to preserve original nominal parameters and dimensions as closely as possible. We used the original resonator mass, geometry, and oscillation frequency as given to us by the designers. We started out by creating a 280x180x5 μm geometric model for the resonator, shown in Figure 2(a). To simplify collision detection, we modeled the set of poles installed on one extreme of the resonator collectively as a single 5 μ-high ridge. The resonator body[2] was modeled as a rectangular slab. In the actual device, resonator motion is induced by applying a voltage between the resonator and an electrode underneath it, as shown in Figure 2(a). Electrostatic actuation was replaced by (i) modeling the supporting rods as a single spring-loaded revolute joint, and by (ii) a torque control-law acting at that joint. The joint's spring and damping constants were chosen according to the torsional elasticity of silicon. The torque wave applied to the joint is a 5 kHz positive square wave, whose amplitude was chosen[3] to produce positive oscillations of approx. 5^o (0.08 rad). Figure 2(b) shows the driving torque law superimposed on θ, the free oscillation angle, which is roughly a 5 kHz positive sinewave (the resonator acts as a mechanical low pass filter which attenuates the higher frequency harmonics of the driving square-wave).



(a) (b)

Fig. 2. (a) The resonator is modeled as a rectangular slab with a ridge at one end. The supporting rods are modeled as a spring-loaded, damped revolute joint. The electrostatic actuation (provided by the electrode) is replaced by a torque control law at the joint. (b) The torque applied to the resonator is a 5 kHz positive square wave. The torque (not shown in scale) is superimposed with $\theta(t)$, the resonator angle. The resulting oscillation is a positive 5 kHz sinewave.

The next modeling step was to assemble the resonators in interleaved fashion, as in Figure 1(b). Though the actual device contains over 10,000 resonators, simulating this many moving parts is impractical with Impulse. However, by considering an 8x5 sub-array of resonators, and a rectangular array-aligned part with silicon's density, we are can remain physically consistent with the full size experiment. The final Impulse model shown as a 3D rendering is depicted in Figure 3. Notice that the part's footprint covers approx. 10 resonators, while in actuality it would cover 100s.

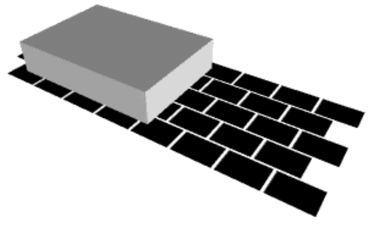


Fig. 3. 3D rendering of the complete model showing a 1.15x0.4x0.25 μm, 550 μg part resting at its initial position over an 8x5 resonator array.

The Impulse simulator resolves collisions between rigid bodies based on a sophisticated discrete-event impulse model. Two global parameters need to be set: the friction coefficient μ, and the restitution coefficient ϵ. These were set to the physically

[2] Fabricated resonators posses a grating of holes, but these have no effect on collisions with the part.

[3] 1 kHz and 2^o are the values nominally used by the designers – w.l.o.g., we chose 5 kHz and 5^o for better numerical stability with Impulse.

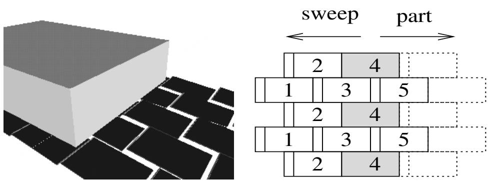
reasonable (and numerically-stable) values of 1.0 and 0.5, respectively. Experiments to evaluate how these parameters affect array performance have not yet been tried, though some theoretical results have been derived for vibratory feeders [7].

III. DESIGN OPTIMIZATION

The first simulated experiment performed with the array was to drive the resonators and simply drop the part on it. Figure 3 shows the part at its starting position for this experiment.

A. *Part jamming*

Once dropped on the array, the part is propelled forward at a constant speed of about 0.8 mm/sec, a value in close match with experimental results performed by the designers. A problem quickly discovered was that the part would jam (i.e., stop its forward motion) as soon as it encountered a new row of resonators, as shown in Figure 4(a). Jamming occurs since the part's vertical hopping is not high enough to allow it to skip over the ridges of the next row of resonators. As the part is driven up, the next row of resonators is also doing so, and the part bounces back.



(a) (b)

Fig. 4. The jamming problem: in (a) the part is shown unable to skip over the ridges of the encountered row of resonators. In (b) the row-strobing method is illustrated. Consecutive rows in the array are labeled from 1-8 in the direction of the part's motion (only the first 5 rows are shown). The process involves selectively turning off consecutive rows of resonators for a few oscillation cycles, in the direction opposite to the part's motion.

The first attempted solution to eliminate jamming was to drive consecutive resonator rows at different phase offsets, so that ridges in the blocking row would be going down while other resonators would be driving the part up. This idea did not work since the out-of-phase impacts occurring under the part cause lower part hops which in turn aggravate the problem of skipping a set of blocking ridges. A solution which proved successful is called *row-strobing*, illustrated in Figure 4(b). The idea is to propagate a wave of off resonators in the direction opposite to the part's motion. Label all rows from 1 to N along the array's feeding direction. The first row to be turned off is row N. That row is left in the off state during γ complete oscillation cycles, at which point it is turned back on. The process continues with row $N-1$ being off for γ cycles, and so on until row 1 is reached. At this point, the wave wraps around and the process restarts at row N. Since part position is not known, any resonator row is potentially causing jamming, thus the need for a sweeping wave (this could be alleviated with sensing). To ensure that the jamming row is located faster, the wave is propagated against the feeding direction so the relative speed between the wave and the part is higher. Choosing too small a γ may not give the part enough time to be pushed over the blocking ridges, however the average jam clearing time is proportional to γ. We found that $\gamma = 50$ oscillation cycles gave the best results. In the case of a very long array (as in the real device) the blocking row can be found faster by propagating several off wavefronts separated by a constant number λ of resonator rows. If λ is too small, too many off rows will lie under the part at any given time, reducing the feed rate considerably. In our case we set $\lambda = 4$ rows, i.e., in our 8x5 array there will be, at any given time, two off rows moving in the $-x$ direction. Since the part is about 4 resonators long, the part receives $1/4$ less impacts at any given time.

As shown in Figure 5, the row-strobing method results in a motion of the part characterized by periods of constant forward feed rate (of approx. 0.8 mm/sec) interleaved with short jamming phases, where the feed rate is null.

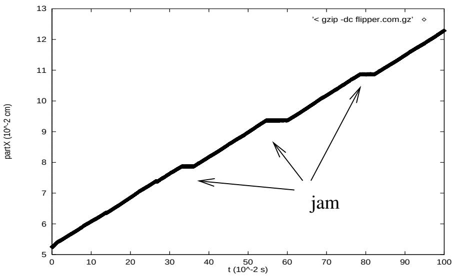


Fig. 5. Plot of the part's forward displacement as a function of time. As shown, the part's motion alternates between constant forward motion and short periods of jamming.

B. *Double bouncing*

The simulator treats collisions as discrete events; each collision is followed by a calculation which computes an impulse force applied at the collision point and in opposite direction to the bodies colliding. We looked at the stream of collisions between the part and the resonator located on the 2nd row and 3rd column of the array during the first non-jamming period of Figure 5, namely, for $0.05 < t < 0.2$ sec. For every such collision we recorded the associated (i) resonator angle θ, (ii) angular speed $\dot{\theta}$, and (iii) F_x, the x component (i.e., along the feeding direction) of the impulse force calculated by the simulator. Figure 6 shows a scatter plot of the $(\theta, \dot{\theta})$ pairs gathered. These pairs cluster into two separate clouds pointed to by the arrow labeled *loaded*. As expected, one of the clouds (signaled by the *free* arrow) follows quite closely the θ vs. $\dot{\theta}$ relation for the resonator's free oscillation – this is an ellipse since the free oscillation is roughly sinusoidal. However, the second cluster of points is anomalous.

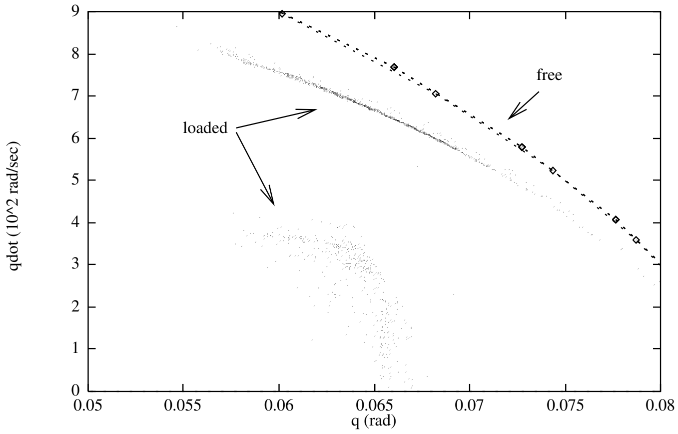


Fig. 6. Scatter plot correlating resonator angle θ (x-axis) w/ the angular speed $\dot{\theta}$ (y axis). The *free* arrow shows this relationship for free oscillation – roughly an ellipsoid. The *loaded* arrow points to $(\theta, \dot{\theta})$ pairs generated by individual collisions when the part is over a chosen resonator. Some of the loaded points follow the free oscillation curve, while others lie in an anomalous region of low values of $\dot{\theta}$.

What phenomenon could be generating the lower cloud of points? That cluster indicates that many collisions are occurring at an angle θ with a much slower $\dot{\theta}$ than that of free oscillations. We found that these collisions were being caused by secondary bounces of the resonator on the part occurring shortly after a normal 5 kHz bounce. To show that, we correlated θ, $\dot{\theta}$, and F_x with the collision's *inter-arrival time*, which measures the time elapsed between the current collision and the one last occurring in the simulation. This is shown in Figure 7(a,b,c). Though one expects collisions with a single resonator to be spaced by $1/5$ kHz $= 0.2$ ms (i.e., the part receives one impact per resonator cycle), the graphs show that collisions cluster over $.02$, $.18$, and $.2$ ms interarrivals. The collisions occurring at $.02$ ms after the normal $.2$ ms ones are *double bounces*; after the first collision, the ridge is still being driven upward by the torque control law, causing the resonator to ricochet one or more times against the part. After the double bounce, the resonator will tend to re-synchronize with the driving square wave, so that the next collision occurs within $.18$ sec: $0.18 + 0.02 = 0.2$ ms. The (a) plot shows that the $.02$ collisions occur at lower values of θ than the normal $.2$ ms ones. This indicates that a downward motion of the part favors the double bounces. The (b) plot shows that double bounces occur at much slower $\dot{\theta}$ than normal bounces, implying that the former transmit less impact energy to the part than the latter. The (c) plot shows that most double bounces are associated with an impulse pointing in the negative feeding direction, i.e., they act as brakes!

Double bounces were eliminated by reducing the *duty cycle* of the square wave torque driving the resonators from 50% to 36%. This shuts off the driving torque law just before a double bounce is likely to occur (i.e., $.02$ ms after the average time normal collisions occur). The results of this change are shown in

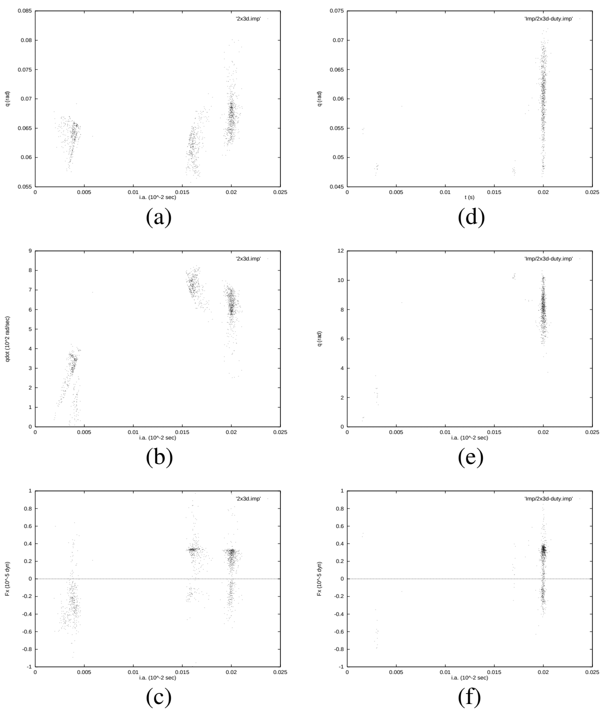


Fig. 7. Graphs (a,b,c) show θ, $\dot{\theta}$, and F_x (plotted along y) clustering over three specific collision inter-arrivals (plotted along x): .02,.18, and .2 ms. Graphs (d,e,f) show the re-clustering of the same quantities, this time solely over 0.2 ms inter-arrivals, when a shorter duty cycle is used.

Figure 7(d,e,f), which side by side with the previous plots illustrate how the cloud of $.02$ and $.18$ ms collisions coalesce into a single cluster over 0.2 ms inter-arrivals. These graphs also show that the new impulses occur in average at a higher value of $\dot{\theta}$, i.e., they transfer more momentum at every collision. This simple reduction in duty cycle increases the feed rate from 0.8 to 1.0 mm/sec, i.e., a 30% improvement. As an interesting note, the points in Figure 7(f) are split evenly in the positive and negative F_x range, indicating that the total force applied to the part over the period considered is zero – the part is feeding forward at a constant speed.

C. *Optimizing the resonator's shape*

The feed rate ν was measured against three parameters: (i) the part's mass M, (ii) the ridge's distance L from the resonator's midpoint, and (iii) the ridge height H. The last two parameters are illustrated in Figure 8.

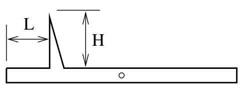


Fig. 8. The shape of the resonator is parametrized according to ridge distance L, and ridge height H.

The plots in Figure 9 show the results of these experiments.

Plots (a,b,c) show, respectively, results from the mass, ridge distance, and ridge height experiments. The x-axis labels the parameter being varied; the feed rate is plotted along y. The collision angle θ, angular velocity $\dot{\theta}$, and the height of the part's center of mass PartZ (averaged over an entire feeding task) are superimposed over the feed rate; their numeric values have been omitted for the sake of clarity.

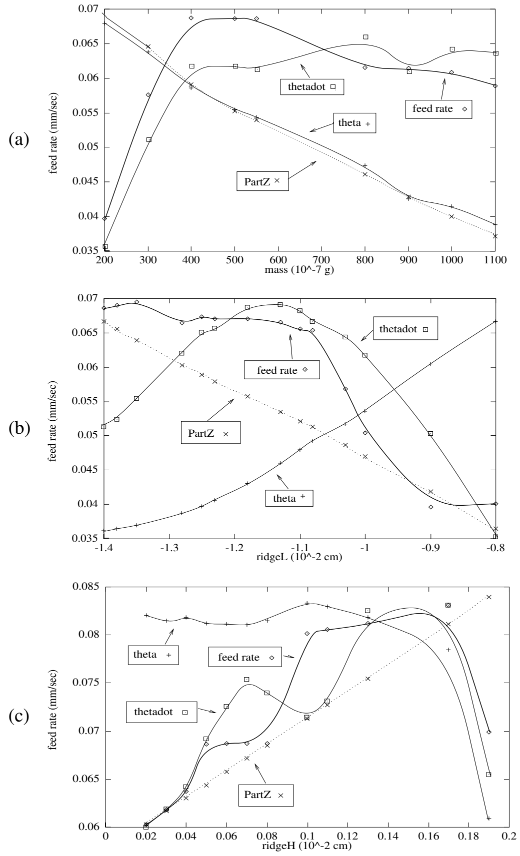


Fig. 9. Results of the performance studies. Each graph's y-axis presents simultaneously ν, θ, $\dot{\theta}$, and PartZ (averaged over a feeding experiment) plotted against the parameter being varied. For (a), (b), and (c), the parameters are, respectively, M, L, and H. The y-axis is labeled in feed rate units, numeric values for the other quantities have been omitted for simplicity.

Figure 9(a)'s optimal ν at $M = 55\ \mu$g is reached since (i) θ is monotonically decreasing (as it gets heavier, the part "sinks" into the array, tracking PartZ) and (ii) $\dot{\theta}$ tends to level off (both these effects contribute to less momentum transferred on the $+x$ direction).

For Figure 9(b), θ increases with L by a simple lever-like effect (ridge gets closer to revolute joint). This effect explains a monotonically decreasing PartZ, since as the ridge gets closer to the center, it transfers less momentum at every impact. These two opposing trends cause the $\dot{\theta}$ curve to go through a maximum at approx. $L = -125\ \mu$m, which in turn causes ν to reach its highest value shortly after $\dot{\theta}$'s maximum.

The first obvious fact in Figure 9(c) is that PartZ moves linearly up with an increase in the ridge height. For the lower region of H values, θ remains constant while $\dot{\theta}$ increases monotonically, and so does ν. At $H = 16\ \mu$m both θ and $\dot{\theta}$ decrease sharply, also decreasing ν. At present we haven't been able to justify what are the geometric/dynamic reasons for this effect.

IV. Device Modeling

A. Array ballistics

The plot in Figure 9(a) shows a linear relationship between M and the PartZ, i.e., the array acts as a linear spring over which the part bounces. This linear relationship yields a spring constant $k = 294$ kdyn/cm. For $M = 55\ \mu g$, this mass-spring system resonates at 370 Hz. We computed the Discrete Fourier Transform (DFT) of PartZ for $0.05 < t < 0.2$ sec, and found two major frequency components: one at 360 and one at 5 kHz, corresponding, respectively, to the spring-like oscillations, and the normal resonator impacts. The DFT also revealed that the 360 Hz component was 5 times larger than the 5 kHz one, suggesting that too much energy is being spent in the upward vibrational mode (more on this below).

B. Friction modeling

To understand the type of friction experienced by the part as it "slides" on the array, we performed the following experiment. A 55 μg part is dropped on the array as in Figure 3. Enough time is waited so that the part achieves its equilibrium feed rate of 1.1 mm/sec. At that point ($t = 0.05$ sec), an external force F_{ext} in the $-x$ direction is applied to the part's center of mass. The experiment consists in observing the resulting feed rate ν' for different values of F_{ext}. Figure 10 shows the part's x position vs. time, before and after F_{ext} is applied – this graph reveals a linear relationship between the new feed rate and F_{ext}, i.e., the array acts as a *viscous* medium, akin to a fluid! This effect motivated us to look at this problem in depth, but for lack of space we direct the reader to another publication [10]. In a nutshell, it is caused by the velocity-independent coulomb frictional force combined with the temporally asymmetric stream of part-resonator impacts.

The viscous model prescribes a frictional force proportional to (i) the part's weight, and (ii) the difference between part velocity and normal feed rate, i.e., $F_{fric} = \mu M g(\nu' - \nu)$, where μ is the the coefficient of static friction. This relation was used to compute values for μ for various externally applied forces, noting that at equilibrium $F_{fric} = F_{ext}$. The results are tabulated in Table I. As shown, the model assumption explain the data quite well, with μ nearly independent of F_{ext} and approx. equal to Impulse's global coefficient of friction.

C. Energetics of part motion

We measured the average energy present in each of the part's degrees of freedom: three translational kinetic energies along X, Y, and Z, denoted LinX, LinY, and LinZ; three angular

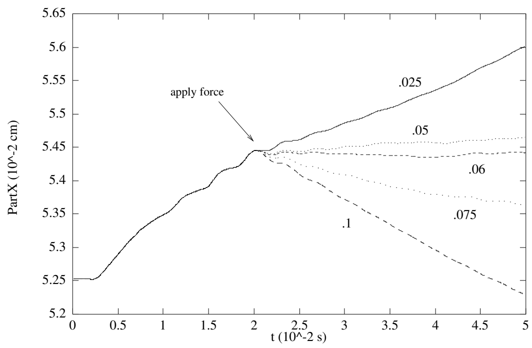


Fig. 10. The change in ν for different values of F_{ext}. After the force is applied, the part reverts to a new constant velocity ν' linearly related to F_{ext}, as in a viscous fluid.

F_{ext} (dyn)	ν' (mm/sec)	μ
0	1.1	N/A
.025	.6	1.16
.05	.06	.99
.06	0	0.93
.075	-.36	1.02
.1	-.88	.99

DOF	E_{avg} (10−5 dyn cm)	Stdev
LinX	1.78	.71
LinY	.1	.15
LinZ	1.38	1.86
AngX	.65	.86
AngY	.63	1.28
AngZ	.0081	.045
PotZ	50.1	14.3

TABLE I
LEFT: FRICTION COEFFICIENT (μ) COMPUTED USING A VISCOUS FRICTION MODEL, FOR VARIOUS EXTERNALLY APPLIED FORCES. RIGHT: ENERGIES AND STANDARD DEVIATIONS PRESENT IN EACH OF THE PART'S INDEPENDENT DOF'S, OVER $0.05 < t < 0.2$ SEC.

kinetic energies about the X, Y, and Z axes, denoted AngX, AngY, and AngZ; the part's potential energy PotZ (with $g = 981$ cm/sec^2) with respect to its resting height. The values for these energies averaged over an $0.05 < t < 0.2$ sec (and the corresponding standard deviations) are shown in Table I.

The above data shows that this type of array consumes a disproportionate amount of energy to keep the part at an average potential energy, rather than for forward motion. This suggests that an array with longitudinal rather than vertical actuators would be more energy efficient (e.g., see [11] for a novel type of design based on thermally-actuated cilia). Notice also that a small portion of the energy is *equipartitioned* between AngX and AngY, implying that part motion along these DOF's is pretty much chaotic. The low value of AngZ shows that the array (as expected) is unable to accelerate the part about the Z axis.

D. An abstract model

From the above studies, the following simplified model of part/array dynamics can be derived: the array acts as a springy conveyor belt over which the part hops. The average part height (how much it sinks into the array), denoted d, is a function of the part's mass M and the supporting spring's stiffness k. Friction with the conveyor belt is viscous with coefficient μ. The belt feeds at a rate ν, also a function of d. This model is depicted in Figure 11.

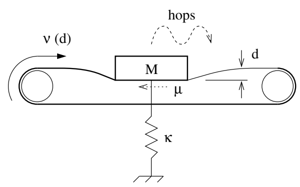


Fig. 11. The array acts as a spring-loaded conveyor belt over which the part hops. The friction is viscous with coeff. μ. The average part height d is a function of the part's mass M and the supporting spring's stiffness k. The belt's feed rate ν is a function of d.

V. CONCLUSION

In this paper we show how dynamic simulation can be used as an effective tool in the characterization an further design optimization of an existing MEMs device called the M-Chip. We are planning to incorporate into the M-Chip's design the various improvements suggested by this work, and find out whether the gains in performance are indeed possible.

REFERENCES

[1] K. Böhringer, B. Donald, and N. MacDonald. Single-crystal silicon actuator arrays for micro manipulation tasks. In *IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, San Diego, CA, February 1996.

[2] K. Böhringer, B. Donald, R. Mihailovich, and N. MacDonald. Sensorless manipulation using massively parallel microfabricated actuator arrays. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.

[3] B. Mirtich and J. Canny. Impulse-based simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*, New York, NY, 1995. ACM Press.

[4] B. Mirtich, Y. Zhuang, K. Goldbergand J. Craig, R. Zanutta, B. Carlisle, and J. Canny. Estimating pose statistics for robotic part feeders. In *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996.

[5] D. Berkowitz and J. Canny. Designing parts feeders using dynamic simulation. In *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996.

[6] D. Berkowitz and J. Canny. A comparison of real and simulated designs for vibratory parts feeding. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[7] G. Boothroyd. *Assembly automation and product design*. Marcel Dekker, Inc., New York, NY, 1991.

[8] M. Cohn, C. Kim, and A. Pisano. Self-assembling electrical networks: an application of micromachining technology. In *IEEE Micro Electro Mechanical Systems*, 1991.

[9] J. Krishnasamy, M. Jakiela, and D. Whitney. Mechanics of vibration – assisted entrapment with application to design. In *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, May 1995.

[10] D. Reznik, J. Canny, and K. Goldberg. Analysis of part motion on a longitudinally vibrating plate. In *International Workshop On Intelligent Robots and Systems*, Grenoble, France, September 1997.

[11] J. Suh, S. Glander, R. Darling, C. Storment, and G. Kovacs. Combined organic thermal and electrostatic omnidirectional ciliary microactuator array for object positioning and inspection. In *Proc. Solid State Sensor and Actuator Workshop*, Hilton Head, NC, June 1996.