The Virtual Vehicle - A Materials Handling System Using Highly Distributed Coordination Control

Jonathan E. Luntz, Eric Rollins, William Messner
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
j19f@cmu.edu, er2g@cmu.edu, bmessner@cmu.edu

Abstract

In this research, we are developing a materials handling system where many small, simple actuators cooperate to convey large objects with three degrees of freedom in a plane. The actuators, or cells, are arranged in a regular array, fixed to a planar surface, and objects are passed over the array. Such an array provides very flexible materials handling in which many objects can be conveyed simultaneously in different directions. The array is coordinated in a distributed manner, rather than by a central controller, where each manipulator has its own controller, and each controller communicates with its neighbors. The system is modeled as a continuous medium rather than a collection of independent entities, and useful behaviors are derived from physical analogies such as heat transfer and fluid flow.

1. Introduction

1.1. Application and Concept

Several applications exist where objects are to be manipulated with translation and rotation in a plane. For example, flexible manufacturing systems often require working parts to be conveyed between several machines in arbitrary order. Also, package handling systems, such as airport baggage handling, require parcels to be sorted and transported to one of several different locations. Two standard approaches for handling materials in such a manner are the use of multiple conveyor belts (see Figure 1), where parcels are pushed from one belt to another, and the use of large robotic manipulators (see Figure 2) which hold and carry objects between destinations.

We propose and are developing an alternative method for manipulating objects in a plane, where many small actuators (cells) fixed in a planar array cooperate to handle materials. Each cell consists of an actuator capable of applying a force in any direction to a parcel which rests on top and a sensor (or sensors) capable of detecting the presence of an object (see Figure 3). This is similar to the cellular robotics concept presented by Beni and Wang [1][2][3] except that in our system one object is handled by many manipulators together rather than one manipulator alone. Possible actuators include powered wheels which can be steered in the desired direction of motion, or small fingers capable of pushing on the above object. Through proper coordination, parcels which ride on top of the array of cells can be made to translate and rotate in a plane. Since sensing and actuation are distributed, each of

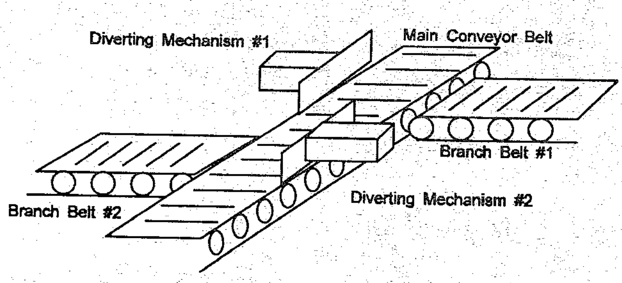

Figure 1: Multiple conveyor system

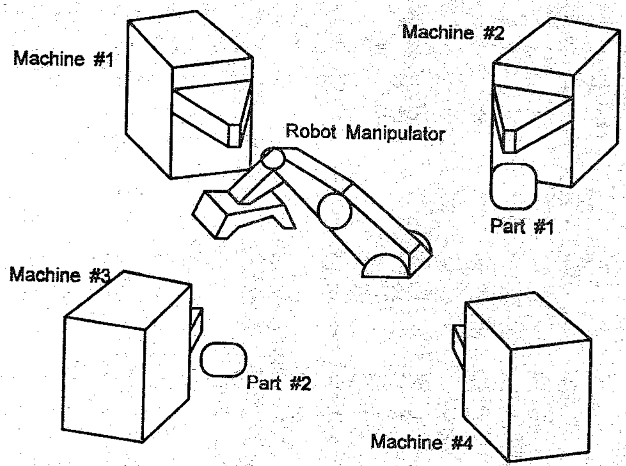

Figure 2: Flexible manufacturing system with a robotic manipulator

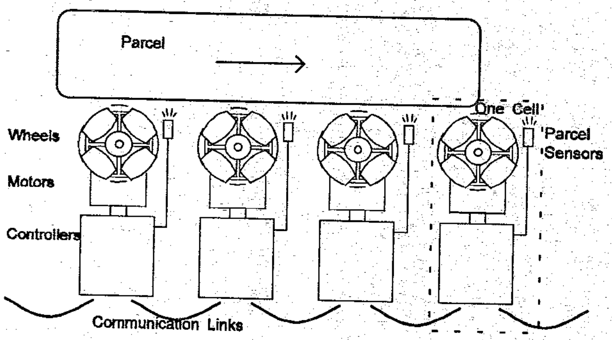

Figure 3: A few manipulator cells carrying a parcel

many parcels can be manipulated independently, appearing as if it were carried by a separate vehicle (see Figure 4). Hence the name *Virtual Vehicle*.

0-7803-2550-8/95$4.00©1995 IEEE

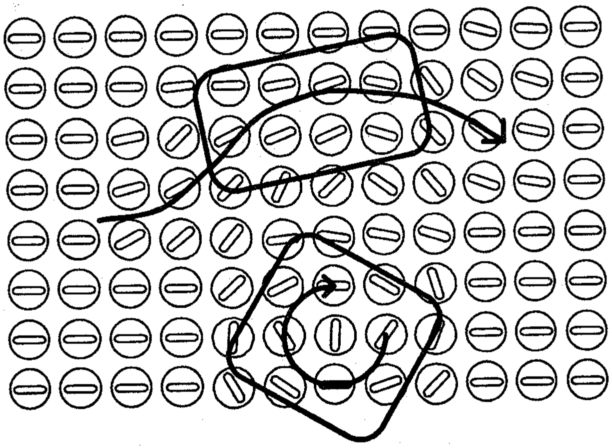

Figure 4: The Virtual Vehicle: Several parcels can be translated and rotated independently.

The remainder of Section 1 of this document discusses general aspects and capabilities of this type of materials handling system. Section 2 presents the mechanical, electrical, and communication configurations of the prototype system we are building. Section 3 examines some of the modeling and control issues for such a system, while concluding remarks are made in Section 4.

1.2. Cooperative Control

One possible way of controlling the array of cells is to use a single computer which sends commands to all the cells on the array. The computer would decide how to coordinate motion between all the cells. We are exploring another paradigm in which each cell has its own controller, and each controller communicates with its neighbors. Each controller only is aware of its immediate surroundings, but can coordinate with neighboring cells to produce larger-scale coordinated motion. This idea of cooperative control is inspired by biological systems, such as swarms of insects and flocks of birds, where relatively simple agents in communication cooperate to produce complex behavior.

1.3. Advantages

For many applications, a dedicated robot or conveyor system is the simplest, most appropriate solution. There are cases, however, where features such as additional flexibility and reconfigurability are required. In these cases, special properties of the Virtual Vehicle give it advantages over other materials handling systems. For example, in airport baggage handling, long conveyors could be used to transports parcels over long distances while Virtual Vehicle arrays could be installed at conveyor junctions to allow parcels to be sorted smoothly and quickly between belts. Also, in flexible manufacturing systems, a Virtual Vehicle array could transport many parts simultaneously between the various machines, while small, simple robotic manipulators could precisely hold parts in position at each machine.

Since the actuation is distributed, multiple parcels can be manipulated independently. This flexibility allows for parcels to be sorted, reordered, and redirected quickly. Reordering objects with conveyor belts is time consuming since a belt has only one degree of freedom, and all objects on a belt must move together. Reordering of objects can be accomplished with robotic manipulators, but only one object per manipulator can be handled at a time and coordination of several manipulators themselves is complex and interference between manipulators can be a problem. Also, objects of many sizes, shapes, and weights can be passed along the array of manipulators. Robotic manipulators, on the other hand, may not be able to lift heavy objects nor be able to grasp objects with complex shapes.

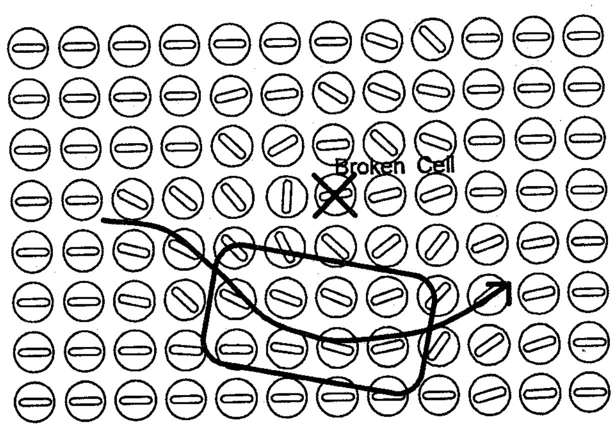

Figure 5: Operating cells can work around a broken cell until it is replaced.

Because of the small size and relative simplicity of the actuators, many cells can be produced at relatively low cost. An entire array of small actuators could be purchased at possibly a much lower cost than enough large robotic manipulators to provide equivalent functionality. The cells can be designed to "snap" together to form an array. This modularity allows cells to be arranged to fill a region of any shape, and the array can be easily modified or expanded by moving cells or adding new cells.

An important property of the Virtual Vehicle is redundancy. Since the cells are relatively inexpensive, spares could be kept on hand, so that if a cell were to break down, it could quickly be replaced by a new cell. In the meantime, functionality is not necessarily lost. The other cells could "work around" the broken cell (see Figure 5), or possibly they could just pass objects over it. In contrast, both conveyor belt and robotic manipulator systems have large integrated components, and when they break, the entire production line must often be shut down while repairs are made.

Another aspect of the Virtual Vehicle is scalability. Cells can be designed to carry objects of practically any size. Micromachined actuators can carry near-microscopic objects [4], such as integrated circuit components, small plastic wheels can carry suitcases through an airport, larger rubber wheels can carry shipping palates, and huge truck tires can carry boxcars around a ship/train yard. Since the objects are much larger than the cells and many cells carry an object at once, very heavy objects can be carried on a Virtual Vehicle made up of relatively light-duty, and hence relatively inexpensive, cells.

2. System Architecture

2.1. Mechanical Configuration

As well as developing the theory to model and control the Virtual Vehicle, we are building a prototype system. This system consists of a small array of cells capable of transporting objects about the size of a breadbox. The simple task of these cells is to create a directable force vector. Some of the options we examined for doing this are shown in Figure 6. The most obvious option is an "upside-down" wheel that can be steered and driven. Figure 6 also shows a cell using linear actuators and a cell constructed as an "inverted trackball".

The option we have chosen is a pair of orthogonally oriented "roller wheels" for each cell (see Figure 7). Each of these wheels is capable of creating a force normal to its axis while allowing free motion parallel to its axis. Thus two of these wheels can, if coordinated, produce a directable force from each cell. Each of these wheels is connected through a speed reducing gear and pinion set to a small 12VDC/1A motor. Along with the controlling electronics, two of these assemblies, orthogonally mounted on a small base make up each cell.

Each of these cells is connected to a larger breadboard style base to create a regular array of simple manipulators - a vector field. Current resources restrict us to build only twenty cells which limits the depth to which we can examine the theory in two dimensions. This mounting system, however, provides the freedom to reconfigure the setup into a one

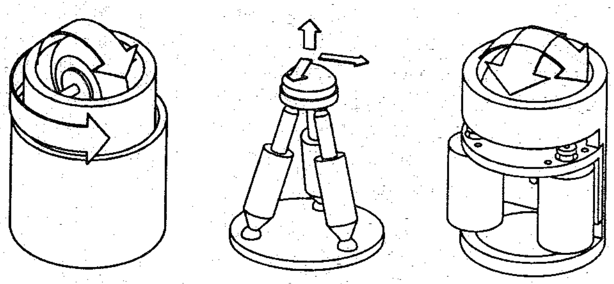

Figure 6: Several possible manipulator options: steered wheel (left), linear actuator "finger" (middle), and inverted trackball (right)

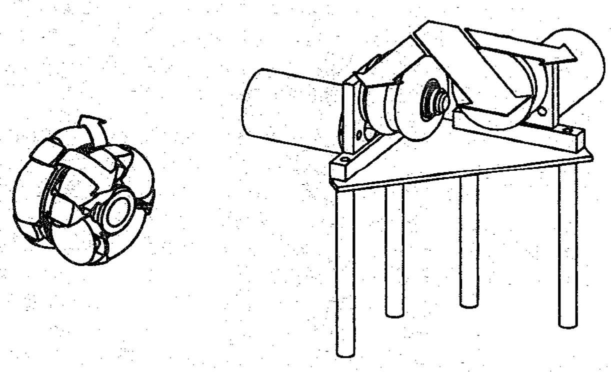

Figure 7: Our prototype manipulator cell: roller wheels (left) are arranged orthogonally to provide a vector force in any direction (right)

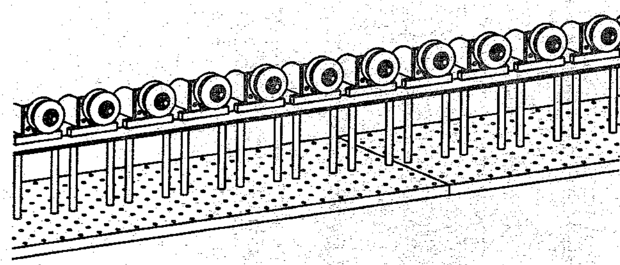

Figure 8: A 1-D row of cells mounted to a breadboard (controllers would be placed under each cell)

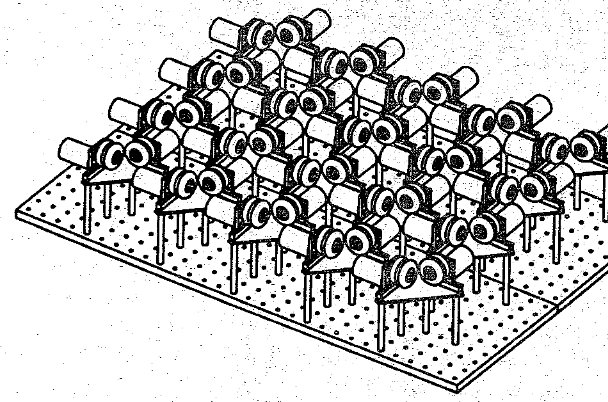

Figure 9: A 2-D array of cells mounted to a breadboard (controllers would be placed under each cell)

dimensional array. With this flexibility, we can study the mechanical and other physical aspects of a small 2D array and gain a clear grasp on theoretical issues with a more extensive 1D array.

The 1D array includes a guide to stabilize the transported items as shown in Figure 8. For the 2D system the cells can be arranged on the breadboard in different arrays. A triangular array would give the closest packing of cells and thus would allow the highest number of support points per item of a given size. Hardware sizing and manufacturing constraints, however, have guided us to produce the cartesian array shown in Figure 9.

2.2. Electronic Configuration

We are exploring the use of distributed control on the Virtual Vehicle because it is a relatively underdeveloped and interesting control paradigm, and because we believe that it would become impractical or impossible for a single computer to control hundreds or possibly thousands of cells. The controller of each cell is a single-board computer based on the Motorola MC68HC11 microprocessor. The capabilities of this processor allow it to control more than one cell, so it may be more practical to have one controller handle several cells. We, however, would like to explore fully distributed control. In application either fewer or simpler processors could be selected to build a more cost effective system.

The speed of the wheels is measured by a home made quadrature encoder consisting of a pattern printed on paper mounted directly to the large gear and two reflectance infrared emitter/detector pairs. We are using reflectance rather than a chopper plate since it is more compact and simple to build. The computer then reads the quadrature signals and calculates the speed and direction using internal timers.

The motor is being driven by pulse width modulation and an H-bridge circuit. Again, this was chosen for its simplicity and its ease of interface with the digital computer. The presence of a parcel is sensed by a photo-transistor which detects the room's ceiling lighting and switches off under the shadow of a parcel. Power is transmitted to each cell by two lines. A 5V line is required to drive the computer and other electronics, while a 20V line powers the motors.

2.3. Communication

An important aspect of our control scheme is the communication between cells. Each single-board computer has a RS232 serial communications port. This port is multiplexed between each of the four neighboring cells, conceptually equivalent to a one-of-four ABCD switch controlled by the processor (see Figure 10). For a cell to send a message to or receive a message from a neighbor, it must actively select it with the multiplexer.

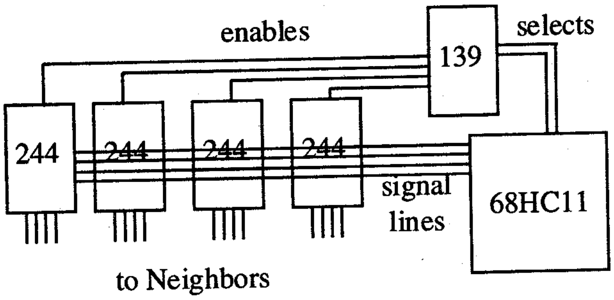

Figure 10: The processor uses an LS139 decoder to enable one of four LS244 buffers to select transmit, receive, and handshaking lines from one of four neighbors.

Handshaking lines are necessary to alert a neighboring cell that a message is waiting to be sent since only one neighbor can be selected at once. The handshaking lines are also multiplexed, to reduce the number of processor I/O pins used, and therefore the processor polls through all four neighbor's handshaking lines to detect any incoming messages. When an active handshaking line is found, that neighbor's serial port is selected, and an outgoing handshaking line is activated. After the message is sent, the serial port is de-selected, and the handshaking lines are dropped. Figure 11 shows a flowchart describing the handshaking process.

Information can be transmitted between cells in two ways. A cell can "force" information upon another cell by simply sending a message containing the information. Information such as the presence of a parcel would be sent in this manner. Alternatively, a cell can request information from a neighbor by sending a request message. The neighbor then responds by sending a message containing the requested information (for example, the current rotation speed of a cell's actuator.)

This type of communication may seem awkward compared to other network schemes where each cell would have an address, and all cells share a data bus. With a shared data bus, however, only one cell would be able to send a message

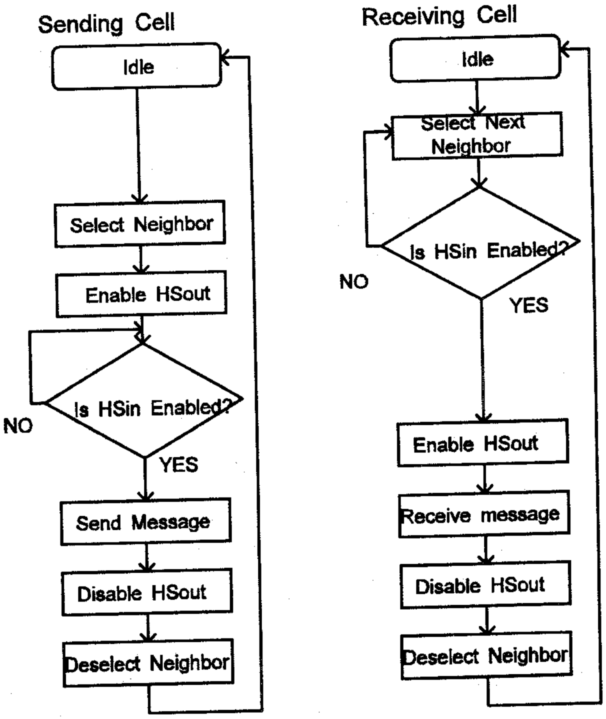

Figure 11: Flowchart showing polling hand handshaking scheme. One cell polls its neighbors (right) until the other cell attempts to send it a message (left).

at any given time. We would like many communications to occur simultaneously around the array. Also, we would like our network not to be limited by a maximum address space, or to require any particular cell to know how it is located relative to the rest of the array (which addressing does require). In our system, each cell need only be aware of its immediate surroundings. If any information needs to be sent further than one cell away, it can be passed along the array, from one cell to the next.

3. Control, Coordination, and Planning

3.1. Continuous Medium Model

In concept, we will develop relatively simple control laws for each cell which, when communicating with other cells, more complicated behaviors will emerge which will be useful for transporting materials. In order to predict the emergent behavior, we need a model of the distributed system as a whole. Such a model can be built up from the model for a single cell. The equation of motion for a first order model of the k^{th} cell (motor and wheel) would be as follows:

$$m \frac{dv_k}{dt} = -bv_k + k_f u_k - F_k \quad (1)$$

where m is wheel mass, v_k is wheel speed, b is the damping in the motor, k_f is the motor torque constant, u_k is the control signal, and F_k is the force developed by cell.

The entire collection of N cells could then be modeled as an N^{th} order system, but this would not provide a natural expression of the dynamics of the system. Instead, we approximate the behavior of the collection of spatially discrete systems as a single spatially continuous system. We propose that if the standard process of approximating a spatially continuous system as a spatially discrete system for purposes of computation is reversed, a useful model of the system as a whole can be obtained.

For example, in a one dimensional row of cells, if the k^{th} cell used proportional speed control with the reference speed being the average speed of its two immediate neighbors, the control protocol would appear as

$$u_k = k_v\left(\frac{v_{k-1} + v_{k+1}}{2} - v_k\right) \quad (2)$$

Examining only those cells which are not under a parcel, the external force term, F_k, in equation (1) is removed. With the control protocol in equation (2) the dynamics of the k^{th} cell become

$$\frac{dv_k}{dt} = -\frac{b}{m}v_k + \left(\frac{k_f k_v}{m}\Delta x^2\right)\left(\frac{v_{k-1} - 2v_k + v_{k+1}}{2\Delta x^2}\right) \quad (3)$$

where Δx is the spacing between cells. The last term contains the finite difference approximation of the second derivative with respect to x. Grouping the constants, and eliminating the index k, the second derivative can be substituted into equation (3), resulting in the partial differential equation:

$$\frac{dv}{dt} = -hv + \alpha\frac{\partial^2 v}{\partial x^2} \quad (4)$$

This partial differential equation is of the form of the 1-D heat conduction equation with heat loss proportional to temperature. Here, the local cell velocity is analogous to temperature. This physical analogy is useful for understanding the behavior of the highly distributed control system. In fact, this equation can be solved analytically for certain boundary conditions to give an approximate description of the behavior of the system [5].

Useful behaviors can be obtained from continuous medium models. For example, if a parcel is traveling along the array, the cells in front of the parcel must be brought up to speed before the arrival of the parcel so that the parcel does not skid when it reaches a new cell. The heat transfer protocol can be applied for this purpose. The velocity of the leading cell under the parcel can be "conducted" along in front, analogous to the conduction of heat in ahead of a moving melting front, creating a velocity profile in front of the parcel (see Figure 12) such that cells begin to accelerate before the parcel arrives. This has been done in simulation [5].

Other continuous medium models can be applied to generate useful behaviors. Repulsive electric fields have been simulated [5] to avoid collisions in 1-D between parcels or to keep a parcel from falling off the edge of the array. Differential equations representing fluid flow can be implemented on the array to move parcels in desired directions and around obstacles, and to cause rotation of the parcels by implementing fluid vorticity.

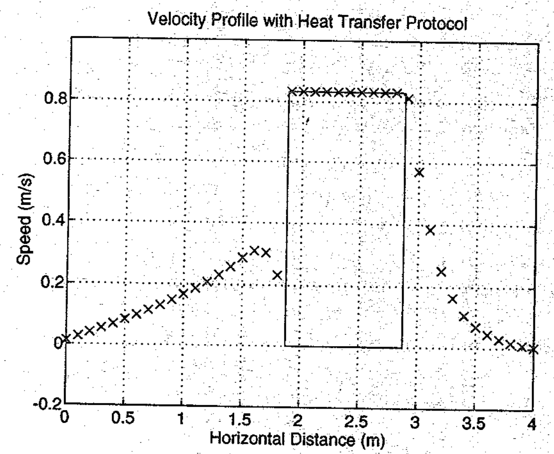

Figure 12: Simulation of the heat transfer protocol. Cells (shown by X's) in front of a parcel (shown by the box, moving to the right) to accelerate up to speed before the arrival of the parcel.

3.2. Robotic Planning Methods

In addition to controlling the array using continuous medium models, standard robotics path planning techniques can be implemented on the array to direct parcels. Grid based methods of generating potential fields [6] fit naturally into the framework the Virtual Vehicle, since each cell can represent one grid cell. V-diagrams can be constructed automatically by the array by propagating distances from obstacles from cell to cell. Methods such as this can be used to set up potential fields from destination to destination while avoiding static obstacles. To avoid moving obstacles (i.e. other parcels) a reactive grid based planning method can also be used, such as the D* algorithm developed by Stentz [7]. Information about new and moving obstacles can be propagated along the array in real time. This type of method is natural for the Virtual Vehicle since it takes advantage of the distributed nature of the system.

4. Concluding Remarks

Here, we have proposed a new method of materials handling, using distributed actuation under distributed control. This method has several important properties which may give it advantages over alternative systems in certain situations. We have described both the mechanical and electronic structures of the prototype system we are currently developing. Our prototype is just one example of this method of materials handling in that many other configurations are possible. This system can be modeled with continuous medium models allowing physical analogies to be used both to better understand the dynamics and to develop useful control protocols. Also, standard robotics path planning techniques may be converted into control laws for this system.

Current resources allow us to build a prototype system with twenty cells arranged either in a single row or in a rectangular array. When completed, the row of cells will allow us to verify the theoretical and simulation studies we have done in 1-D [5]. The small rectangular array is not enough to actually transport materials, but we can examine communication between cells in 2-D.

We plan to extend our theoretical and simulation studies to include a full 2-D array carrying several parcels simultaneously without collisions. In simulation, we will validate fluid flow and electric field models as well as robotic planning methods. We will then build a full scale 2-D prototype consisting of at least a hundred cells in order to fully implement and test this method of materials handling and control.

It is also important that we carry out a feasibility study into the application of the Virtual Vehicle. We need to determine when (and if) the Virtual Vehicle is functionally and economically advantageous over other systems, and how the Virtual Vehicle can best be implemented in a real application. We feel that the Virtual vehicle not only provides an interesting control problem, but also is a useful device.

5. References

[1] Beni, G., "The Concept of Cellular Robotic Systems," Presented at the 3rd annual IEEE Symposium on Intelligent Control, Arlington, Virginia, August, 1988.

[2] Wang, J., and Beni G., "Pattern Generation in Cellular Robotic Systems," Presented at the 3rd annual IEEE Symposium on Intelligent Control, Arlington, Virginia, August, 1988.

[3] Beni, G., and Wang, J., "Cellular Robotic System with Stationary Robots and its Application to Manufacturing Lattices," Presented at the IEEE International Conference on Robotics and Automation, Sacramento, California, April, 1991.

[4] Konishi, S., and Fujita, H., "System Design for Cooperative Control of Arrayed Microactuators," Presented at the IEEE Micro Electro Mechanical Systems Conference, Amsterdam, The Netherlands, January, 1992.

[5] Luntz, J.E., and Messner, W., "Work on a Highly Distributed Coordination Control System," Presented at the AACC American Controls Conference, Seattle, Washington, June 1995.

[6] Latombe, J-C., Robot Motion Planning, The Kluwer international series in engineering and computer science. Kluwer Academic Publishers, Boston, Massachusetts, 1991.

[7] Stentz, A, "The D* algorithm for real time planning of spatial traverses," Technical report, Carnegie Mellon University, The Robotics Institute. CMU-RI-TR-94-37, 1994.