THEME ARTICLE

Distributed MEMS: New Challenges for Computation

ANDREW A. BERLIN
Xerox Corporation
KAIGHAM J. GABRIEL
Defense Advanced Research Projects Agency

How do you program a cloud of dust? That is just one computational challenge posed by MEMS, a technology in which multitudes of interacting tiny machines can add computational behavior to materials and the environment in an embedded, massively distributed fashion.

Microelectromechanical systems, often abbreviated as MEMS, are an emerging set of technologies that make it possible to miniaturize and mass-produce large numbers of integrated sensors, actuators, and computers. By merging sensing and actuation with computation and communication, MEMS devices can be distributed throughout the environment, coated on surfaces, or embedded within everyday objects to create distributed systems for sensing, reasoning about, and responding to events in the physical world on a scale never before possible. Distributed MEMS applications go well beyond the scaling limits of today's computational paradigms, posing serious challenges and new opportunities for information technology.

At first glance, from a computational perspective, coupling computation to the physical world might not sound like something new or terribly challenging. After all, for the past 20 years the microelectronics revolution has led to an increased reliance on computation throughout our daily lives. Computation is embedded in watches and telephones, in automobiles and aircraft, and even in toasters. A rich variety of computational tools has been developed for these embedded systems, allowing them to be limited by issues of size and cost rather than by fundamental limits of information technology.

MEMS changes the rules. Today's embedded systems typically consist of a handful of discrete sensors and actuators that are physically wired to a central control computer. In contrast, as illustrated by the example in Figure 1, MEMS-based systems can consist of thousands of integrated sensors, actuators, and computers acting over a large area. How can we structure computation and communication to enable large arrays of spatially distributed devices to act in coordination on global goals, while con-

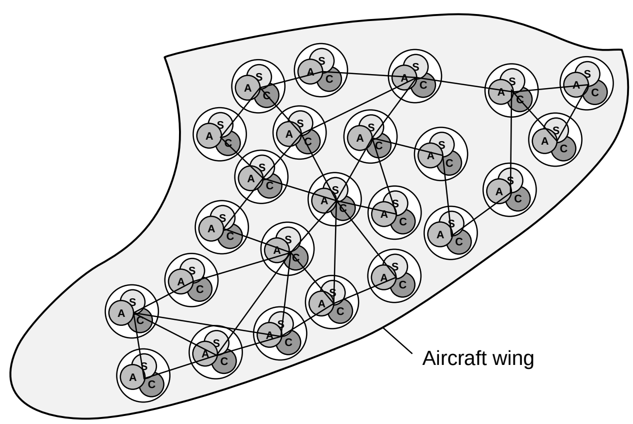

Figure 1. The distributed MEMS approach involves spreading integrated sense–act–compute modules over large areas to sense the physical world and act upon it. Tiny "flaps" embedded in the surface of an airplane wing, for instance, can reduce drag by sensing vortices and interacting with them. See http://ho.seas.ucla.edu.

1070-9924/97/$10.00 © 1997 IEEE

stantly interacting with and adapting to the environment in real time? In essence, the challenge that MEMS places on information technology is not merely to coordinate lots of tiny computers, but rather to add a bit of computational behavior to materials and the environment.

What is MEMS?

Using the fabrication techniques and materials of microelectronics as a basis, MEMS processes construct both mechanical and electrical components. Mechanical components in MEMS, like transistors in microelectronics, have dimensions that are measured in microns and numbers measured from a few to millions. MEMS is not about any one single application or device, nor is it defined by a single fabrication process or limited to a few materials. More than anything else, MEMS is a fabrication approach that conveys the advantages of *miniaturization*, *multiple components*, and *microelectronics* to the design and construction of integrated electromechanical systems. Regardless of what type of micromachining process is used, all MEMS fabrication approaches share these three key characteristics.

Miniaturization

Miniaturization is not the only characteristic of MEMS, but it is important. It brings many advantages to the performance of electromechanical devices and systems. Structures that are relatively small and light lead to devices that have relatively high resonant frequencies. These high resonant frequencies in turn mean higher operating frequencies and bandwidths for sensors and actuators. Thermal time constants—the rates at which structures absorb and release heat—are shorter for smaller, less massive structures.

But miniaturization is not the principal driving force for MEMS that it is for microelectronic devices such as integrated circuits. Because MEMS devices are by definition interacting with some aspect of the physical world (such as pressure, inertia, fluid flow, light), there is a size below which further smallness is detrimental to device and system operation. For example, reducing the size (and thus the mass) of an accelerometer makes it harder to detect low-*g* accelerations. This minimum size varies with the application, but for most MEMS applications the size limits are one to two orders of magnitude larger than the smallest microelectronic device features. Figure 2 illustrates the scale of one type of MEMS device, shown in detail in Figure 3. The reason for the dandelion seed in Figure 2 will become more clear later.

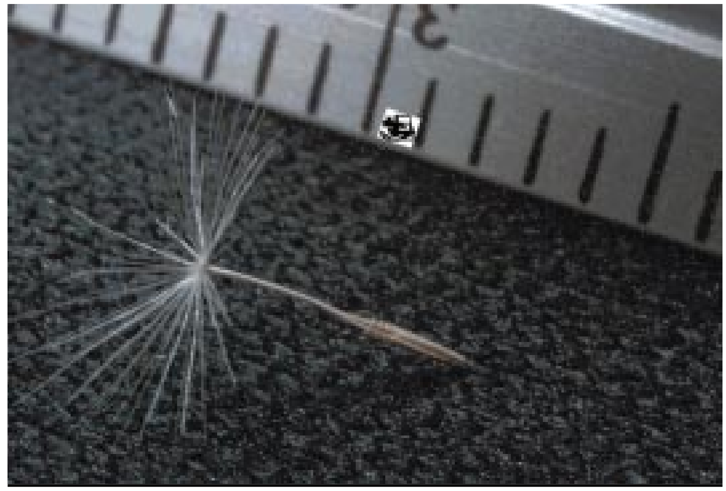

Figure 2. The scale of MEMS technology: a MEMS on-chip laser and optical system (the small rectangle overlaid on the ruler) is shown here on the same scale as a single dandelion seed—something so small and light that it literally floats in the air.

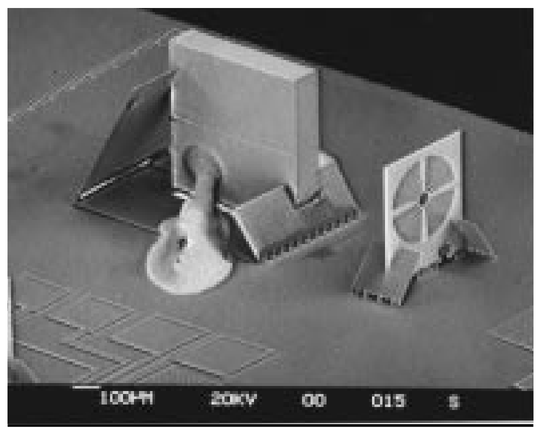

Figure 3. The laser from Figure 2, in detail. This MEMS laser, with optics suitable for transmitting light off-chip, gives an idea of what is technologically possible today. It was fabricated by Lih-Yuan Lin and Shi-Sheng Lee in Ming C. Wu's research group at UCLA.

Multiplicity

Multiplicity, or the batch fabrication inherent in photolithographic-based MEMS processing, is as important as miniaturization. It provides two important advantages to electromechanical devices and systems. Multiplicity makes it possible to fabricate ten thousand or a million MEMS components as easily and quickly as one. As the semiconductor industry has proved, such economies of scale are critical for reducing unit costs. The second, equally important advantage

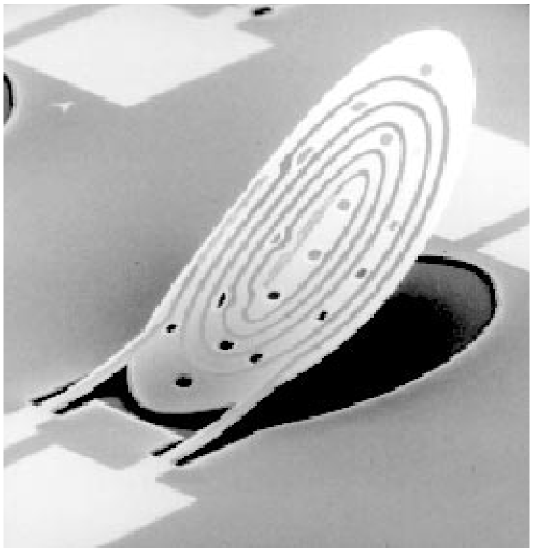

Figure 4. Example of an "active surface" MEMS device. This one is a tiny "flap"; many flaps working together could potentially reduce drag on an airplane wing as suggested in Figure 1, or position a part for assembly.

of multiplicity is the additional flexibility in the design of massively parallel, interconnected electromechanical systems.

Rather than designing components, the emphasis can shift to designing the pattern and form of interconnections (interactions or coordinated action) among thousands or millions of components. This approach to design has been standard operating procedure in microelectronic systems design for nearly three decades.

When integrated circuit engineers design and lay out a new circuit, they don't design new components, but instead design the pattern of interconnections among millions of relatively simple and identical components. The diversity and complexity of function in integrated circuits is a direct result of the diversity and complexity of the interconnections. It is the differences in the interconnections that differentiate a microprocessor from a memory. The multiplicity characteristic of MEMS has already been exploited in the development and recent demonstration of a digital micromirror display. In an array about the size of two standard postage stamps, over a million mirrors—each the size of a red blood cell—collectively generate a complete, high-resolution video image. Trying to build and operate such a display using conventional methods of mechanical component manufacturing and assembly would be nearly impossible and certainly not affordable.

Microelectronics

Finally, neither the miniaturization nor the multiplicity characteristics of MEMS could be fully exploited were it not for the microelectronics that is merged with the electromechanical components. It does not matter if the electronics processing and micromachining steps are interleaved, or if electronics processing precedes micromachining, or if microelectronics processing and micromachining are done separately and the components later packaged together by flip-chip or wire bonding. Regardless, the microelectronics integrated into MEMS devices provides the latter with intelligence and allows closed-loop feedback systems, localized signal conditioning, and the control of massively parallel actuator arrays. Moreover, the considerable investment that has been put into microelectronics materials and processing, and the expertise built up in this field, is helping the development of MEMS devices and will also help in their acceptance by systems designers and integrators.

Computational challenges

MEMS will draw on and drive computation in four key areas:

(1) control of large numbers of distributed MEMS sensors and actuators,
(2) distributed intelligence, raising the general intelligence and capability of machines and matter,
(3) MEMS devices as computational elements,
(4) multiple-energy-domain simulation, analysis, and design.

We will look briefly at only the first of these areas: the problems and opportunities created by the control of large numbers (thousands to millions) of MEMS sensors and actuators, including coupling to the physical world and environment-driven event-time demands on computation.

One of the most significant computational challenges posed by MEMS is the tight coupling of MEMS devices to the environment. This coupling causes the computation to be directed by events in the environment in real time, effectively making the differential equations that model the behavior of the physical world an implicit part of a distributed computer program. Questions of where and when to sense, of how many different sensor readings need to be correlated to determine what action to take, and of where and when an action will have the desired effect, are all determined by the physics of the environment associated with each application. The answers can vary over time as the environment changes.

Distributed MEMS applications can be grouped roughly into three classes:

Static	Topology	Dynamic
Fixed	Logical organization	Varying
Global	Degree of coordination	Local
Long	Relative lifetime of node	Short
Global	Coupling with physical world	Local

Figure 5. Rough classification of the challenges that various MEMS applications pose for information technology.

♦ *Smart particles* distributed in the environment, in which the location of the devices relative to one another varies over time. Challenges unique to smart particles include determination of relative locations, establishing a time-varying communication network, and synchronizing collaborative actions over extended distances. In the next section we'll discuss "MEMS dust" as an example of smart particles.

♦ *Active surfaces*, in which the devices are permanently attached to a surface so that there is a fixed topology (see Figure 4). The devices are coupled primarily to the dynamics of the medium they are manipulating, not to the dynamics of the surface they are attached to. The primary challenge to information technology posed by active surfaces is to dynamically recruit neighborhoods of devices to work together to interact with the physical world at a local scale, whether it be to position a part for assembly or to influence a vortex moving on an airplane wing.

♦ *Smart structures*, in which the MEMS elements are fixed in place and their interactions are coupled to one another through the dynamics of the material to which they are attached, leading to a need for some degree of global as well as local coordination.

Various MEMS configurations each pose a somewhat different set of computational challenges, which can be classified by the factors shown in Figure 5. For example, in the case of airborne surveillance particles, part of the challenge is a time-varying spatial configuration (dynamic topology). For active aerodynamic surfaces, control is complicated by the dynamics of the fluid being manipulated, and by the need for actuators to take on different roles depending on the location of vortices (varying logical organization, coupling with the physical world).

A concrete example: MEMS dust

Projections of technology trends indicate that within 4 to 5 years, it should become possible to use MEMS technology to construct a "smart dust" particle, as illustrated in Figure 6. A cloud of smart dust or dirt particles would be useful for a wide variety of applications, ranging from military reconnaissance (what's over that hill?) to precision farming (how much fertilizer does this particular square foot of farmland need?) to monitoring air quality or rush-hour traffic conditions.

The device shown in Figure 6, designed for aerial release, would last about an hour (with no recharging) in continuous operation of sensing, computing, and transmitting data. (Recharging by solar cells would increase its effective life.) This continuous operating mode, however, is for use when the particle finds itself in the vicinity of an interesting event. In sample mode, where it just wakes up roughly 1 percent of the time and takes a look around to see if anything interesting is happening, a MEMS dust particle would have enough power to operate for more than 4 days.

How do you program a cloud of dust?

MEMS dust pushes the limits of today's computational paradigms in areas from networking to computer vision, raising many questions.

How can we coordinate large numbers of unreliable parts? Some dust particles aren't going to be working and/or awake at any given time. The individual nodes will need to wake up, find out who their neighbors are, coordinate with one another, and take on tasks like distributed

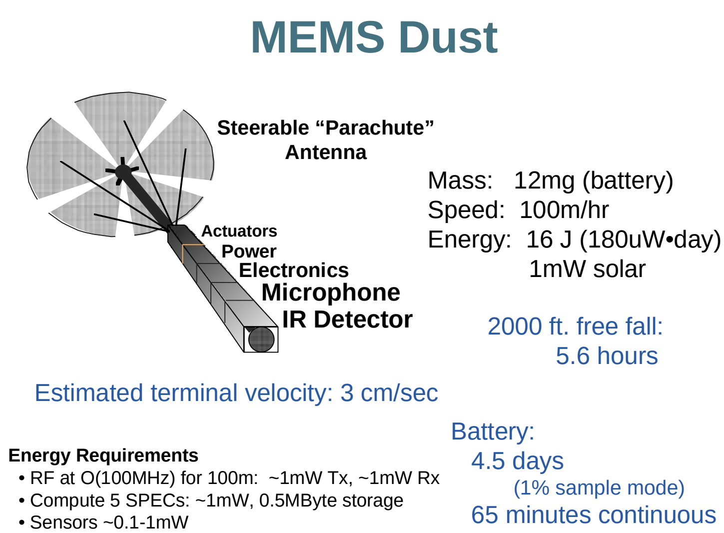

Figure 6. Schematic of a MEMS dust particle 1 cm long and 1 mm wide. It would fall at about 3 cm/s; thus if released at 2,000 feet it would stay aloft for about 5.5 hours (possibly longer if it rides thermal currents).

acoustic sensing and sound localization that require coordination of multiple particles.

How can complex real-time signal processing computations be divided to run in parallel on a cloud of dust particles? Smart dust blurs the distinction between distributed computing (using paradigms like client-server models) and parallel computing (with paradigms like SIMD and VLIW). Parallel computing has traditionally involved relatively close temporal synchronization among processes, which tend to be running on a set of processors located quite near one another. Distributed computing, on the other hand, has typically been associated with processes that may be operating in distant locations, synchronizing and exchanging results far less frequently. Distributed MEMS will require new paradigms that support a high degree of synchronization over fairly large distances in order to enable applications, such as sound localization, that require tight synchronization to correlate data about events in the environment in real time.

Smart dust raises many other questions that pose serious challenges for computational science. Should all the smart dust particles run the same program, as in a SIMD machine, or should particles specialize and diverge from one another? How do the particles synchronize with one another? How can particles be dynamically recruited into collaborative groups? How can communication be established and maintained in a system where the physical topology varies over time? How do we build a global view of a situation using many small pieces of information that are collected in different places at different times? In an energy-limited programming environment, when is it better to compute (interpolate) a result, when is it better to communicate with a neighboring particle to obtain the result, and when is it better to sense the result in the environment? How can spatially distinct data streams collected at different times be combined in an energy-efficient manner? What are the abstraction mechanisms that allow easy programming of these systems?

MEMS technologies have a rich assortment of applications, including inertial measurement and navigation, micro-optomechanical devices, mass data storage, distributed sensing and control, and aerodynamic control of aircraft. The binding theme of MEMS technologies is the merger of sensing and actuation with computation and communication. As advances in MEMS enable higher levels of electronic–mechanical integration and greater numbers and densities of devices, computational scientists will be called upon to devise new computational strategies and new architectures that reflect distributed MEMS structures. These contributions will be critical to advancing and fully exploiting the opportunities created by MEMS. ♦

Acknowledgments

This article is based in part on the report of the MEMS Working Group of the ACM 50th Anniversary Workshop on Strategic Directions in Computing Research (SDCR), chaired by Kaigham Gabriel; in part on the report of the 1995 ISAT Study on Distributed Information Systems for MEMS, chaired by Andrew Berlin; and in part on a DoD Industrial Assessment Report on MEMS led and principally written by Kaigham Gabriel. ISAT study participants included Harold Abelson, Natalya Cohen, Larry Fogel, Chih-Ming Ho, Mark Horowitz, Jonathan How, Thomas F. Knight, Richard Newton, and Kris Pister. The SDCR working group included Andrew Berlin, Rick Carley, Ann-Marie Lanzillotto, Keith Uncapher, and Jacob White.

Andrew A. Berlin is a member of the research staff at the Xerox Palo Alto Research Center. He received SB, SM, and PhD degrees in electrical engineering and computer science from the Massachusetts Institute of Technology. His research interests lie in coupling computation to the physical world, including applications of MEMS and smart materials, as well as parallel programming languages and architectures. From 1989–91, Berlin worked as a principal designer of the MIT/Hewlett-Packard Supercomputer Toolkit. In 1993 he joined Xerox PARC, where he initiated a project that explores (from both fabrication and computational perspectives) ways to use large arrays of MEMS devices to control the dynamic behavior of macroscale objects.

Kaigham J. Gabriel is director of the Electronics Technology Office at the (US) Defense Advanced Research Projects Agency. He received his SM and PhD degrees in electrical engineering and computer science from MIT. In 1985 he joined AT&T Bell Labs in the Robotic Systems Research Department, where he started the silicon MEMS effort, and led a group of researchers in exploring and developing IC-based MEMS for applications in photonic and network systems. During a sabbatical year from Bell Labs, Gabriel was a visiting associate professor at the Institute of Industrial Science, University of Tokyo, Japan. Since 1992 he has been at DARPA in various posts. E-mail, kgabriel@darpa.mil.

Corresponding author: Andrew A. Berlin, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA; e-mail, berlin@parc.xerox.com.