University of Southern California

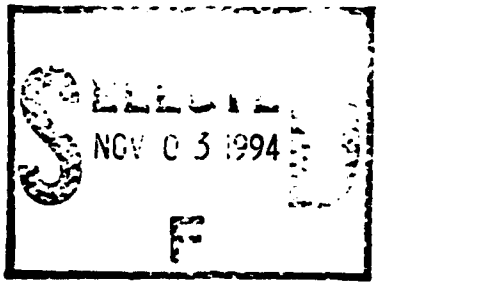
11.

Parts Manipulation on a MEMS Intelligent Motion Surface

Peter Will & Wenheng Liu

ISI/RR-94-391

May 1994



2695

INFORMATION SCIENCES INSTITUTE

310/822-1511

4676 Admiralty Way/Marina del Rey/California 90292-6695

ISI Research Report
ISI/RR-94-391
May 1994

Parts Manipulation on a MEMS Intelligent Motion Surface

Peter Will & Wenheng Liu

ISI/RR-94-391
May 1994

University of Southern California
Information Science Institute
4676 Admiralty Way, Marina del Rey, CA 90292-6695
310-822-1511

REPORT DOCUMENTATION PAGE

FORM APPROVED
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1994	Research

4. TITLE AND SUBTITLE
Parts Manipulation on a MEMS Intelligent Motion Surface

5. FUNDING NUMBERS
DABT63-92-C-0025

6. AUTHOR(S)
Peter Will
Wenheng Liu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
USC INFORMATION SCIENCES INSTITUTE
4676 ADMIRALTY WAY
MARINA DEL REY, CA 90292-6695

8. PERFORMING ORGANIZATON REPORT NUMBER
RR-94-391

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
ARPA/ESTO
3701 N. Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
A shorter version of this report has been submitted to the IEEE Transactions on Robotics

12A. DISTRIBUTION/AVAILABILITY STATEMENT
UNCLASSIFIED/UNLIMITED

12B. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An intelligent motion surface consisting of an array of micro-robots has been designed and simulated for use in applications requiring the manipulation of parts for micro and precision assembly. The array was designed for implementation in Micro-Electro-Mechanical Systems (MEMS) technology. A specific form of micro-robot, called a LAMBDA machine, was used in the initial design. Individual micro-robots can be programmed so the ensemble can operate in a variety of modes, called gaits. Groups of contiguous micro-robots executing identical but phase shifted motions (gaits) have the effect of imparting a spatially defined force field on a part or set of parts, placed on the group. Appropriate choice of force fields in shape, spatial extent and vector direction is shown to cause parts to be moved to assembly operations like translation, rotation, orientation, aligning and filtering based on part shape.

A simulator was built to explore the practicality of programming a miniature assembly bench constructed by silicon micromachining an array of these micro-robots. Image and graphics based modelling techniques were explored for speed: graphics based simulation is superior especially when multiple parts are involved.

Experiment results on primitive assembly operations, limitations of approach, extensions and possibilities for future work are discussed.

14. SUBJECT TERMS
Microrobotic array, intelligent motion surface, parts orienting, shape dependent filtering, micro-robotic assembly system, micro-robotic gaits.

15. NUMBER OF PAGES
22

16. PRICE CODE

17. SECURITY CLASSIFICTION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reoprts. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay within the lines to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month,a nd year, if available (e.g. 1 jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered
State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element numbers(s), project number(s), task number(s), and work unit number(s). Use the *following labels:*

C - Contract	**PR - Project**
G - Grant	**TA - Task**
PE - Program Element	**WU - Work Unit Accession No.**

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the repor.

Block 9. Sponsoring/Monitoring Agency Names(s) and Address(es). Self-explanatory

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of ...; To be published in... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	**- See DoDD 5230.24, "Distribution Statements on Technical Documents."**
DOE	**- See authorities.**
NASA	**- See Handbook NHB 2200.2.**
NTIS	**- Leave blank.**

Block 12b. Distribution Code.

DOD	**- Leave blank**
DOE	**- Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.**
NASA	**- Leave blank.**
NTIS	**- Leave blank.**

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17.-19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contins classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Parts Manipulation on a MEMS Intelligent Motion Surface

Peter Will & Wenheng Liu

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina Del Rey, CA 90292

Email Address: will@isi.edu, liu@isi.edu

0. Abstract

An intelligent motion surface consisting of an array of micro-robots has been designed and simulated for use in applications requiring the manipulation of parts for micro and precision assembly. The array was designed for implementation in Micro-Electro-Mechanical Systems (MEMS) technology. A specific form of micro-robot, called a LAMBDA machine, was used in the initial design. Individual micro-robots, initially without sensing, can be programmed so that the ensemble can operate in a variety of modes, called gaits. These gaits, some of them explored in the paper, have different lifting and translatory properties and can be chosen for best effect in a particular situation. The array can be operated in various gaits to impart distributed vector forces to a part placed on the array. Groups of contiguous micro-robots executing identical but phase shifted motions have the effect of imparting a spatially defined force field on a part or set of parts, placed on the group. The appropriate choice of force fields in shape, spatial extent and vector direction is shown to cause parts to be moved in manners that are useful and appropriate to assembly operation such as translation, rotation, orientation, aligning and filtering based on part shape.

In the first stage of the study, described here, a simulator was built to explore the practicability of programming a miniature assembly bench constructed by the silicon micromachining of such an array of micro-robots. The simulation allowed the entry of the shape and form of the force fields, the shape of the parts or parts to be moved and the interactive placement of the part on the field for the subsequent motion.The simulation used simple physical models of the forces contributed by each micro-robot. Both image and graphics based modelling techniques were explored for speed: graphics based simulation is superior especially when multiple parts are involved.

The results of experiments on primitive assembly operations, limitations of the approach, extensions and possibilities for future work are discussed.

1. Introduction

The goal of the work described here is to explore the methods of use and limitations of a bed of a large number of small manipulator arms in moving, orienting and holding micro-objects; that is, the use of arrays of micromanipulators as a miniature assembly bench. The application domain includes the assembly of multi-chip modules (MCM), opto-electronic integrated circuits (OEIC) and implantable medical devices, etc. It is expected that assembly using this method will show an increase in throughput and in the quality of the product compared to the current human assembly techniques which are fraught with difficulty at small geometric scale; today, chips used in MCM's are first loaded in particular orientation visually and then placed in the correct location by pick-and-place machines. The intelligent motion surface has the potential to replace the traditional visual recognition system by its intrinsic design. Meanwhile, unlike the traditional method, the Intelligent Motion Surface can orient and align the component parts in parallel to speed up the assembly work. On the other hand, however, the miniature mechanism is hard to implement; only a few degrees of freedom may be possible in the real design of the micro-robot. Several factors need to be considered in the overall system design, these include the structure of the unit robot, the spatial arrangement of an array of them, the system control modes, and assembly scheduling schemes.

In the literature, previous work includes papers on the design of the unit manipulators, [IBM93] [PISTER90] [PISTER92], papers on the theory in orienting parts without sensing [GOLDBERG93] and the theory of manipulation and control for micro-actuator arrays [BOHRINGER94] respectively. The interest here is in exploring the feasibility and limits of the ideas necessary for building the micro-system, the trade-off between sensory and inert arrays, their communication and control needs, their gait manners to drive a part forward and the implementation of an integrated system.

The micro-robot assembly system will be implemented on VLSI wafers. The micro-robot configuration chosen for the current study, the LAMDA machine, is a four bar linkage micro-manipulator. Each robot applies a force to the part in contact with its tip. The unit manipulator can be used to populate a two dimensional array in a variety of configurations and, therefore, generate different force fields. Each robot can be controlled in different modes to perform motion. For an array of robots, the individual robots can move either synchronously or asynchronously to achieve ways to translate a part on the array. The surface of the array may become "intelligent" when the control system is installed so that the operations of the unit manipulators are programmable; i.e. the various force fields supply the ability to perform different functions to translate, rotate, or sort parts to destination positions for the assembly purpose. The assembly jobs can be scheduled by calling sequences of basic functions.

The array was studied first in simulation. In the simulation approach, a simple user interface has been implemented in the X-Window environment. It supports the ability to dynamically change configurations of the force fields generated by its simple micro-manipulators. Users can also define parts in arbitrary shapes and locate them on the field array. The simulator will then display the translations and rotations of the parts within the fields.

The simulation was designed to show the basic concepts from the micro-robotic operations to the

overall system mechanism. Several experiments were designed including the centering, aligning and spinning of parts, the motion in a convergent or a divergent field array, and the programmable field array to perform assembly.

The paper is arranged as follows. Section 2 introduces the concepts of the design of both unit manipulator and planar manipulators. The user interface, simulation models and procedures are introduced in section 3. The sets of experiments and results are described in section 4. Section 5 contains the conclusions and plans for the future study.

2. The Concepts of Unit Manipulator and Array of Manipulators

2.1). Unit manipulator

The mechanism studied in most detail so far in the work described here, is the simple 4-bar linkage λ (LAMBDA) machine in Figure.1. This shows a manipulator cell, pin jointed, with the main beam inclined at some angle to the right of the vertical with two motors driving the base joints. Under control of the base motors, the tip of the manipulator can be designed to operate in relatively high left to right motion followed by a relatively lower right to left motion and thus to make a loop motion in space that will be defined as an R-motion.

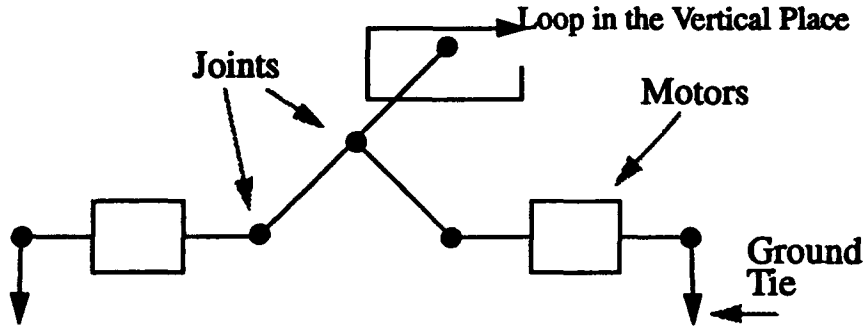


Figure 1. 4-bar linkage λ machine

2.2). Gaits of Array of Manipulators

The operation of an array of micro-robots on moving a part placed on its surface can be illustrated using a linear array of such manipulators operating in the same direction. The control of individual micro-robots may be synchronous or asynchronous.

2.2.1). Synchronous Operations

An object placed on the one dimensional array of manipulators shown in Figure 2, each of which is executing an R-motion, will move, under suitable conditions, to the right. A plausible explanation is that, if the motion is executed at a high speed relative to the effective time constant of the mass, the inclined angle of the main beam will cause the tip to tend to slip on the leftward motion

and to "dig into" the object on the rightward motion and push the carried object to the right, hence the term R-motion. An alternative way of looking at the operation of the array is to note that there is a differential friction between the two directions of motion of the base manipulators relative to the object and this impresses different forces on the object in the two directions. Within constraints to be determined and simulated, the array will support the object without being crushed or overpowered and fail to operate. An array of manipulators using a unit cell in the form of a mirror image of the R-manipulator would move parts to the left, i.e. execute an L-motion.

However, it may turn out on close examination, that the difference in friction in the two motion directions is insufficient to drive the part forward. In that case, it might be better to use a second mode of control in which the drive array for a single direction is broken up into two interleaved sets, the odd and the even manipulators in order to achieve more positive control of the forces applied to the part being carried. This motivated examination of the following two asynchronous modes of operation.

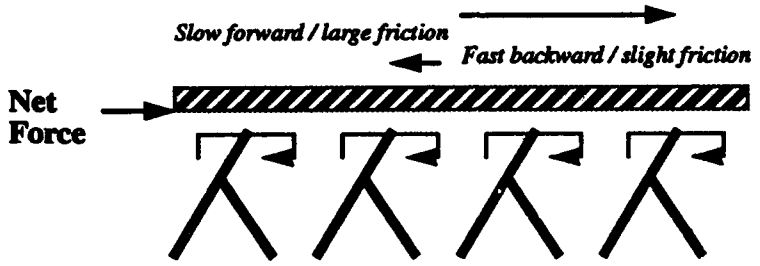


Figure 2. Single-Phase Gait

2.2.2). Asynchronous Operations

The idea of the asynchronous operation is that the odd and even manipulators hold and drive the part alternately. This mode of motion, a type of the general idea called "gait" in robotics literature, was used intrinsically by Fujita[FUJITA93] in his experiments and by Bohringer [BOHRINGER94]. Gaits need not be strictly interleaved but may have random phase or a pattern suited to the part being carried. Various one- and two-dimensional random gaits can be explored in simulation. Some manipulators can be used in supporting the parts while some of the manipulators retract and others drive forward. Each motor in the array needs to be under individual control in order to achieve maximum manipulation flexibility. First, the even set could be used to support the object while the odd set moves to the left-most position to execute the up and to the right motion, thus moving the supported object one unit to the right. At this point the sets change function and the odd set supports and the even set drives the object one unit to the right. Continuing this mode of operation causes the supported module to continue to traverse to the right. Figure 3 and Figure 4 illustrate two methods of the control of the base motors. Both the control modes can be used for the asynchronous gaits. Figure 5 shows snapshots of animation of these two gaits.

A 4-phase pushing-forward motion is shown in shown in Figure 3. The tip of the manipulator can be designed to make a motion to move from left to right in a relatively up position, move down, and then move right to left in an down position, move up and repeat the motion in a closed curve

in the plane of the mechanism. This mode of motion is a Right or R-motion. Its reflection gives an L-motion. The other approach, called 3-phase lifting-up motion and shown in Figure 4, moves the tip to the center in an up position, retracts the left arm in a relatively low position and then retracts the right arm to a down position and, similarly, repeats in a closed curve. The motion, on the other hand, generates a Left motion or L-motion. Similarly, its reflection gives an R-motion.

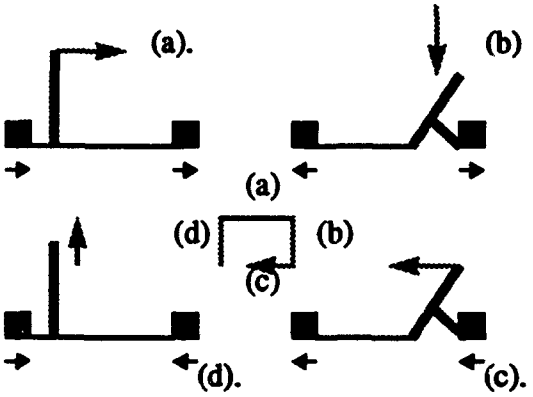


Figure 3. 4-Phase R-Motion

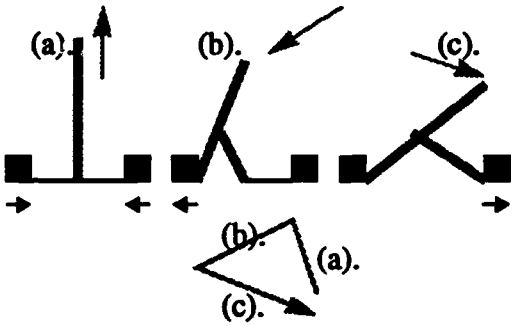
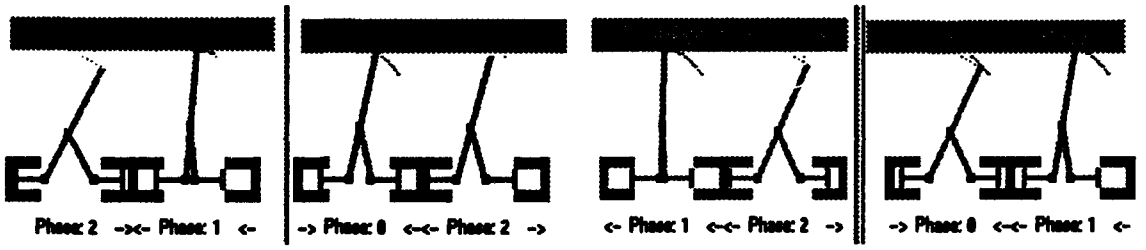
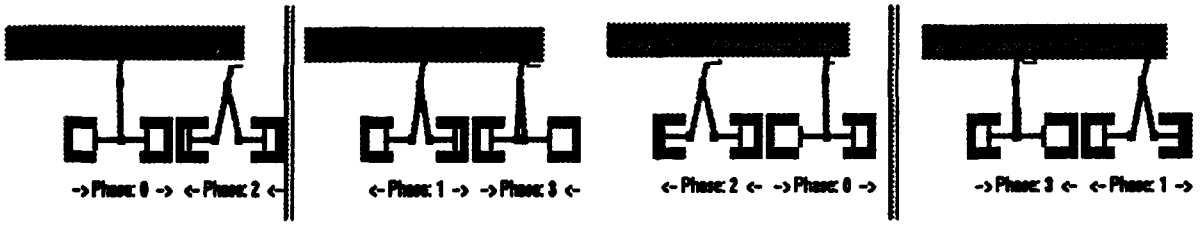


Figure 4. 3-Phase L-Motion



(a). 3-Phase Lifting-Up Design

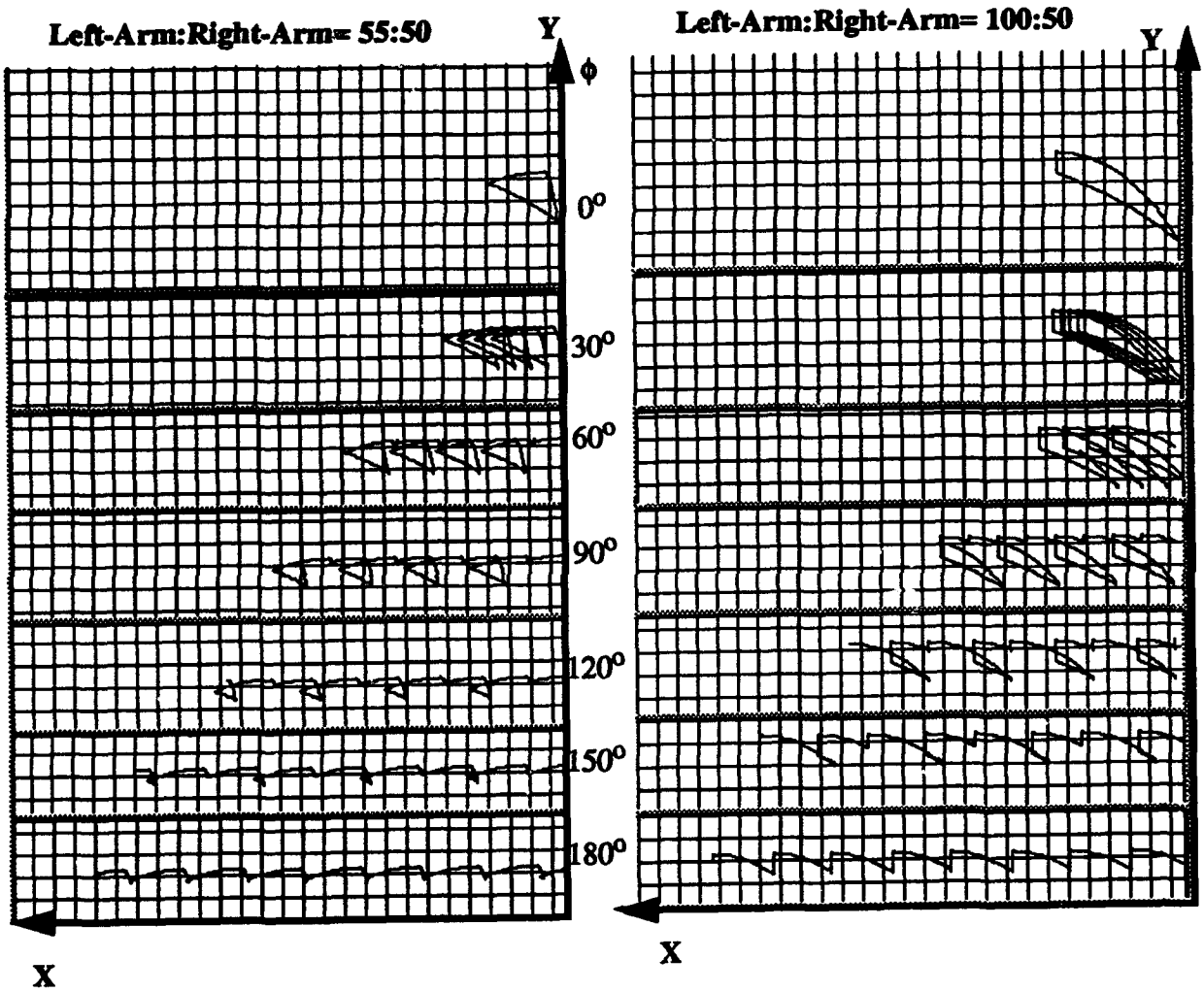


(b). 4-Phase Push-Forward Design

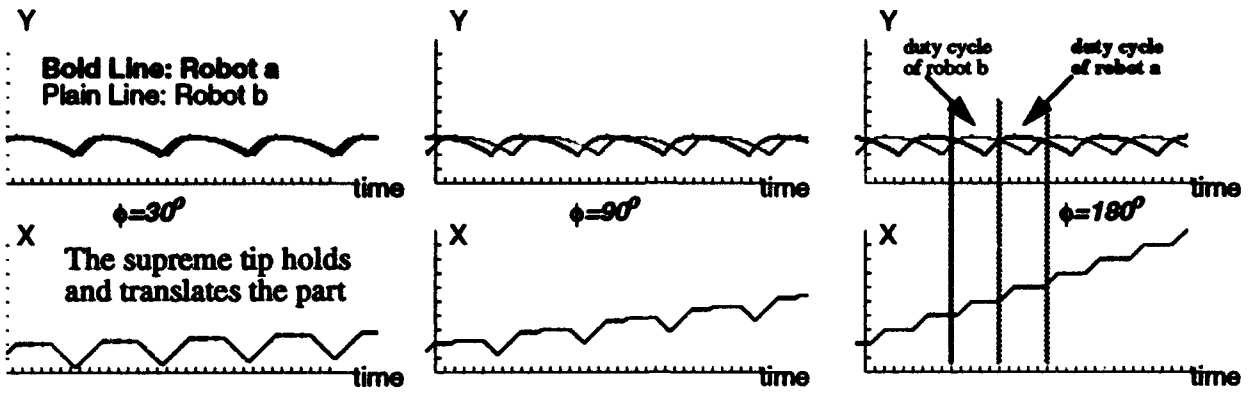
Figure 5. Animation of the translations

The 3-phase "lifting-up" primitive can translate any object efficiently, no matter if the tip of the manipulator is rough enough to push the object. In other words, the friction coefficient between the tip of the robot and the surface of the part is not a significant factor for the motion. The throw and efficiency of the translation are sensitive to the length of the two arms of the manipulator, and the

phase difference of the operation between the neighbor robots. Figure 6 illustrates the effects of changing the phase angle between the motion drives of adjacent LAMBDA machines and changing the LAMBDA arm length. The net throw and the duty cycles of a pair of adjacent robots can be observed in Figure 6. (b) which shows the y-motions of the tips of two adjacent robots and x-motions of the translated part respectively.



(a). The Tracks of the Motion of the Part in 4 Cycles of Operation



(b). The thacks in three various phase differences.

Figure 6. The Motion of the 3-Phase Gait Primitive

The closed curves labelled 0 degrees in Figure 6 (a). show that if all the LAMBDAs are operated synchronously, the part will not translate on the surface. At 30 degrees the part slowly drifts forward while making larger oscillatory excursions, while at 180 degrees the part moves forward smoothly. Tables 1 and 2 list some of the experimental results.

Table 1: The Speed and Efficiency of the 3-phase Gait (1).

arm ratio	max_throw	ϕ=0	ϕ=30°	ϕ=60°	ϕ=90°	ϕ=120°	ϕ=150°	ϕ=180°
left : right	units per cycle	Speed efficiency	Speed efficiency	Speed efficiency	Speed efficiency	Speed efficiency	Speed efficiency	Speed efficiency
50:50	15	0 0	3.96 13.20%	11.88 39.60%	19.01 63.38%	21.50 71.66%	26.01 86.70%	29.03 96.77%
75:50	23	0 0	4.41 9.59%	10.35 22.50%	16.88 36.68%	22.5 48.91%	26.32 57.21%	31.03 67.47%
100:50	31	0 0	3.78 6.01%	7.74 12.5%	14.5 22.4%	23.89 38.54%	26.32 42.44%	34.53 55.69%
125:50	46	0 0	4.23 4.60%	6.21 6.75%	10.17 11.05%	28.07 30.51%	28.07 30.51%	34.53 55.69%
150:50	61	0 0	3.60 2.95%	3.60 2.95%	7.56 6.20%	32.14 26.35%	28.37 23.26%	44.85 36.76%

Table 2: The Speed and Efficiency of the 3-phase Gait (2). (ϕ is 180^o.)

	50	60	70	80	90	100
max_throw	15	18	21	25	28	31
speed	29.03	29.54	30.33	31.24	34.76	34.53
efficiency	96.77%	82.06%	72.23%	62.50%	56.72%	55.69%
	110	120	130	140	150	160
max_throw	37	43	49	54	61	67
speed	34.97	39.97	39.13	39.47	44.85	45.30
efficiency	47.26%	40.67%	39.93%	36.55%	36.76%	33.81%

In the tables, ϕ is the phase difference (in degree) of the operations between two neighbor robots. Since a robot is operated in phase difference ϕ ahead of a neighbor one is the same as 360^o-ϕ behind of the neighbor, the phase difference is considered from 0^o to 180^o. The max_throw is the maximal displacement of the tip of the elementary robot in each cycle of operation. The speed is the throw of the part in each cycle. The efficiency is the speed divided by 2*max_throw, because the part can be translated at most 2 times of the max_throw in each cycle.
Given the fixed lengths of the two arms, the speed of the gait monotonically increases as the phase difference increases up to 180^o. In table 2, when the phase difference is fixed at 180^o, the speed increases as the length ratio of left arm to right arm becomes larger. But the efficiency, on the contrary, monotonically decreases due to the larger maximal throw. When the lengths of the arms are equal, and ϕ is 180^o, the efficiency is almost 100%.

The efficiency of the 4-phase "pushing-forward" primitive, on the other hand, is only slightly affected by the length of the longer arm of the manipulator; the part is simply translated by half of the motor excursion per cycle. The best efficiency can be acquired when the neighbor robots are operated in opposite phases.Unlike the operation of the 3-phase design, it is easy to operate two robots in 180^o phase difference to maximize the throw. The disadvantage of the design is that the maximum throw is limited by the motor excursion no matter how large the difference of the arms.

For both of the gait designs, there exists a greater force at the tip and a better stiffness of the arm if the length difference is small. Both of the designs can have optimal efficiency when the length difference is minimized. The throws of the two designs are about equal when the phase is 180^o. However, it is hard to make the 180^o phase difference which is 1.5 periods of one operation state. If the 120^o phase difference, which is easy to be generated for 3-phase design, is considered, the throw of the 3-phase design in each cycle is larger than the 4-phase design when its length ratio of the two arms is larger than 5:2. (Refer to table 1 in the row of 125:50). If the stiffness of the longer arm would not cause any problem, then the three-phase design will have larger throw when the length ratio of the arms is increased.

2.2.3). Planar Manipulators

The unit manipulator can be used to populate a two dimensional array in a variety of configurations. The array of manipulators essentially creates a force field, which is utilized to perform different operations on a part. In the remaining sections of the paper, the study concentrated on the force models of the array of planar manipulators, the stability condition of the oriented part, and a simulation approach to show the translation and rotation of the part located in the force field.

3. The Implementation of the Simulator

3.1) User Interface

A user interface for the simulation system was implemented in the C language and using the X library routines associated with the X Open Look tool kit on the Sun workstation environment. [XWIN] The user is given the ability to create a polygonal region on the surface of the array and label it with a force direction. The force directions have been arbitrarily chosen to be positive and negative X and positive and negative Y. Other directions are clearly possible. In the simulations described here, different force fields (labelled north, south, east, and west) are labeled with different textures (surface patterns with color backgrounds and arrows in four directions). Users can define force fields in terms of the polygon-shaped areas on the main canvas of the interface. In the main canvas of the user interface, each image pixel represents a unit manipulator and the square field area is the active region of two-dimensional array of the manipulators. Overlapping of the drawing of the fields is allowed in the graphics user interface, but only the topmost field shape (which is also the only visible region of the overlapped set) is used to compute the contribution to the forces. Parts are defined in another specific canvas using the similar drawing scheme. In this canvas, objects can be edited if required. After the editing of the object is finished, it is located in the field array on the main canvas. The location is arbitrary; the user can drag, drop and rotate the part manually to the proper position. Then, the translation of the object is performed step by step in this array.

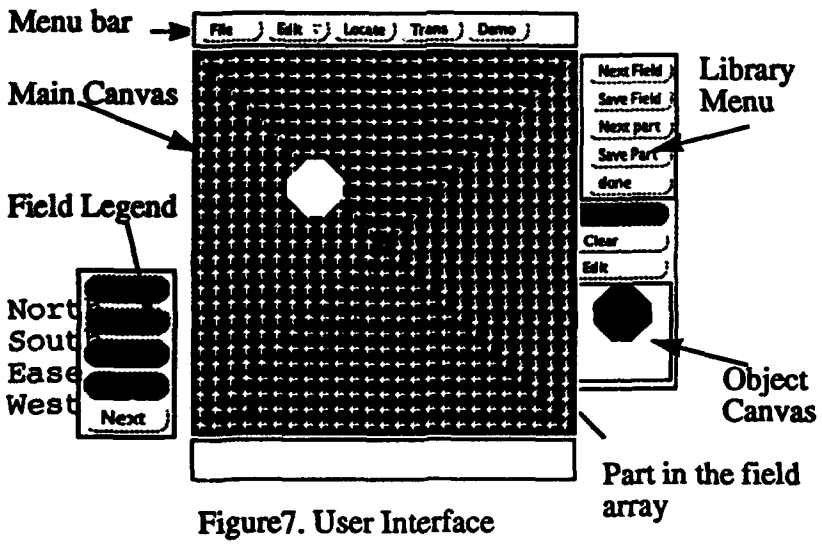


Figure7. User Interface

The technique of plane overlay in the computer graphics was introduced in the animation; it is an efficient approach to translate a part on the area without destroying the texture of the background fields.[XWIN] The field arrays and the parts defined by users can be stored Users can down load them and make any combination for different requirements. Figure 7 shows the user interface, the arrows show the direction of the force fields.

3.2) Force Model: A Static Analysis

A static analysis is applied to the current model. The later models will use rigid body dynamics. The following assumptions are made for the modeling process: The unit robot imparts a force to the carried object.

a. The unit robot imparts a force to the carried object.

b. The mass of the part is uniformly distributed; each micro-robot holding the part give the same force contribution to it.
c. The object is translated and rotated in discrete fashion; the elementary manipulators, at most, move the part for 1 unit step and/or rotate it for 1 unit angle in each cycle.
d. The initial and final velocity in each micro-cycle are both zero; there is no momentum of the part.

e. The intersections of each field and the object placed on the field array conceptually separate the object into several sub-regions with each region generating a net force and a net torque with respect to the center of gravity of the object. The object is stable when the summations of both the net forces and net torques contributed by all object/field intersections are zero.

The distributed forces, based on the above assumptions, resulting from the manipulators are summed up cell by cell for all micro-robots holding the part. The torque with respect to the center of mass of the part is similarly computed. Equations 1 and 2 list the basic formulas.

$$\hat{F} = \sum_{\forall i} \hat{f}_i \qquad \text{(Equation 1)}$$

$$\hat{\tau} = \sum_{\forall i} \left(\hat{f}_i \times \hat{l}_{cmi} \right) \qquad \text{(Equation 2)}$$

* where F is the vector of total force, f_i is the individual force vector contributed by the manipulator i.
* τ is the vector of total torque of the part, $l_{cm\,i}$ is the vector from mass center to the manipulator i.

Given the net force and torque generated by the force field, the motion can be described and measured as follows.

$$(F = ma),\ (\tau = I\alpha) \qquad \text{(Equation 3)}$$

$$\left(s = v_0 \times t + \frac{1}{2} \times a \times t^2\right), \left(\theta = \omega_0 \times t + \frac{1}{2} \times \alpha \times t^2\right) \qquad \text{(Equation 4)}$$

Motion in the physical world will be very complex due to a variety of effects including chip surface roughness and force available from the micro-robots and the model may be different when other elementary micro-manipulators are used. A more precise, physically realistic, model may need to wait for experimental experience for validation and understanding.

A simplified motion model is used here. Two thresholds are set proportionally to the mass and the moment of inertia of the part for the force and torque respectively for simulation convenience. If the total force exceeds the force threshold, the part will be moved for one unit distance along the dominant direction. Similarly, if the total torque exceeds the torque threshold, the part will spin for one unit angle.

3.3). Simulation Approach

There is trade-off between accuracy and speed of the simulation; two different levels of details of the simulation have been designed; they perform image level and graphics level simulation respectively.

3.3.1) Image Level Simulation

In the image level simulation, under the assumption of the force models in section 3.2), the distributed forces and torques resulting from the manipulators are summed up pixel by pixel by detecting the overlapped region and the textures of the field array and the part.

The image processing function calls are available in the X window system. The polygon fields are defined and mapped to the main canvas, the part is also defined and overlaid to the main canvas. The overlaid region in the canvas represents the active region of the force field array that will affect the motion of the part. The bit patterns of the pixels in the superposed region give a way to detect the superposition and recognize the direction of the field vector so that the total force and total torque can be calculated pixel by pixel. Figure 8 shows the subroutine for the purpose. The advantage of the system function calls is the convenience that the programmers do not have to deal with the flood/boundary fill-in algorithms for the polygons fields and parts in the paradigm of high level language. The disadvantage is the fact that the screen image access speed is by no means satisfactory as the workload increases; a lot of redundancy in the image processing could be avoided by means of computational geometry methods.

```
Force_Torque(w, start_x, start_y, end_x, end_y, cmx, cmy)
  Widget  w;
  int start_x, start_y, end_x, end_y, cmx, cmy;
{
  XImage  *image;
   char *pixmap;
   int pixel;
     long force_x=0,force_y=0;
   long torque=0;
    register i,j;

  for(i=start_y;i<=end_y;i++)
    for(j=start_x;j<=end_x;j++) {
            pixel=XGetPixel(image,j-start_x,i-start_y);
       switch(Back_Masks(pixel)) {
          case NORTH:       force_y-=Delta_force; torque+=(j-cmx)*(Delta_force); break;
          case SOUTH        force_y+=Delta_force; torque+=(j-cmx)*(-Delta_force); break;
          case EAST:        force_x+=Delta_forc  Delta_force; torque+=(i-cmy)*(Delta_force); break;
          case WEST:        force_x-=Delta_force; torque+=(i-cmy)*(-Delta_force); break;
       }
    }
  }
```

Figure 8. Image-Level Force / Torque Calculation

In most motions, only the pixels around the contour of the part will change between iterations. Nevertheless, the change is hard to detect especially when the shape of the part is very complex. This makes graphically based simulation preferable.

3.3.2) Graphics Level Simulation

A graphics level alternative was developed to reduce the computational workload. The model is based on computational geometry schemes including polygon decomposition, polygon clipping [SUTHERLAND74], and polygon triangulation algorithms. The polygon shaped fields are decomposed into a convex set t. The decomposed convex polygons then behave like windows to clip the "visible" portions of the part. The portions of the part are then triangulated in order to estimate the force and torque contribution individually. (Refer to Figure 9.)

Triangulating the portions of the part is used because the mass center of the triangles can be derived directly by measuring the intersection of two segments s1 and s2 in Figure 9(c). Each of them is the connection between one vertex of the triangle and the medium point of the opposite edge. Equation 5 and 6 are the formulas for calculating total force and torque in the graphics level simulation. Figure 10 shows the accumulation of the total force and torque contributed by each triangulated sub-region of the object/field intersection.

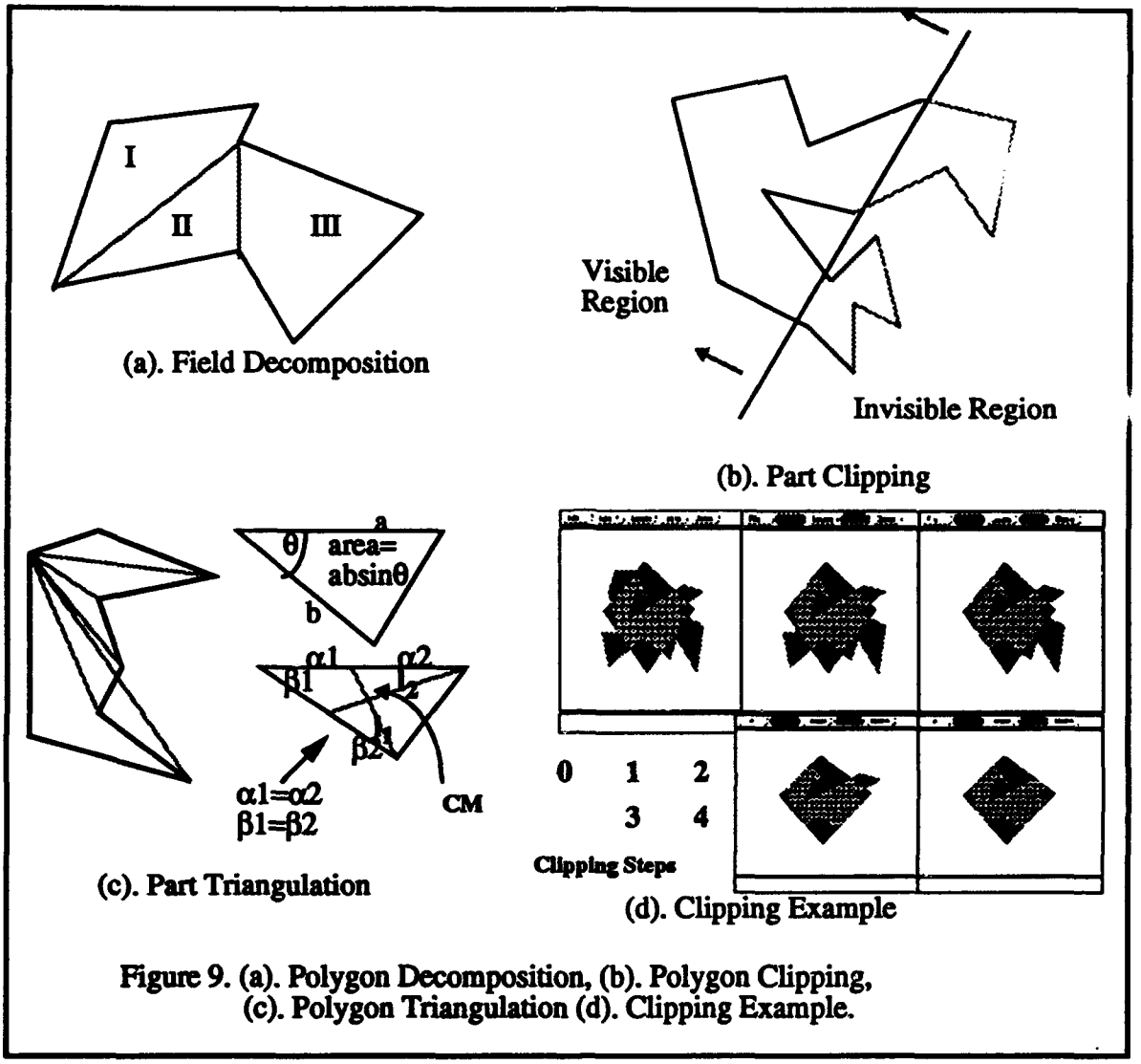


Figure 9. (a). Polygon Decomposition, (b). Polygon Clipping, (c). Polygon Triangulation (d). Clipping Example.

$$\hat{F} = \sum_{\forall triangles} \hat{f}_{(i)} \cdot Area_{triangle\,(i)} \qquad \text{(Equation 5)}$$

$$\hat{\tau} = \sum_{\forall triangles} \hat{f}_{triangle\,(i)} \times \hat{l}_{CM-cm\,(i)} \qquad \text{(Equation 6)}$$

* where F is the vector of total force, f_i is the unit force vector of the triangulated region i.

* τ is the vector of total torque of the part, $l_{CM\text{-}cm(i)}$ is the vector from mass center of the whole part to the mass center of the triangulated *region i.*

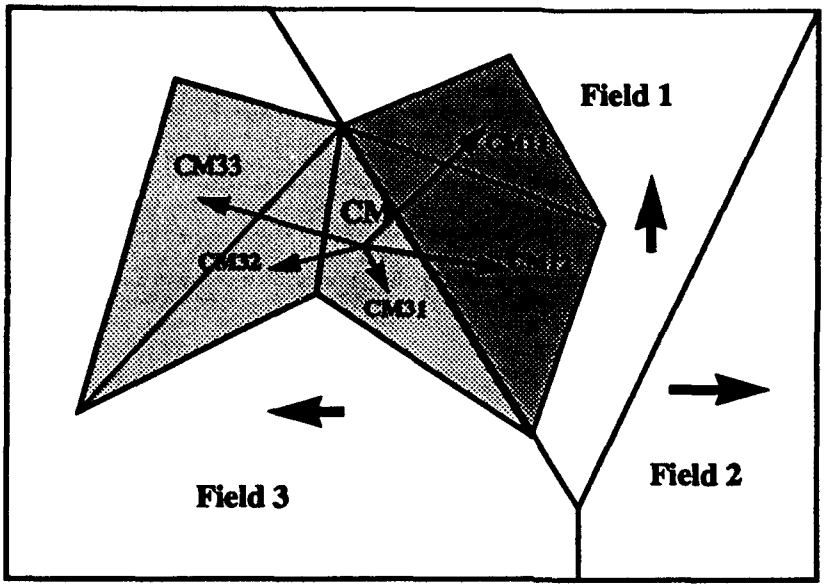


Figure 10. Graphics Level Calculation

The graphics level simulation is much more efficient but less flexible; it increases the speed for calculation in the simulation, but, when a more sophisticated analysis is required, say, the realistic friction force in each manipulator is considered, the force model will become more complicated and the computation in the graphics level simulation may need revision.

3.4). General Notes on Stability

The general stability property for a simple arrangement of motion cell is that the area of the part on each side of the boundary should be equal and that the line joining the centroids of the areas of the part on either side of the line of demarcation of the motion domains should be perpendicular to that line, i.e., force and torque should be zero. For more complicated configuration of the field array, it is obvious from Equation 5 and 6 that the stable condition exists when the areas of the fields imparting forces in opposite directions are equal and also the net torque of the cross product terms of each triangulated region in equation 6 is zero.

4. Experiments and Results

Several experiments useful in micro assembly were designed and executed on the simulation. The pictures illustrate the operations of the experiments.

1). Cetering a part until stable.

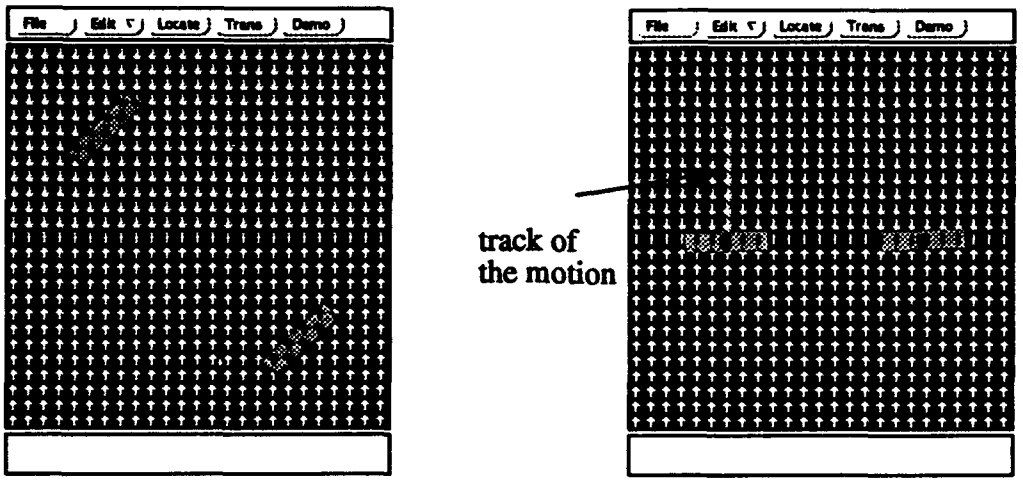


Figure 11. Centering Parts till Stable

In Figure 11, any part of suitable size placed anywhere on uniform motion part of the plane (say all Up or all Down) translates until an edge crosses a boundary into a different motion domain. At this point the motion is a combination of translation and rotation until the part migrates so that the center of the supported area (i.e., the simply or multiply connected area in contact with the micro-manipulators) sits on the boundary of the Up and Down-motion areas. The general stability property for this arrangement of motion cells is that the area of the part on each side of the boundary is equal and that the line joining the centroids of the areas of the part on either side of the line of demarcation of the motion domains is perpendicular to that line, i.e., net force and torque are zero.

2). Spiraling or Rotating a part

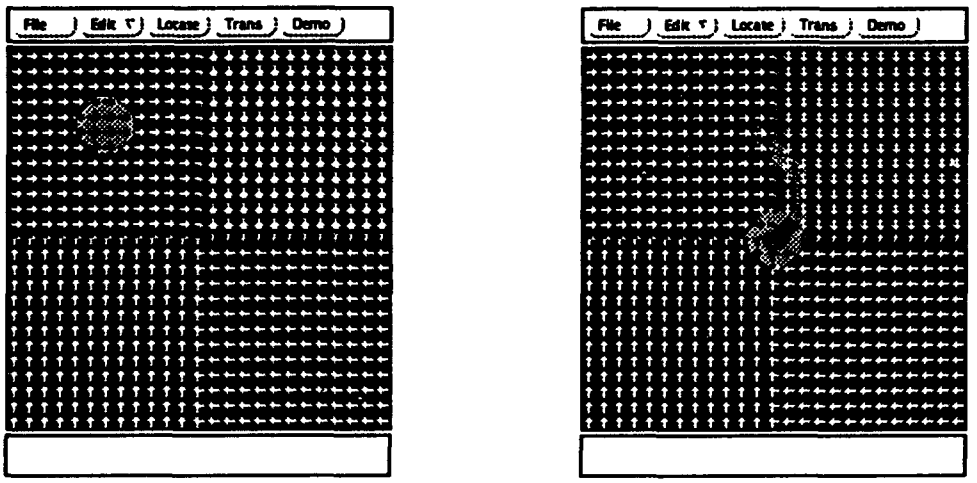


Figure 12. Spiraling a part to center

Three geometric patterns of L,R, U and D-motion fields will spiral or rotate a part in different fashions.

a. Spiraling a part to center in a convergent field array. (Figure 12)
The whirlpool-shaped field array is convergent since each segment of the circulated fields sweeps the part, without translation, to a position just inside the next orthogonal field which sweeps the part closer toward the center of the array.

b. Spiraling a part away from the center in a divergent field array. (Figure 13)
This configuration, on the contrary, moves the part in outward manner. The center is a position of unstable equilibrium. A part placed at the center in a conditionally stable state migrates to the boundary under slight disturbances.

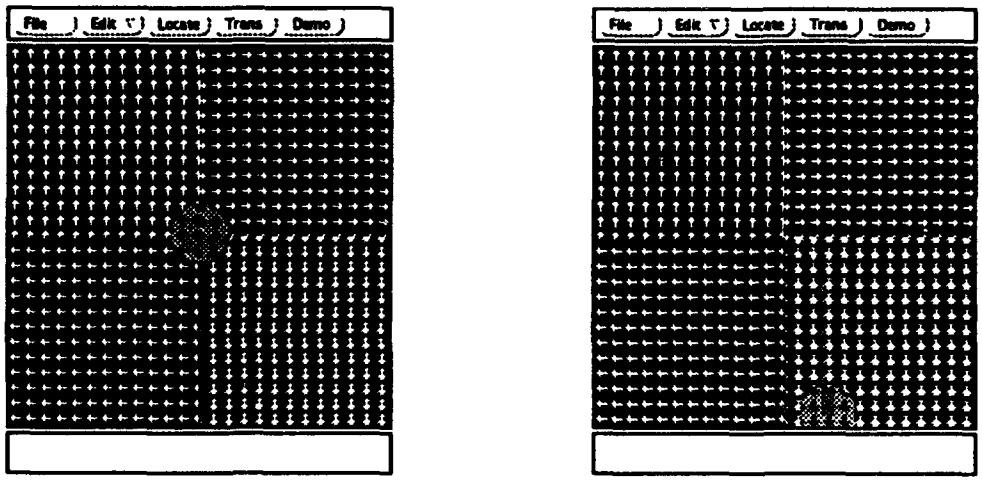
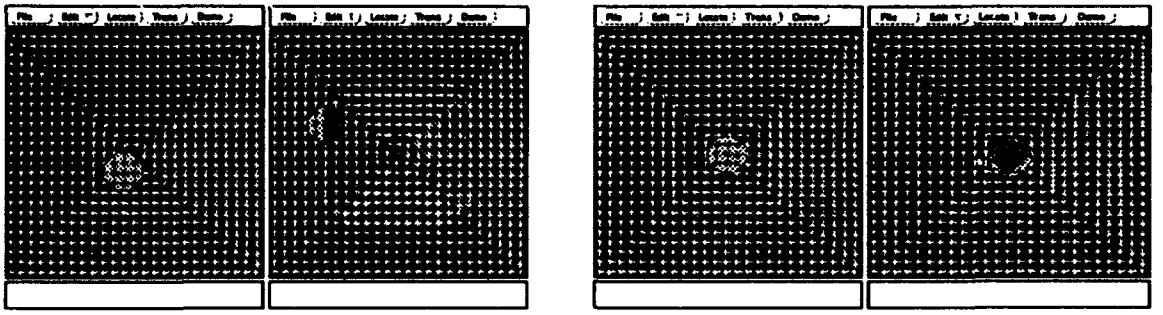


Figure 13. Spiraling a part away from the center

c. Rotating a part about the center in four triangular fields. (Figure 14)
In the arrangement shown, the boundary edges and the field directions are not orthogonal and it is not obvious if the rotation is convergent or divergent. Figure 14 (a). illustrates a divergent case. However, if the part is almost symmetrical and placed in the center of the array, experiments showed that even when a very long term translation was executed, the center of mass of the part was still moved within a small cycle among the center of the field. (Figure 14 (b).) Further work needs to be done to determine whether this is an intrinsic feature or due to the interaction of quantization effects or due to the part shape.

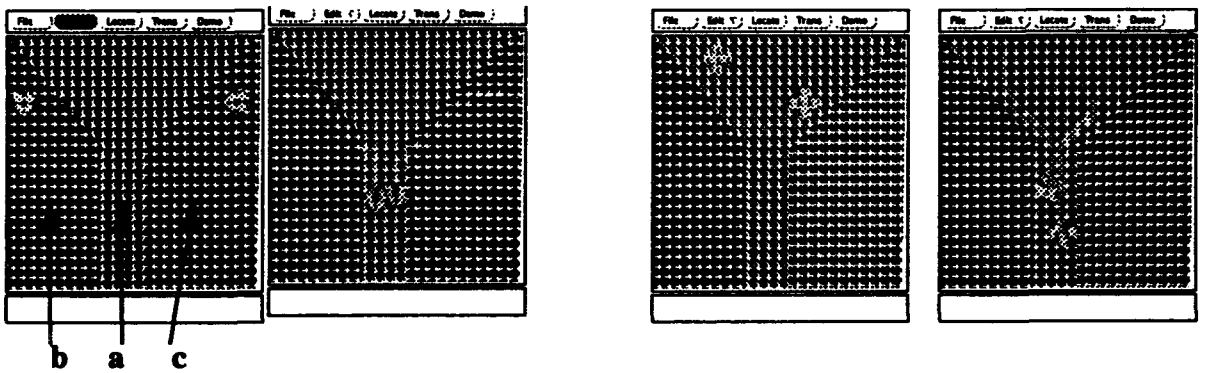


(a). (b).

Figure 14. Rotating a part about the center .

3). Spatially filtering a part

Figure 15 shows a a spatial filter. The area **a** can be made of type D orientation and the two areas labelled **b** and **c** of type R and L respectively. A part placed entirely in the funnel-shaped area **a** migrates downward until it enters either area **b** or **c** where it receives a force causing a torque to align the part along the slanted sides. It will then slide down the side until it reaches the output slot at the bottom of the diagram. Here because of the way the picture is drawn, the system acts in a manner reminiscent of the action of a fluid under gravity.



before filtering filtering

Figure 15. Spatially filtering a part

4). Translating a part in a complex path. In Figure 16, a complex path was arranged with walls (using fields operating in opposite directions) orthogonal to the path to constrain the track of the translation.

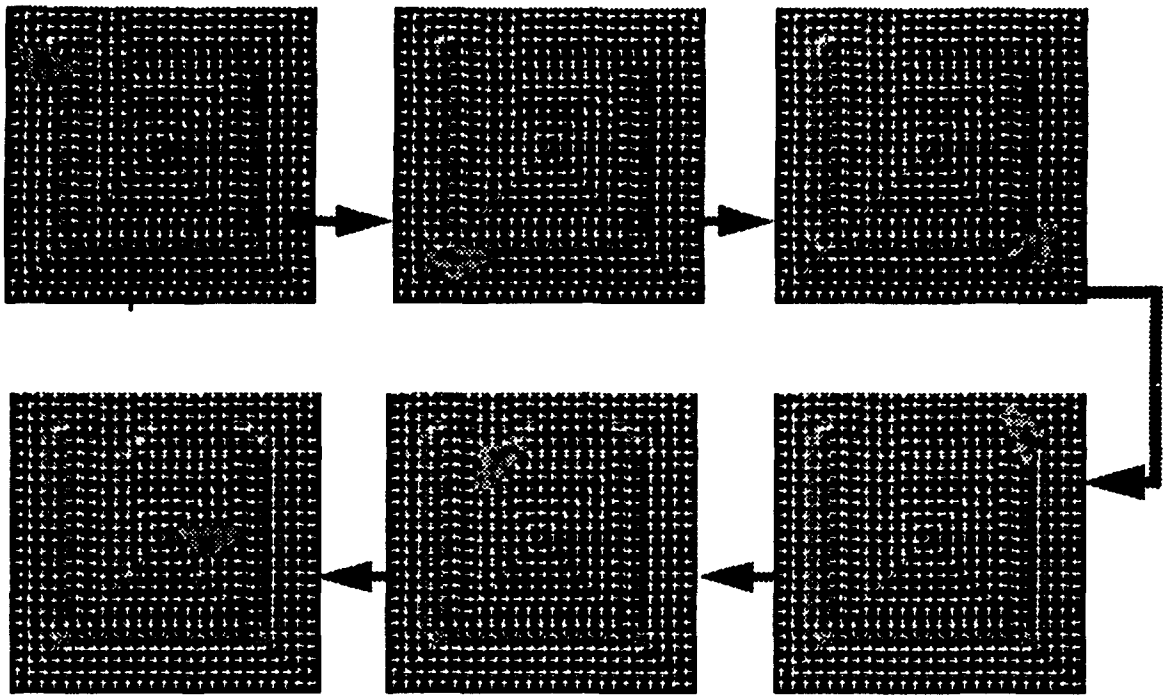


Figure 16. Translating a part in a complex path.

5). Orienting a part in a programmable field array.

So far, the discussion has centered on arrays that can be arranged at compile time, that is the force directions could be built into the silicon as a single complex field array. Instead of designing a single complex field array to translate a part, another perspective is to make the array dynamically programmable. Figure 17 shows the idea of scheduling (programming) a sequence of basic operations to translate and orient a part for assembling purposes

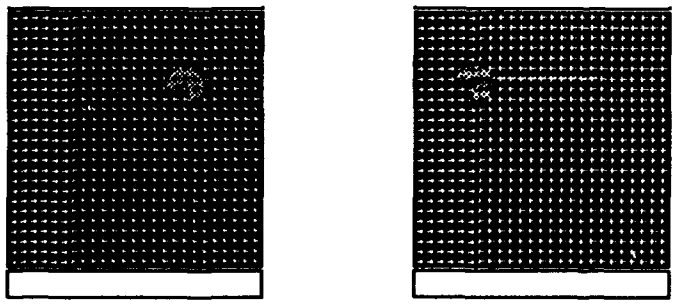
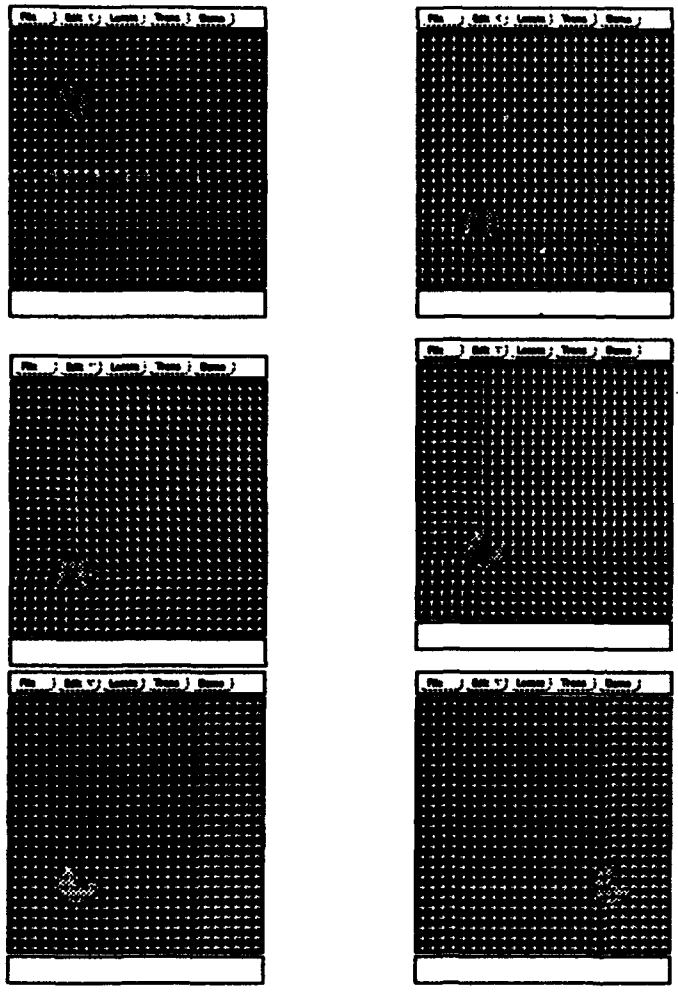


Figure 17. Orienting a part in a programmable field array.



Step 2:

Aligning the Part to a Horizontal Boundary

Step 3:

Rotating the Part about Its Center of Mass for a Certain Angle

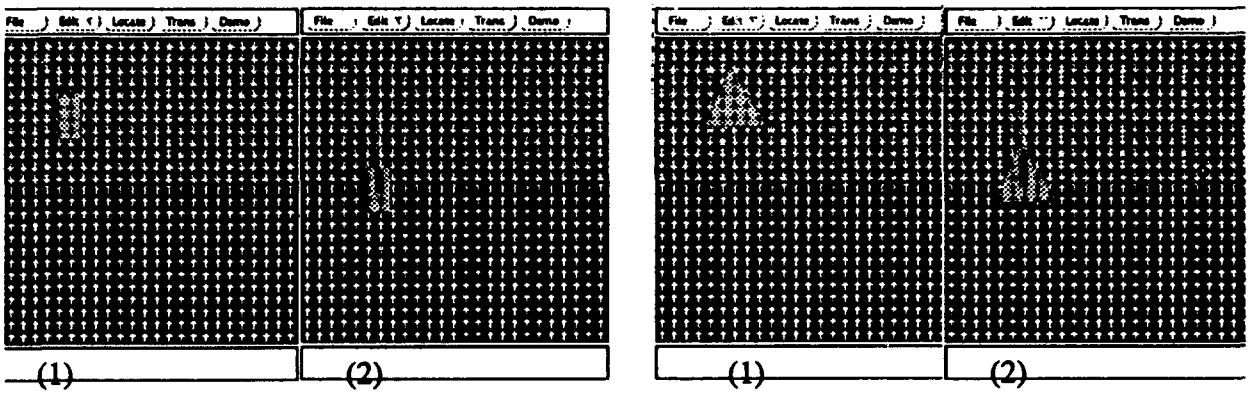
Step 4:

Transmitting the Oriented Part out of the Workbench

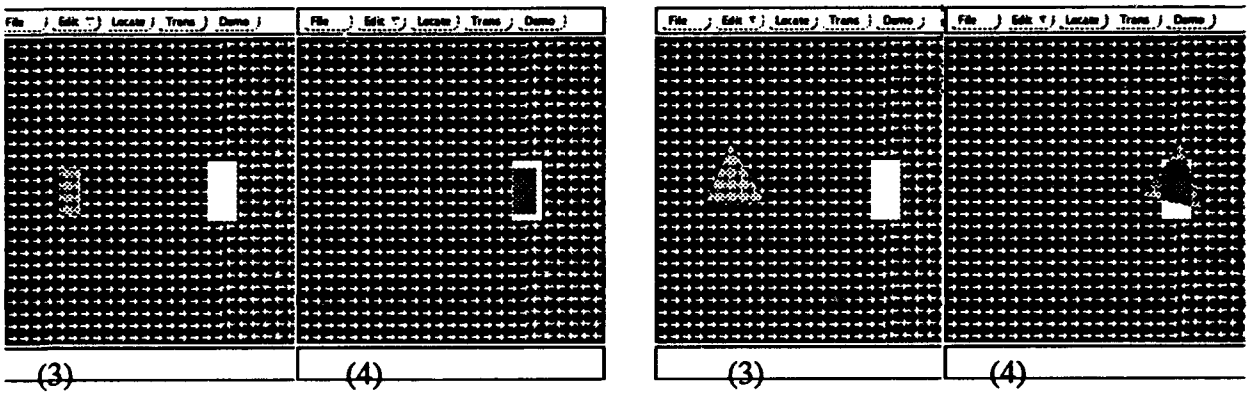
Figure 17. (Continued) Orienting a part in a programmable field array.

6). Sorting two parts using a hole

The idea is that a hole may be used as a trap for a part being translated past it. The shape of the hole set at a boundary of fields should be geometrically similar and a little larger than one of the parts being sorted. A part moving past the hole would fall into the trap while another part, larger than the hole, would not be perfectly overlapped with the hole and would pass by perhaps changed in direction but not captured. The trapped part can then be moved by reprogramming the array to moved as desired. Thus, two parts with different shapes can be discriminated: the operations perform a sorting function of the field array. Notice that before the stage for matching, trapping and sorting, the parts need to be well aligned and oriented. Otherwise, even the part is similar to the hole in shape, it may not be trapped. Figure 18 shows the idea.

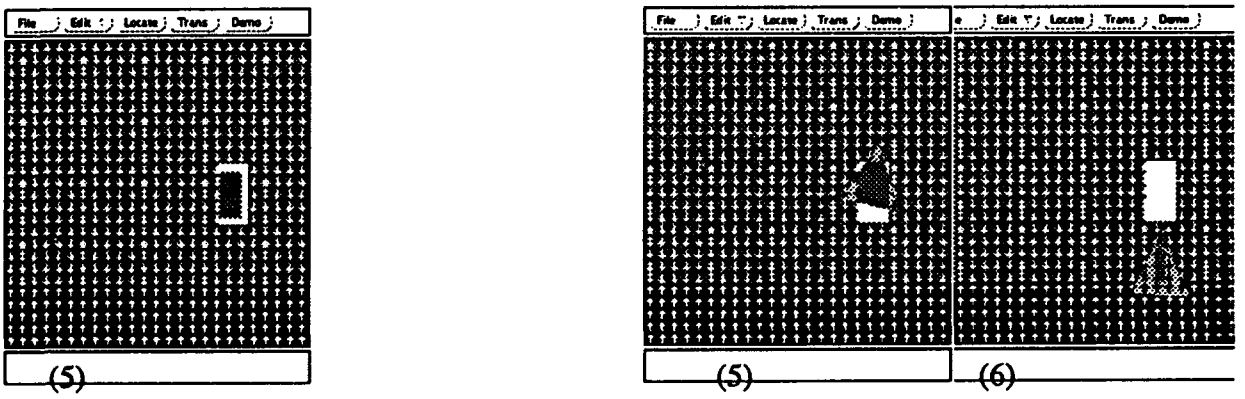


Step 1: Aligning the two Parts for Matching



(Success) (Failure)

Step 2 :Matching and Trapping



Step 3: Discreminating

Figure 18. Sorting two parts using a hole

5. Conclusions and Future Research

A simple 4-bar linkage λ machine was used as the unit manipulator. Under control of the two base motors, with various combinations of the motion states (forward, backward, and still) of the two arms of the robots, the robots perform different "gaits". Two efficient gaits were shown for the translation and rotation of parts.

A simulation system with its associated user interface has been designed. Simple incremental motion modalities were simulated,. The simulation will be extended to work on the dynamics of rigid bodies.

The experiments showed the basic operations of the miniature work bench. A sensorless micro-manipulator system, as demonstrated above, can translate and orient a part in a variety of ways under program control. The functionality of the work bench can be, for instance, applied to the chip-positioning in an MCM; the advantage is that the assembly job can be operated in parallel if well scheduled. However, an open loop control scheme does not keep track of the location of the part placed on the array and will cause difficulty and inefficient operation in controlling parts for complicated automatic assembly applications. It is hard to predict when the part is oriented to the destination position, and whether it is rotated well enough for further assembly work. This difficulty could be avoided by operator intervention in a manual-assist mode but addition capability may be needed for automatic operation. The easiest addition may be to add a sensory to the micro-robot cell; e.g. a strain gauge on the main beam or a photocell on the base of each cell. The sensor would detect the presence of a part in contact with or occluding the cell and be used to allow much more flexible control.

The work here has assumed that each cell is individually controllable in mod and direction.This puts a functionality, power and area burden and challenge on the design of the control processor required to drive each individual machine.

The main issues for future study are:

a. A study of different mechanical structures for implementing the unit robots.

b. The adjustment of the dynamics model to suit the operations of the specific unit robots.

c. The investigation of friction effect in both robot-to-part and part-to-part contacts required for automated assembly studies.

d. The inspection of the error tolerance in the micro assembling work.

e. The design of an array control language: the integrated system must provide a simple control language and the corresponding interpreter or compiler for loading the specific area of motion of a specific pattern of gaits in order to support the programmability of the array and its use in applications.

f. The design of the control processor residing in each cell.

[References]

[IBM93] Personal communication. IBM in the mid-70's built a wafer fab line in which the machines and transport mechanisms were enclosed in local cleanroom conditions. Wafer transport between machines was by air levitation with lateral and rotational control by directed airjets.

[PISTER90] K.S.J. Pister, R. Fearing, R. Howe, "A Planar Air Levitated Electrostatic Actuator System.", IEEE workshop on MEMS, pgs 24-71, Napa Valley, CA, Feb 1990.

[PISTER92] S.R. Burgett, K.S.J. Pister, R.S. Fearing, "Three Dimensional Structures made with Microfabricated Hinges." Micromechanics ASME Winter General Meeting , Anaheim, CA., 1992.

[FUJITA94] M.A. Ataka, Omodaka, H. Fujita, "A Biomimetric Micro Motion System." Transducers Digest International Conference on Solid State Sensors and Actuators, pgs 38 - 41, Pacifico, Yokohama, Japan, June 1993.

[BOHRIMGER94] K.F. Bohringer, B.R. Donald, R. Mihailovich, N. MacDonald, "A Theory of Manipulation and Control for Micro Frabricated Actuator Arrays." to be presented at IEEE workshop on MEMS, Osio, Japan, January 25-28, 1994.

[GOLDBERG93] K.Goldberg, "Orienting Polygonal Parts Without Sensing." Algorithmica, Vol. 10 (2/3/4), pgs. 201-225, August/September/October 1993.

[XWIN] E.F. Johnson, K. Reichard, "Professional Graphics Programming in the X Window System", 1993

[SUTHERLAND74] I. E. Sutherland, G. W. Hodgman, "Reentrant Polygon Clipping", Communications of the ACM, Vol17, no.1, Jan., 1974.