DRAFT

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence
Memo. 165

September 1968

Description and Control of Manipulation by Computer-Controlled Arm

Jean-Yves Gresser

The immediate purpose of the research on Intelligent Automata is to have an autonomous machine able to understand uncomplicated commands and to manipulate simple objects without human intervention.

This thesis is concerned with the programming of a special output device of the present machine existing at Project MAC: an arm with eight degrees of freedom, made of our identical segments. Classical approaches through hill-climbing and optimal control techniques are discussed.

However a new method is proposed to decompose the problem, in an eight-dimensional space, into a sequence of subproblems in spaces with fewer dimensions.

Each subproblem can then be solved with simple analytical geometry. A simulation program, which applies this method, is able to propose several configurations for a given goal (expressed as a point in a five-dimensional space).

TABLE OF CONTENTS

FIGURES

ANNEX

ACKNOWLEDGEMENTS

My stay in the United States was made possible by a grant of the French Foreign Office and the support of the Centre National d'Etudes de Telecommunications, under the direction of Mr. J. Libois.

I would like to express my gratitude to Professor Minsky for his supervision of this thesis, and to Professor Papert. Many ideas expressed in this thesis were developed over several fruitful discussions I had with them.

I also wish to thank Annette Herskovits, Tom Binford, for their permanent assistance, Michael Beeler, for his introduction to the subtilities of the new arm, the whole AI group of Project MAC for providing a sympathetic environment where this work could be carried out, the Biology group, and Randle Ware for his adaptation of this text from a rough French-English version.

As An Introduction

In the next few years man will step on the moon, and think of going further. But, preceded by machines, he may not be the first explorer on Mars or Venus. To build these machines is part of the challenge for artificial intelligence and advanced programming, a challenge to every level of engineering.

The transit time and the bandwidth limitations will restrict the communication between machine and man. The remote device must have a large autonomy of decision in acting and in transmitting data to earth. Such autonomy requires the ability to analyze the environment and from these observations to plan a sequence of actions which will alter the environment with respect to the machine.

Two perceptibly different approaches could be taken. One considers such a device as a remote manipulator, and the other is a research on intelligent automata; equipped with special purpose output or input hardware [11] [13]. One may use a particular technique of the other but they differ by the leading idea.

Remote manipulation - the task involves a human operator and a machine. It is most practical when the environment is not too distant and has a geometry which is familiar to man. Research was initiated after World War II, when it became necessary to use a remote manipulator in experiments using radioactive materials. Most of the present manipulators are not different from the early models.

One trend is toward man-machine symbiosis [4]. Using force-feedback and spatial correspondence, one hopes that the sense of remoteness will disappear. Time delay is an important limiting factor, in such a conception.

The other tendency [15] calls for a "Supervisory Controlled Manipulator." The manipulator and the task site are considered as a system to be controlled by an operator, aided by a computer. The remote device may be equipped with its own small computer, able to make quick decisions under small changes in the environment, to interpret commands and to send back data from the sensors.

The validity of the results which will be obtained in studying remote manipulation seems to me limited in time. The goal in such research is to have a man-machine system, where each component assumes the part of the task for which it is the best adapted. I suspect we know a lot about man but very little about the machines. By machines I mean not only the computers we have now but also the ones we will have in the next five years. An extrapolation of the progress, made in Computer Technology during the last few years, invite us to think that machines of decreasing size will be able to assume more and more of the task. Thus a balance between man and machine is a result which may be soon outdated, because of the new machines.

Intelligent automata - It seems to me that the limitations imposed by the task itself lead to a type of approach which relies essentially on the machine alone [11] [11].

For an artificial explorer on Mars or Venus, the time delay would make uselessly slow any system using anything than general, goal setting commands and transmitting to earth anything other than the essential elements of the environment.

Research on intelligent automata started at M.I.T. with the pioneer work by H. Ernst [1] as early as 1961, on a "computer-operated mechanical hand." The major effort began August 1965.

The purpose of such research is explained in the first progress report [#I], August 1966:

"[the] goal is to develop techniques of machine perception, motion control, and coordination that are applicable to performing real word tasks of object-recognition and manipulation...[the] aim is to have a computer controlled system accept a relatively uncomplicated command and without human assistance, locate, grasp and assemble parts of a simple mechanical device."

The components of the ROBOT Project - At the present time the system is composed of:

- a large, general purpose computer (DEC) PDP6, which can deal with the generality and the flexibility required at this early stage of the project. Certain functions may be assumed later by special purpose hardware components.
- special input and outpute devices: a TV camera, an image dissector for controlled scan analysis, tactile sensor, several oil-powered manipulators, more or less versatile.
- programs to analyze the visual scene, control the motion of the hand, plan and control the overall activity. [#s] [13].

Programming the new arm - This thesis is concerned with the programming of a special output device - the arm MA-3[#I], the "new arm." It comes as the third arm, after Ernst's and the AMF verstran arm.

It was initially designed with four joints each of these having two degrees of freedom. It has actually three whole joints, and one half-joint at both ends, which still makes eight degrees of freedom. Its modular design makes it extensible.

The design is original with respect to any other arm actually available.

The eight degrees of freedom are supposed to give the arm a great versatility, in particular, the ability to move around objects. The multiplicity of solutions, for example, to reach a point within the range, makes it difficult to find any solution at all.

In 1967, M. Beeler [[illegible]] developed a "hill climbing" or error reducing iterative program. The method he has been using has not proved fully satisfactory, in his opinion, essentially because of the irregularity of the convergence of the process - and it seems also because in an attempt to find a solution he gives up the possibility of finding more than one solution.

I have set his program and certain research in molecular biology [6,7] as the starting point for a reflexion on the methods used generally to control an object or a model with several degrees of freedom.

Chapter II stresses what good may be expected from these methods and what are their limitations. In Chapter III, the problem in an eight-dimensional space is decomposed into a sequence of subproblems in spaces with fewer dimensions. Each subproblem essentially uses simple analytical geometry.

The method has been tested in a simulation program (written in LISP), and real world experiments. It is apparently more successful than the previous ones.

Chapter II

Critique of the existing methods

The parameter space will also be called solution space, and the goal is given in a goal space, with a number of dimensions equal to or less than that of solution space.

One may assume that the initial description of the arm is an application of the solution space onto the goal space, G=A(S). Application, which may be obtained from the description of the joints and of the segments. Our problem is to find B, application from the goal space into the solution space such: $B^{-1} = A$.

Where the number of parameters, or degrees of freedom, exceeds three, it is difficult to find for B an expression valid over the whole goal space, even when A is given by an analytical expression (then complete). To find numerical values with a relaxation method does not seem applicable as the number degrees of freedom increase.

When the goal space has fewer dimensions, B is not unique. Is it possibl to describe the multiplicity of B's and to choose among them?

Hill-climbing techniques give up the overall knowledge one may have on A, and expect simplification from local considerations.

Optimal control theory attempts at the same time a means to finding a solution and the best one according to a chosen (how arbitrarily) cost interior.

<u>Hill climbing</u> - A solution to a "hill-climbing" problem is a set of values $x_{10}, x_{20}, \ldots, x_{20}$ which minimizes (or maximizes) the value of a function $F(x_1, x_2, \ldots, x_u)$. $F(x_1, x_2, \ldots, x_u) - F(x_{10}, x_{20}, \ldots . x_u)$ can be interpreted

as a "distance" to the goal. The hill-climbing technique consists in starting from one point in the space (X_1xX_2---xXu), and exploring locally around that point, in moving in the direction of the steepest variation (either down, if one seeks a minimum, or up).

There are two cases in which the method fails. If the hill-climber reaches the region of a local extremum which is not absolute, and if the step by which it moves in the space is small compared to the dimensions of that region, it will certainly be trapped. The "mesa-phenomenon" [12] in which the space (X_1x ---xXu) is composed of large regions where F does not vary, separated by smaller ones where F changes for any of the parameters, may also occur.

The "mesa-phenomenon" seems unlikely to occur when there are few parameters. For a well designed device - one may expect a change for any of their variations. One still has to worry about local extrema, real or apparent. A point may appear as an extremum along the directions of exploration, but not along other directions (such is a saddle point). The only program existing for the new arm [[illegible]] combines hill-climbing methods with the use of language multipliers. Apparently all the apparent extrema are often of one type and it is easy to get rid of them. But there are still cases where the program gets trapped at points which are not easily identifiable; these are probably local extrema. I suspect this is due essentially to the physical limitations to the mobility of the different parts of the arm:

the point (x_1, x_2,...,x_u) is restricted to some region of the solution space, a value of F on the limit may appear as a local extremum when the real extremum (which is probably absolute) is outside the allowed region.

There are unfortunately few facts to support this claim. One is a counter example on a similar problem: the manipulation of computer models of proteins. The other will be an illustration with a two-bar plane linkage.

Two-bar plan linkage - A mechanism operating in a plane needs at least two degrees of freedom to reach any point in a certain two-dimensional portion of that plane. If it has a fixed point it must be made of two levers, at least. The arm, we consider, is of that simplest type, made of two levers of equal lengths (l), to avoid any superfluous complication, rotating about a shoulder and an elbow, each with one degree of freedom.

A natural definition for F is the euclidean distance from the hand (H) one of the extremities of the second lever, to the goal (g). The state of the arm is described by θ_1 and θ_2, the angles of rotation of the levers around the shoulder and the elbow, respectively.

Given a goal it is possible to represent the corresponding variations of F by the curves of equal distance in the (θ_1, θ_2) plane. There are three families of figures, depending on the ratio of the distance from the shoulder to the goal, to the common length of the levers. The curves show an expected double periodicity $(2\pi, 2\pi)$ and a symmetry with regard to a θ_2-axis, which absciss θ_0 is the value of the angle between the origin axis, in the original plane, and the line from the shoulder to the goal. For each period F has three extrema, an absolute maximum corresponding to $\frac{HG}{l} = 2+SG$, and two absolute minima F=0. The existence of the two minima is an interpretation of the fact that, given any point in the original plane within a circle of radius 2l, there exists a solution made of two couples of values such as $(\theta_{10},\theta_{20})$.

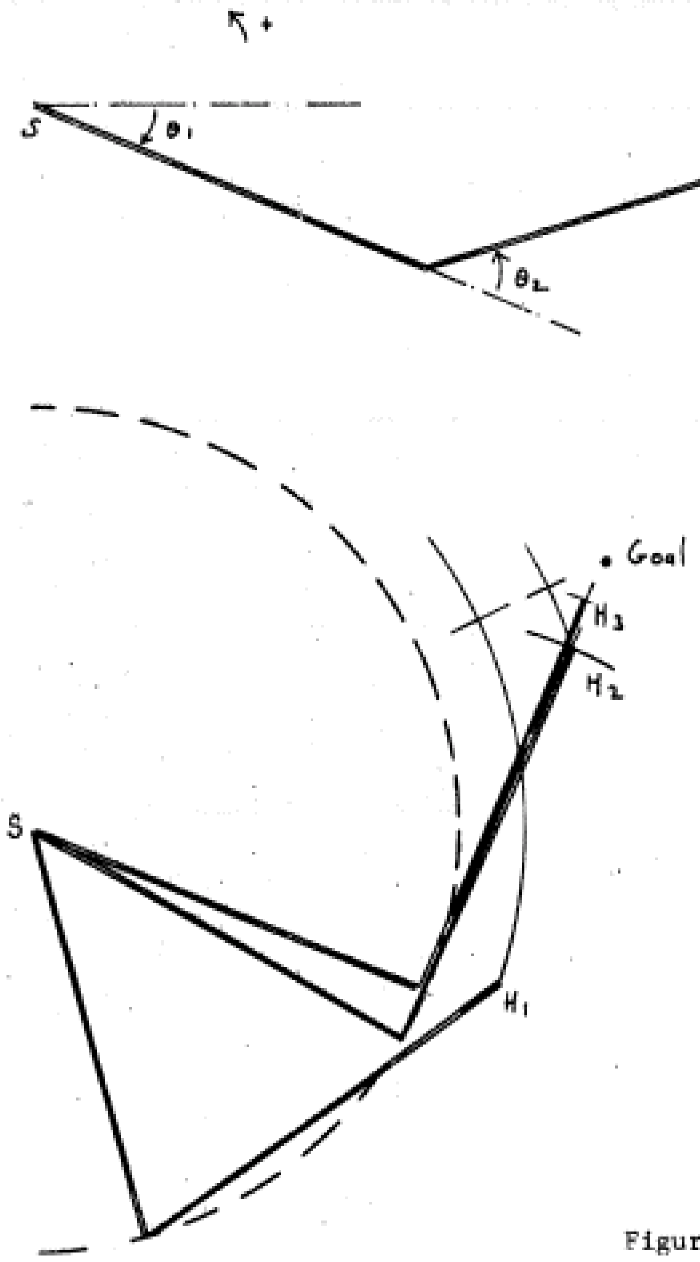


Figure 1. Two-bar plan linkage

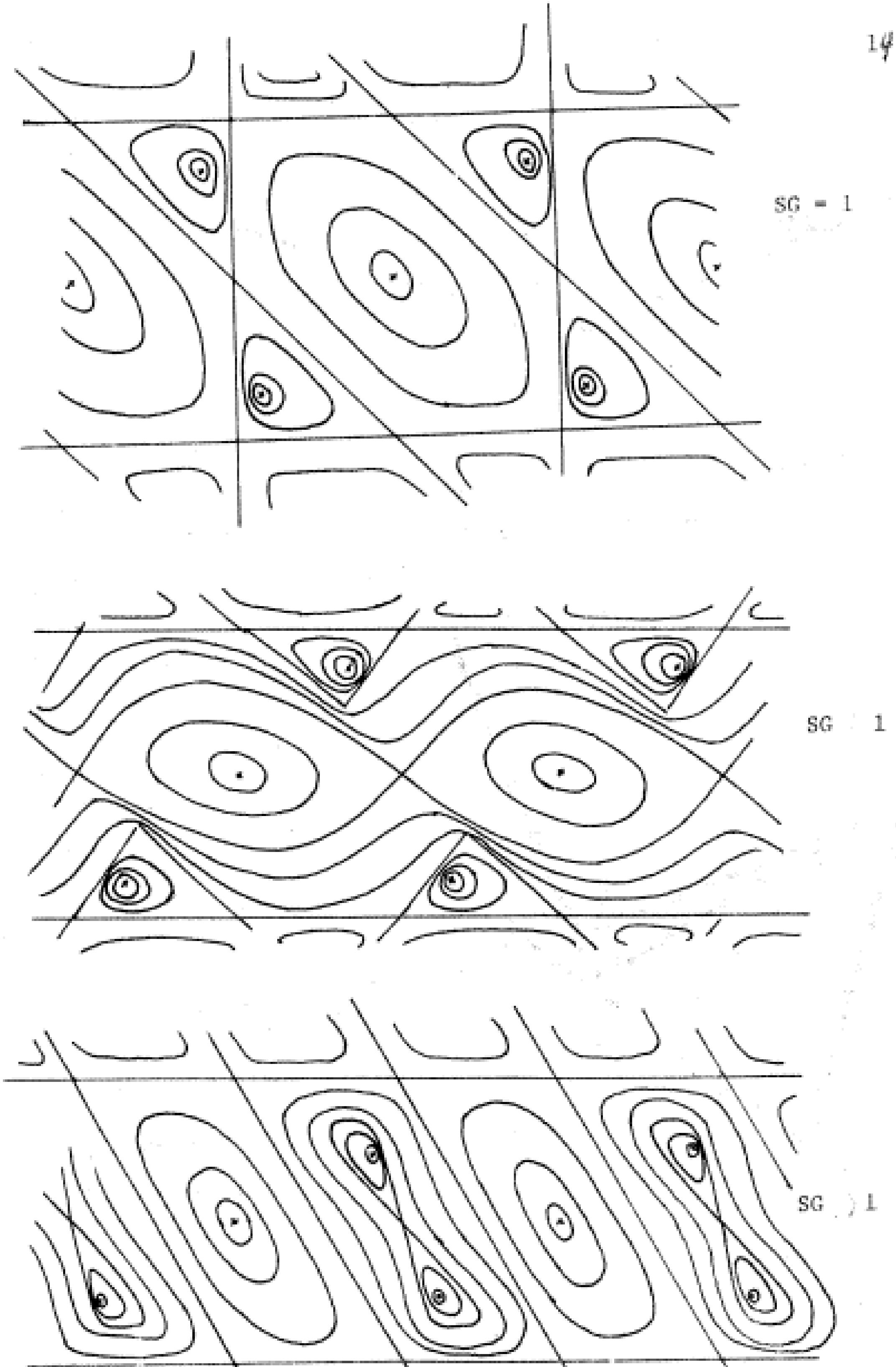


Figure 2.Curves of equal distance in the parameter space

Let us first assume no restriction on the real values which can be taken by θ_1 and θ_2. To find a solution we can apply any hill-climbing algorithm. One of the simplest is the following:

1) move the first lever, stop when HG is minimum

2) move the second lever, stop when HG is minimum

3) if the value of F=HG is less than E, the goal has been reached, the couple of values (θ_1, θ_2) is part of the solution - otherwise go to 1.

We may have a problem with any saddle point $(\theta_0+2u\pi, 0)$, but it is not difficult to get rid of it.

Anyway, the algorithm we may use is not important as soon as it is of the steepest-descent type. It is easy to convince oneself, looking either at the (θ_1, θ_2) plane or at the original (x,y) plane, that starting from any position the hand can reach any goal at a distance equal to or less than 21 from the shoulder.

Let's now assume a finite area for the domain of F in the (θ_1, θ_2) plane. The important condition there is that it is finite and not that it could be inside a square of area less than $4\pi^2$.

For the sake of simplicity the domain is limited by a rectangle. It is possible to follow on one of the two figures, how the hand can get trapped in a position (θ_1, θ_2) corresponding to a point near a corner of the domain, and on the limit.

To realize this one can think of the following fact. One can reach with only one hand nearly any point in one's own back. But this has to be done eithe from above or from below. There is a region that can be reached from loth positions, but starting from a "below" position it is impossible to go upward, outside that region in the continuous move and starting from an "above" position it is impossible to go outside, downward.

Does there exist an algorithm to get out of such situations with a reasonable amount of computation?

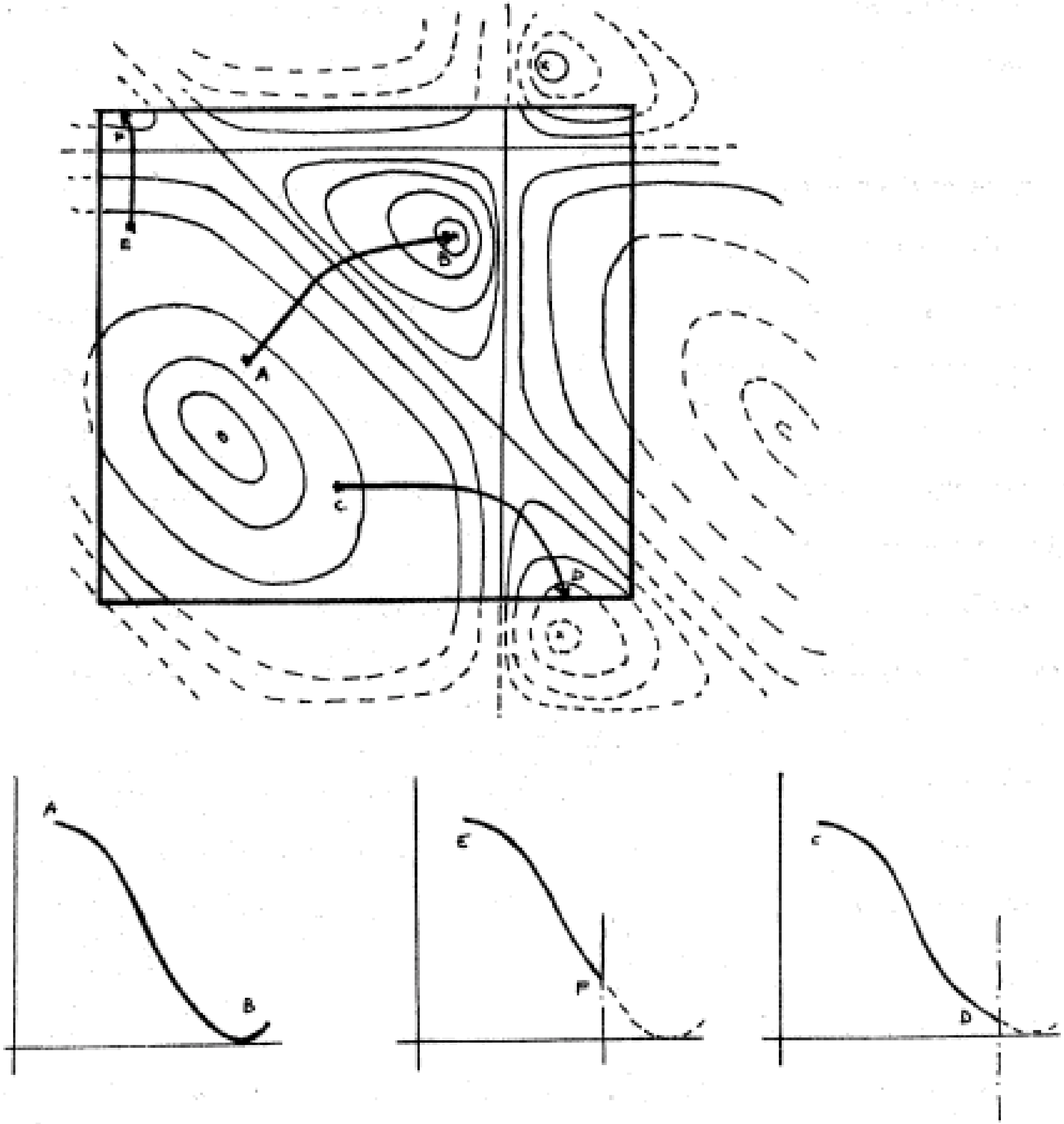


Figure 3. Effect of the physical limitations

There is no general answer to that question.

An idea is to go back inside the domain with a sufficiently large step and in such a direction as to get out of the dangerous region. The method may be successful if it is easy to predict size and shape of such a region. But hill-climbing is a local method precisely used because one cannot give a global picture of the variety $F(x_i, x_2, u_, xu)$ in the $Fx(x_1 xX_2 \ldots xXu$ space, and likely any region of large size with respect to the step of the search.

Beeler uses a very ingenious way to compute the step of reentry in the domain, generally along a direction perpendicular to the limit. But this is not an absolute parry.

In two-dimensions there seems to be one, using the symmetry of the problem (which suggests that it is a general solution for all two-dimensional problems, instead of just this particular problem).

1) the hand is trapped in position (θ_1, θ_2), go to 2.

2) start the search from $(\theta_0 - \theta_1, \theta_2)$.
   if no success, go to 3.

3) start from $(\theta_1, -\theta_2)$

4) start from $(\theta_0 - \theta_1, -\theta_2)$
   this must be a success.

   Remark: the four points must be within the domain. So there is eventually a simple transformation (e.g. homothety of center (θ_o, θ)) to be performed initially on (θ_1, θ_2) See figure 12.

My claim, easily checked, is that if there is a solution it can be found using the first algorithm and that simply parry. Unfortunately this is inapplicable when the number of dimensions of the parameters space increases. The amount of computation grows exponentially. For example, with the new arm having eight degrees of freedom, we would have to start the search from 2^8points! Does there exist a powerful heuristic to choose among those points?

Another idea is to go out of the domain, with the hope of coming back later with acceptable values of the parameters, i.e. for which the search converges. It may or may not work depending on the particular problem. With the new arm it does not. If the two-dimensional model is generalizable it is easy to see why. Once out of the domain (which is convex) the point (θ_1,θ_2) will not come back but go to the next depression and stay there.

If we try to infer the "inner" solution from the "outer" one, we are led to a search similar to the foregoing parry, which is inapplicable in an eight-dimensional space.

There is at the present time no other idea to improve or assist the hill-climbing method.

I have shown that the physical limitations imposed on the movement of the arm are sufficient to make the use of hill-climbing techniques inefficient, even in the simple non-redundant two-dimensional case. To reduce the width of the domain of each parameter by P is to shrink the domain in the solution space by $(P)^N$, if there are N parameters. This may be bad, if N is large enough.

There may be other causes of failure, but certainly not to be found out easily.

It would help with angles as parameters to make them vary with an amplitude of 2π. Any solution would be made (2π) inside the domain, and the amount of computation to bring it back effectively would only grow linearly with the number of parameters. But unfortunately such a case bears no reality, we may only hope to approach it as close as the materiality of the arm permits it.

The manipulation of computer models of molecular structures [6,7] - (The discussion refers to a series of programs written under the direction of Professor C. Levinthal.)

The reactions between the small molecules of the living cell are specifically catalyzed by protein molecules, the enzymes. The framework of a protein is a chain of peptide groups to which are attached amino acid groups in a characteristic sequence.

To describe such a molecule the number of parameters can be reduced from several thousands to several hundreds on the assumption of the rigidity, due to chemical constraints, of certain parts. The plane configuration of the peptide bond allows the representation of the path of the central chain by the sequence of the rotation angles, two of which define the relation of two successive peptide groups. The specifications on the side chains are stored for each of the varieties of amino acid.

Such a description is suitable for study on a computer ofthe deformations of the molecule toward a minimal configuration.

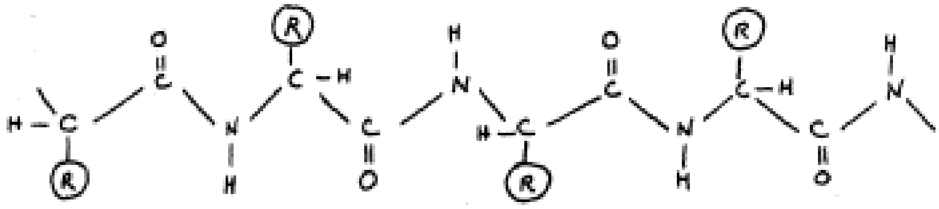
The function one wishes to minimize may be the total energy of the configuration, or more simply the distance between two points of the molecule.

The search is done through local methods, either hill-climbing or Euler-Lagrange equations with a close formula for the derivatives.

From all the atomic interactions a set of constraints is chosen, among them the van der Waal forces which are a way of taking into account the finite volume occupied by an atom. Checking for these constraints can be bypassed. The user may also introduce his own set of interactions and eventually reduce the number of degrees of freedom of the model by freezing any part of it.

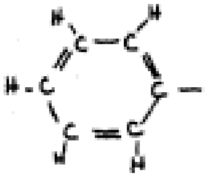
Two kinds of trouble may appear during the search. Small rotations can produce a large variation of the total energy, so angles have to vary by very small steps. But in other regions of the parameter space, these steps may be too small to produce any interesting change. It takes a large amount of computations to reach a useful configuration. At the present time the program does...

Protein Chain

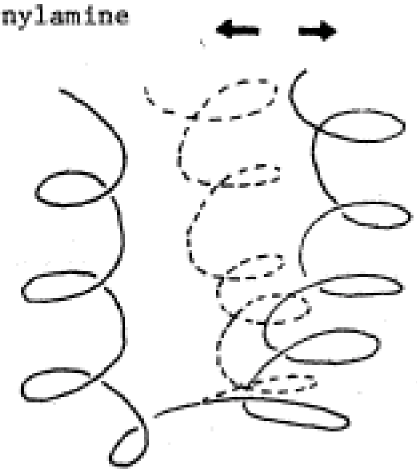


Ⓡ is the side group from the contributi aminoacid

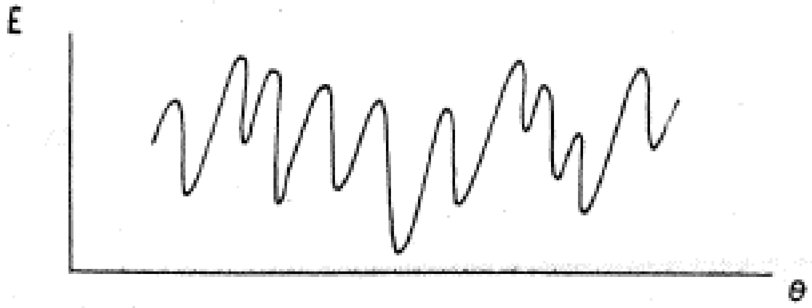
ex: H - glycine



phenylamine



The effect of the CLOSE routine



free energy vs an angle-parameter

Figure 4.

time the program does not have the ability to recognize such things happening as the user can generally do.

This defect reminds one of the mesa phenomenon and requires the same type of correction: larger steps in the search, and some sophisticated heuristics to transcend the hill-climbing.

Its cause is undoubtedly in the large number of parameters. If introducing a new arm with a comparable amount of degrees of freedom appears seductive, because, as we will see, we no longer need be concerned about the physical limitations, we must think about introducing new defects.

As expected, the search can get trapped in a local minimum, which is not absolute. Such minimum does not exist for the real molecule. The thermal vibration gives it enough energy to reach the next absolute minimum (the binding force is small, unless the atoms are very close). The program does not simulate this vibration and the intervention of user which makes small alterations of structures is required.

So far nothing new of our concern (for an arm) seems to have happened. But one has to look a little bit closer. Any function F used in the search depends on a very large number of parameters. The total energy is sensible to any small variation of any of these parameters. So the closest relative minimum of such a function may be extremely close indeed to the absolute minimum, the goal of the search. Said differently, the size of the domain where F has only one extremum is extremely small, compared to a domain where any angle parameter could have a variation of ten or twenty degrees. There may be simpler functions than the total energy, simpler in a sense; they do not depend equally on all the parameters in any region of the parameter space. One would expect such functions behave more "smoothly" in larger domains.

As a matter of fact there may exist for any problem a function, continuous and with as many continuous derivatives as required, that will have its minima at the solution points alone. An interesting theorem would be one telling (at least for some typical problems) whether the search for such a function would be jusitifed by the economy of computation on the whole hill-climbing process.

In the molecule-modeling problem a simpler function is for example the distance between the two extremities of the chain, as used in the "close" routine.

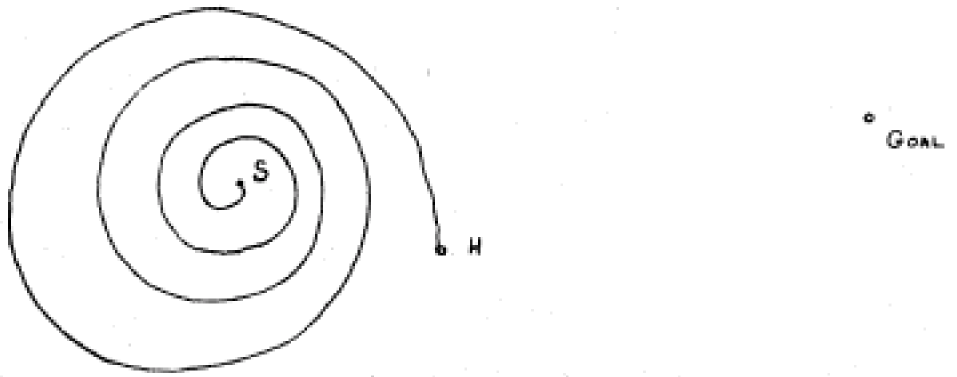
To consider such a function does not suppress the local minima. Let us assume for example the chain is a portion of planar spiral wound up several times about the origin. To reach a far enough point in the same plane, it has to unwind and along the process the euclidian distance between the extremity of the chain and the goal goes through several local minima. But the distance between these minima, in the domain is rather important ($\sim\frac{2\pi}{\sqrt{n}}$).

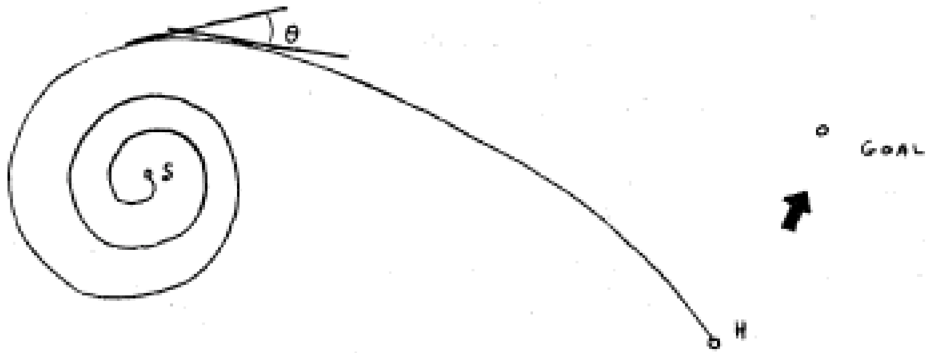
So there is a relatively large region where the only minimum of the function is the one to which the search is directed.

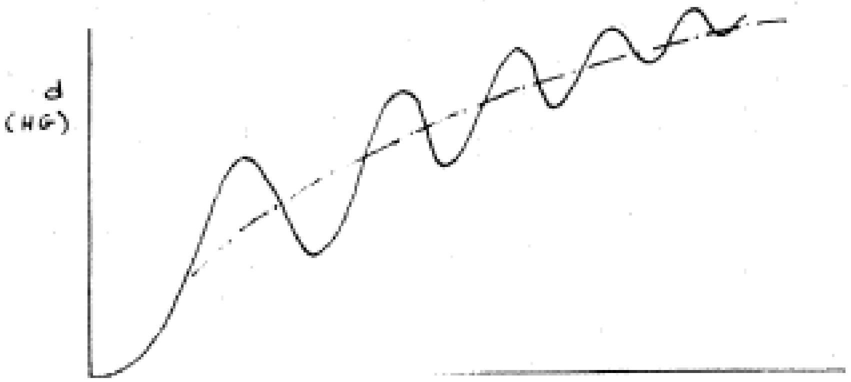
That region is where the spiral (or the helix) has unwound enough.

This is not the only region. The chain can be frozen in an helix, about a curve which is also an helix, and so on..., the process is iterative. It has been effectively used and with success, the CLOSE routine for example, effectively closes an open chain. The control on the FREEZE procedure is at the present time by the user, but it does not seem so sophisticate that it could not be implemented in the program.

The use of a function F as simple as it may be does not free the program of the physical limitations. These limitations which are not directly on the angles but which result from the fact that any physical entity occupies a volume in space, that it cannot share with any other entity. The consideration of these limitations can be simply skipped in perfect legitimacy, as we have seen. In conclusion, for a simple function we have a simple heuristic to pass the relative minima:







If the minimum is not a solution

1) unwind the helix

2) switch off the routine on the physical limits

do 1 and 2 until the function decreases.

We have now a model for an arm, with a large number of degrees of freedom. To consider movements restricted to those of helix-like configurations may appear as a very inefficient use of such an arm. But this inefficiency may be the price we must pay to have a totally reliable device in a range we know well.

Unfortunately, such a model cannot be realized, at the present time; the lack of a joint of reasonable size which would allow the angles between two limiting segments to vary over a large domain (with width equal to or greater than 2π).

<u>The optimal control theory approach</u> - Both foregoing programs may use, according to the relative number of parameters and constraints, Euler-Lagrang equations, taken from the control theory techniques.

Typical approach along an exclusive optimization criterion have been made: on a manipulator with several degrees of freedom by Mergler and Hammond [8], on the "target approach in biological systems" by Tomović and Petrović [16]. (What follows is a critique of this particular article and not of the whole work by Tomović.)

The choice of an optimization criterion directs the search toward a solution which is a unique set of values of the parameters. The computation reduces to the integration of a set of differential equations.

To use differentiation one needs a mathematical description of the constraints and of the optimzation criterium. An exact description may be difficult when the system has a large number of degrees, so linear approximations are used. This adds to the method a problem of step-by-step search.

Moving an arm is not stearing a boat. Unfortunately, Tomovic keeps the confusion in neglecting the constraints in such a problem, thus skipping one step of the solution, optimizing the search before knowing whether the search will go anywhere.

The optimization reduces generally to such a panacea like minimizing

$$\int x_i \left(\frac{dxi}{dt} \right) dt \qquad \text{or} \int \left(r x_i^2 + \left(\frac{dxi}{dt} \right)^2 \right) dt$$

where r is a real number, x_i are the paramters of the system and t paramter for the optimal curve from the initial position to the goal. In Mergler and Hammond study, such a criterium results in movements which the author finds unsatisfactory. Tomović more interested in the timing of the moves, presents the uninteresting example of a movement along a straight line in a free space, with a two-bar linkage having three degrees of freedom.

Such models do not pretend to rely on a description of what happens in the living organisms. Since their results are unsatisfactory, the best models are still the living organisms. But the description by the naturalists of the movements and their subjacent mechanisms are extremely elliptic and of no use to formulate any serious hypothesis from their explanations.

All one can say is that the use of optimization principle does not seem adequate to describe what we see in nature. It is difficult to pretend that any action is optimized from its start to its end, even the simplest, like reaching a pen on a table. In a first step using kinesthesis, the arm moves its extremity in the neighborhood of the pen. In a second step, vision, tact and kinesthesis are used (How?) for guiding the hand in a closer position to the right object. In the first step the arm knows from where it starts but does not know where it will end. If there has been optimization it is certainly not the simple way one usually thinks.

Is a style in racing or dancing compatible with optimization?

I suspect the control theory approach not to be the right one at the present time. It is the attitude in which one does not look into things. Problems are not so complicated with the manipulators that we cannot do that. Also it does not seem useful when the main problems are still at the gear-and-bar level. A multidegree arm is capable of giving several solutions for one task; it is still interesting to know what they are, at least at the primitive stage that we still are in.

Chapter III

Programming the new arm - a geometrical approach

Analytical geometry in two or three dimensions may be simple. The modular conception of the arm suggests looking for an iterative process to solve our problem. Each step would be a problem in a space with fewer dimensions than the solution space, so' ,ible with analytical geometrical means.

Such an idealized process is presented on an idealized arm.

An attempt is made to adapt the solution to the real arm; this leads to a slightly different method which may give more insight on the motion of living organisms (though it is not the goal).

An arm with ball-and-socket joints - Using simple analytical geometry, it is possible, given any point within the range of the arm, to find a configuration of the linking segments which makes the hand-extremity coincide with that point.

A metho.i is given to generate any solution from a particular solution.

The theory is illustrated on an arm with four segments. Generalization is possible as far as the segments are not too many. Movements of such an arm in a three-dimensional maze are not considered.

The arm - The relative position of two segments, say Su and Su-1, can be found through the value of two angles, one is of rotation about Su-1, which can take any value between 0 and 2π Rd (eventually any real value), the other is cf rotation about an axis perpendicular to Su and Su-1, which goes through t.e center of the joint between Su and Su-1. This angle can vary in a smaller interval, e.g. from $\pi/4$ to $\pi/4$ Rd.

There is no distinction in the following description between a segment and its axis, between a joint and its center.

The locus of the extremity of Su, opposed to the joint between Su and Su-1, in a space fixed with respect to Su-1, is the portion of a sphere, centered at that joint, with radius the length of Su, inside half a cone, which summit is the joint, which axis has the same direction as Su-1, and which angle is $\pi/2$.

We wish to describe the locus of the extremity of Su in a system attached to Su-p. All segments have the same length 1. P=2. The locus is the volume generated by the foregoing portion of sphere moving about Su-2, with two degrees of freedom, i.e. the two angles which define the relative position between Su-1 and Su-2.

Due to the symmetry of revolution of such a locus, it is easy to find out what it is:

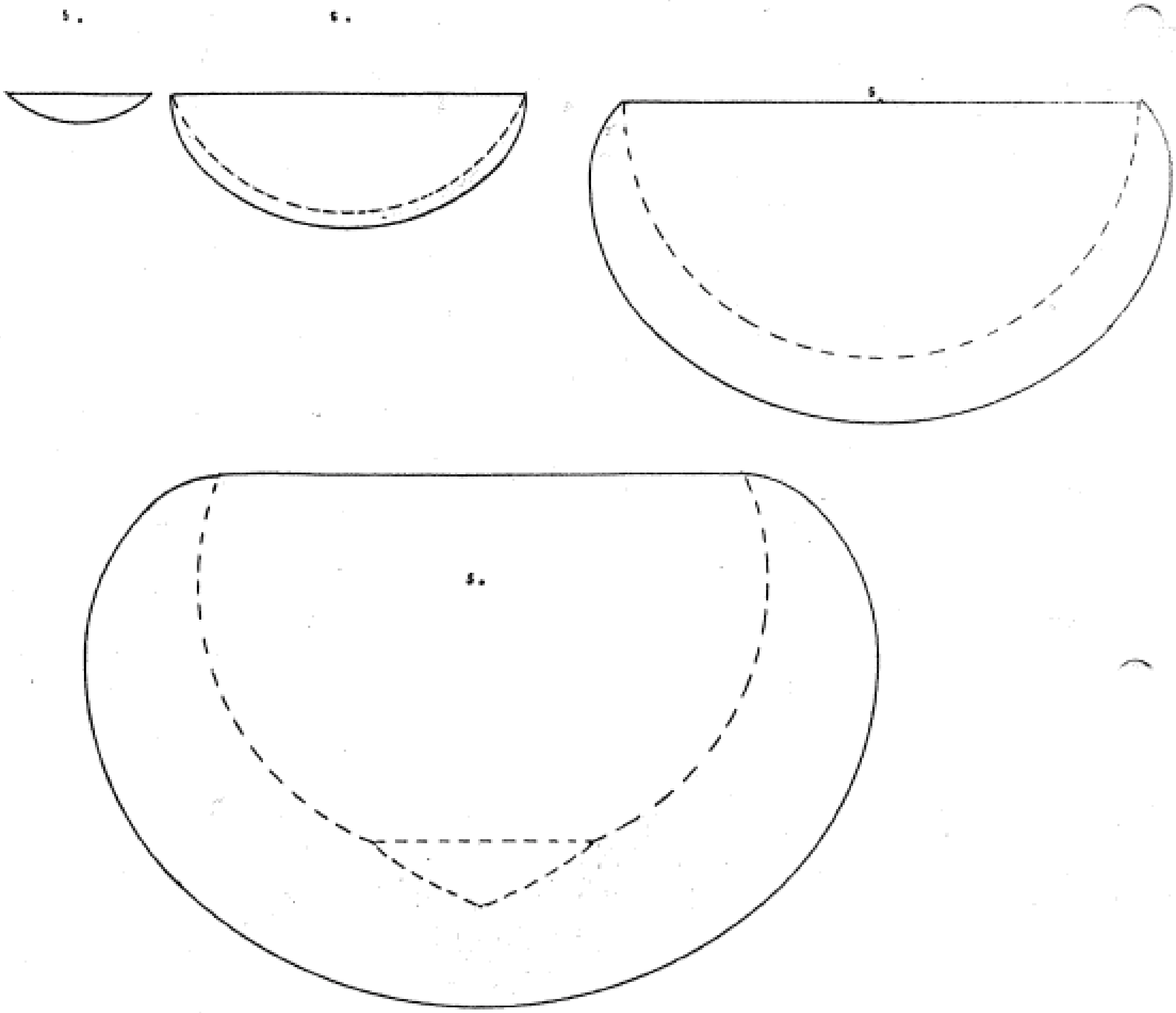


Figure 6.L_s Volumes

it has a mushroom-like shape limited by two curved faces. The convex face can be defined by its section in a plane which contains its axis:

a quarter of a circle, with radius 2l, centered on the Su-1, Su-2 joint, completed at each extremity by an eighth of a circle, tangent, with radius l.

The concave face is a portion of a sphere centered on the Su-1, Su-2 joint.

p>2. The locus Lp can be generated by the rotation about Su-p of a planar surface, which limits are that axis and the two following curves:

- part of a spiral made of p tangent eights of circles, which radii are in arithemetic progression from pl to l. The circle of radius pl is centered on the joint Su-p-1, Su-p
- part of a circle centered on the same joint, which radius is given by the point of the spiral, closest to the joint.

Eventually there may be a depression in the center of the inner face of the locus. This depression appears for p=3 and disappears for p=6.

<u>Existence of a solution</u> - The first use we can make of Lp is to find whether a given point in the three dimensional space is within the range of the arm with p segments. A point H is reachable if and only if it belongs to the section of Lp by the plane defined par Su-p and M.

One may define the limits of that section by circles and by straight lines separating them, circles and straight lines defined by their equations in a coordinate system attached to the plan, with origin at the joint Sh-p+1Sh•p:

with $s_j = \sum_{i=0}^{j} (p-i)\ell\cos i\pi/k$ $\quad \sigma_j = \sum_{i=0}^{j} (p-i)\ell\sin i\pi/k$

$$C_j \equiv (x - s_j)^2 + (y - \sigma_j)^2 - (p-j+1)^2\ell^2 = 0$$

$$P_j \equiv (y - \sigma_j)(\sigma_j - \sigma_{j-1}) - (x - s_j)(s_j - s_{j-1}) = 0$$

$$0 \leq j \leq p-1$$

$$c \equiv x^2 + y^2 - r^2_p = 0$$

$$c' \equiv (x-a)^2 + (y-b)^2 - r^2_{p-1} = 0$$

$$p \equiv x - x_p = 0$$

C_j, P_j define the outer face ,c ,c',p the inner one.

An algorithm to find the existence of a solution is:

1) Compute x and y, the coordinates of the point in a coordinate system which origin is the joint Su-p+1Su-p, which x axis is Su-p, which y axis is in the plane defined by Su-p and the point.

2) Evaluate some (a) of the Cj(x,y), Pj(x,y), c(x,y), eventually (b) c'(x,y) and p(x). This defines a signs pattern.

3) There is a solution if that signs pattern is the one (a) of a point belonging to the section.

(a) depending on p

(b) 3 $\leqslant$ p $\leqslant$ 6

To be able to use this algorithm one must know the signs pattern. One has primarily to figure out directly (making a drawing is the most direct manner) how the section looks like in order to eliminate circles and lines in overlapping regions. This is simple if p is not too large (several units).

Algorithm for a solution - Su-p is on the axis of Lp, with its remoter extremity (toward the shoulder) at the distance pl from the center of the outer face of Lp. Then from Lp one can find simply where Su-p is. A method to find solution is then, with a p-joints arm:

- Once the point is known to be in Lp
- find one Lp-1 containing the point
- then Lp-2, etc.
- then Li.

The configuration for the arm is Su-p, Su-p+1, in, Su where Su-j corresponds to the Lj found. It is a solution. One may ask a question about finding Lp-1 after Lp. In principle it ~~is~~ is possible since Lp is generated by Lp-1 moving with Su-p+2, with respect to a system attached to Su-p+1. But it is not easy practically since Lp-1 is a volume moving in space with two continous degrees of freedom.

A search, with discrete steps, for the values of the two parameters, using the analytical expressions of the contours of Lp-1 in order to find

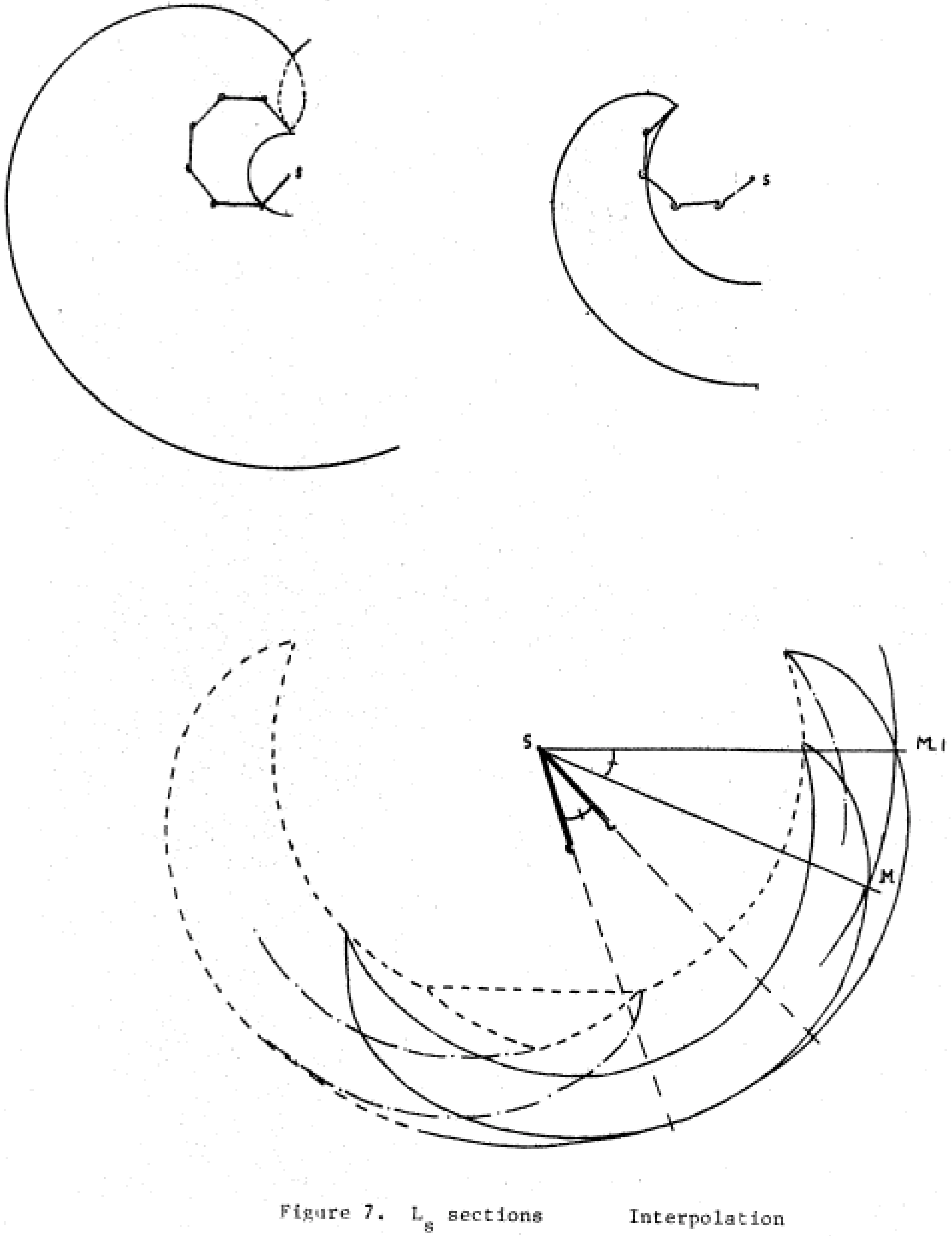


Figure 7. L_s sections   Interpolation

the position Lp-1, for which the goal-point gives the right signs pattern, may be time consuming.

If we just want to go one step further than with the methods seen in Chapter I, we only need to find one solution for any goal point. We can restrict ourselves to planar solutions, even to a particular type of solution, among those whose plane contains Su-p.

What has been said about Lj in the three-dimensional space can be said about the section of Lj in that plane. The description is one parameter-less simpler. One could, in order to find a solution, use the analytical expression of the curves limiting the section of Lj.

But there is a faster method.

- Lj has now the meaning: section of Lj in the plane of the solution - Lj-1 is given, inside Lj, by the position of Sh-j, i.e. by an angle θ about the joint Sh- +1 Su-j+2.In a circular coordinate system, whose center is that joint, the goal-point M has α as angle coordinate. All one needs is to relate θ and α, in such a way that Lj-1 can contain the goal-point. For a value α there are no unique solutions θ, but one can choose one of them in the following way.

The origin axis contains Su-j+1, by definition. A circle centered on Su-j+1Su-j+2 and containing M, crosses Lj contours in two points whose angle coordinates are $+\beta$ and $-\beta$. Then θ is chosen as:

$$\alpha - \left(\frac{\pi}{4} - \frac{\beta}{2} \right) \quad \text{if} \quad \alpha \geqslant 0$$

$$\alpha - \left(\frac{\beta}{2} - \frac{\pi}{4} \right) \quad \text{if} \quad \alpha < 0$$

This means M is taken on the positive side or on the negative side of Lj-1, outer contour.

β is the coordinate of the intersection of two circles, which problem reduces to the resolution of a second degree equation.

Thus, to find a solution we first check the existence, then iteratively apply the last method with j taking all the integers' values from p-1 to 1, included.

The conditions are for the success of the global method:

- the existence of a symmetry which makes the problem reducible to a problem in the plane
- the simplicity of the contours of the loci Lj.

The two conditions are a restriction on the nature of the joints. They should be taken into account as much as possible for a later design of the next arm. The last conditions could be a restriction on the number of joints. But for a large number, anyway, we should start thinking of other methods.

<u>To find more solutions</u> - let us call the joints starting from the shoulder J_1, J_2,...J_p. Given a solution we can find more solutions, by rotating the segments of the arm from S_1 to Su, about the SJ_k axis and the segment S_k to Sp, about the MJp-k axis. Given one solution this makes

2x(p-1) + 1 = 2p-3 infinities of solutions available.

For an arm with p joints, we have 2p parameters, the indetermination about theose parameters for a point M in a three dimensional space is 2p-3.

Thus as far as that type of reasoning is valid one has here a way to find <u>any</u> solution.

Movements in a maze - The next step is to move the arm in the presence of objects some of which should not be knocked down. At the present time the only reference on such a problem is a doctoral thesis by Whitney [17] who limits himself to the rather uninteresting case of two dimensions. To my opinion the three-dimensional case is a highly sophisticated task, combining the use of kinesthesis, tactile and visual senses. Achieved, it will be a decisive step for the ROBOT. But the preliminaries go through the programming of a bar-and-gears device.

The foregoing method has not been simulated on a computer, since it appeared more urgent to program the real arm. But it contains the initial idea for the program actually running, to attack the problem with the simplest geometrical means. It is important to remark that the method subdivides a problem in a 2p-dimensional space into p problems in a 2 dimensional space.

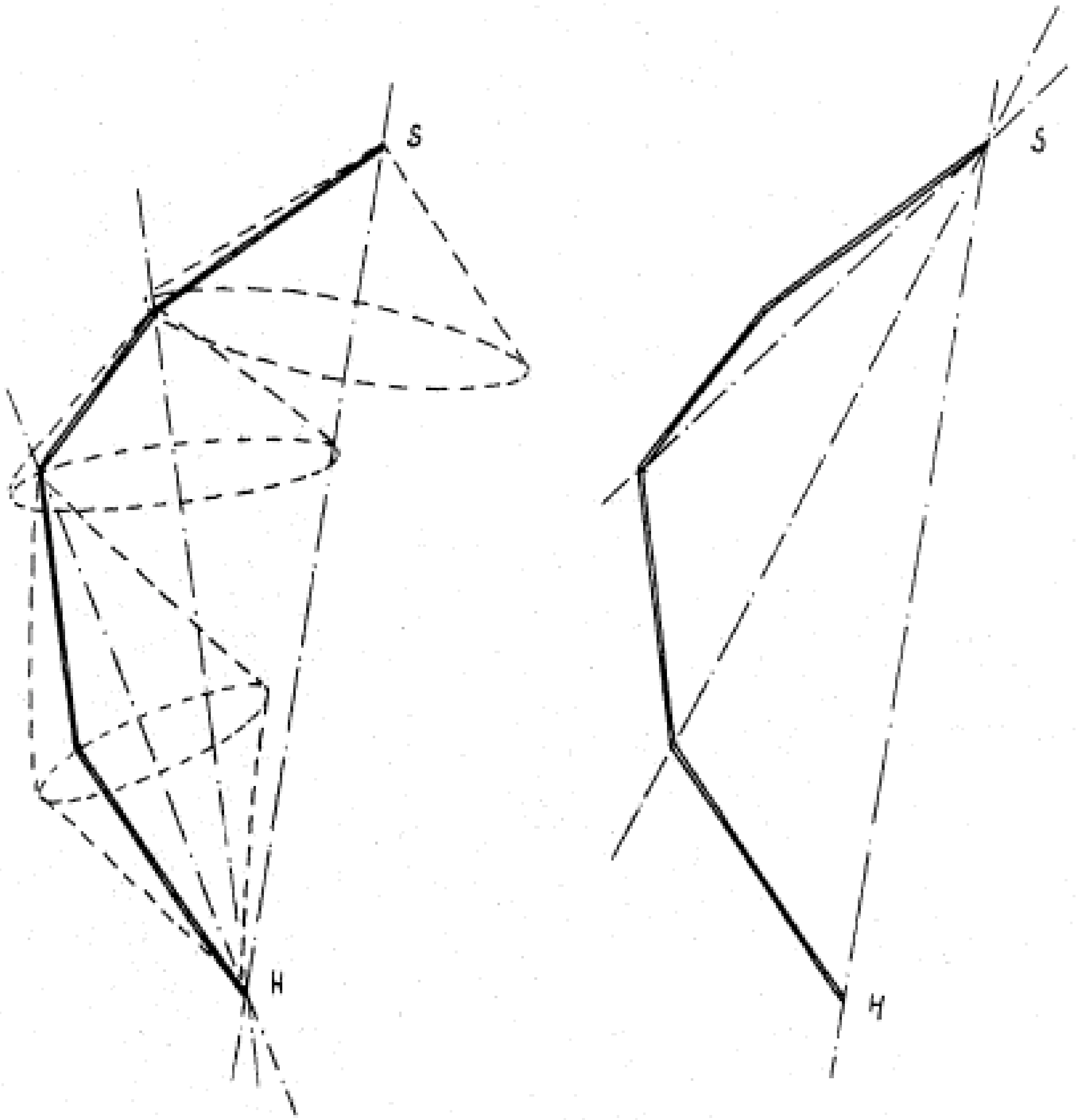


Figure 8. More solutions

MOVING THE NEW ARM

Description of the arm MA-3 - The main parts are a shoulder and an arm. This device is made of four long tetrahedral segments or "bones", with a double symmetry about the longest median in the direction of the corresponding perpendicular vertices. These four bones are related to each other through three joints, and through what may be considered as two half-joints to the shoulder and to the hand in a symmetric design. The joints are identical, each of them has two degrees of freedom about two perpendicular axes, and is powered by two cylinders moved by oil pressure, the displacements of which are measured either by linear or by rotational potentiometers.

The arm has, then, from the shoulder to the hand, eight degrees of freedom. [11ø]

Search for a method - With the ball-and-socket joints arm the search for a configuration in 2p dimensions is found equivalent to a sequence of p subproblems in two dimensions.

The segments composing the arm are placed the closest to the shoulder first, by the consideration of geometrical figures (in two or three dimensions) of decreasing complexity. Despite the variations of the complexity the method used at each step is the same. This is due to the symmetry of those figures and to the relative simplicity to generate this description, i.e. in fact due to the nature of the joints.

Such qualities of symmetry and simplicity are not found in the new arm.

A priori the half-joints on both extremities disturb the uniformity of the iteration process.

In the ball-and-socket joints arm one of the angles could vary from 0 to 2π Rd, which creates that useful symmetry of revolution about each segment. Unfortunately here better angles can vary into an amplitude only slightly greater than π/2 Rd.

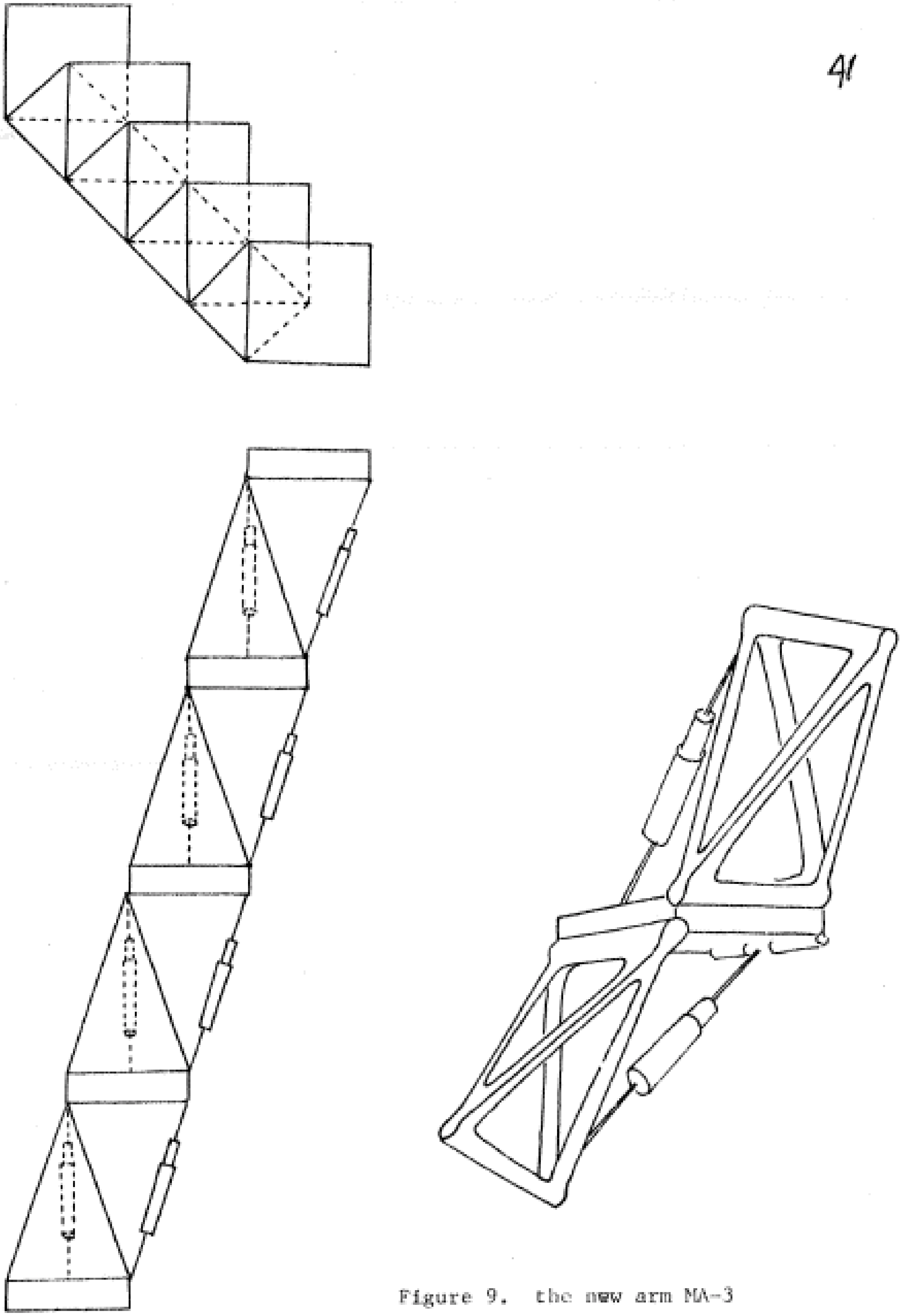


Figure 9. the new arm MA-3

This creates odd shapes for L_3, L_4, etc. - apparently difficult to work with. It could help instead of symmetrical tetrahedra, having asymmetrical ones with four right angles. But I do not think the improvement is worth the modification. It is more important to try to extend the amplitude of variation of angles-parameters.

The last disadvantage and the major one at the present time is the fact that the joints are without center, their two rotation axes do not meet and by a non-negligable distance. This makes any nice portion of sphere one could have expected become a portion of torus.

This fact has been neglected in the first version of the program then taken into account with error-correction formulas in the second version.

Thus the method used for the ball-and-socket joints arm is not applicable from the shoulder to the hand.

Let us assume now the joints have a center, called C_0 C_1 C_2 C_3 C_4 from the shoulder to the hand.

The position of C_2 in a system attached to the shoulder is given by the value of three parameters. Symmetrically in a system attached to the hand, the position of C_2 is also given by three parameters.

This suggests a way to solve the following problems:
find the configuration of the arm which gives a known position of the hand, defined by a point in the three dimensional space of the arm and the orientation of a coordinate system attached to the hand, in the same space.

In any configuration of the arm, C_2 is in the common part of L_{2s}, which is L_2 taken with the shoulder system as initial system, and of L_{2H}, an L_2 with the hand system as initial system. L_{2s} and L_{2H} have identical shapes. But any common point to L_{2s} does not correspond to a configuration since the joint between the second and the third segment has physical limitations.

Thus the method is:

1) find the intersection of L_{2s} and L_{2H}, then

2) check among the points of this intersection the physical limitations, and keep those which verify them.

One has now to define L_{2s} and L_{2H} and find their intersection. Using exclusively analytical means do not seem workable. This leaves two possibilities:

1) define L_{2H}, L_{2s} by two tables of poins and take as their intersection the couples of points which distance is less than a certain threshold.

2) define L_{2H} or L_{2s} by a table, and keep the points to which by analytical means, it is possible to associate three parameters' values in the other system.

About one hundred points seem to define a good resolution for L_2. This makes the first possibility a little prohibitive in terms of amount of computation. The second has been preferred as it has revealed success.

However, there is in the choice of that method more than exclusive pragmatical reasons, and compromises to adapt the method found for the ball-and-socket joints arm.

More insight from biological processes - Naturalists and behavioral psychologists have published a large number of studies on motion. These studies are essentially descriptive and not very explicative about the control of motion. For human beings one explaines generally that the cerebellum, which performs a complex and precise continuous action of control, is analog to an "electronic computer controlling the flight of a guided missile" [18]. At the present stage of anatomy, it is difficult to say more about the cerebellum and its organization but at the present stage of computer science it is possible to say more about computers. In particular, one can say, after the unsuccessful attempts from the optimal control theory, that moving an arm requires a substantially different organization in a program than the one necessary to guide a missile.

Another possible source for help would be the work by Piaget.[14] But Piaget's concern is at a rather higher intellectual level than one which could be useful for our primitive machine. As a matter of fact, his studies on the child start too late for us when the baby has already learned a lot of tricks, where one could hope finding one of those hypothetical "primitive" motions that a mechanical arm could copy.

So very little in fact can be taken from the natural sciences. But program does not pretend to be a model on its own, it may be a link between the two fields.

Either through learning or through "wired in" organization, we know how to perform a certain number of motions. A primitive implementation of such an innate or acquired memory is the table describing L_{2H}. This table simply means that the part of the arm, between the elbow L_2 and the hand, knows how to reach certain points in space.

Very often when a task is performed only one part of the body seems to be under a nearly conscious control, when what remains of the body seems to "follow" the motions of that part. L_{2s} is just the portion of space where the arm can follow the hand.

Of course, L_{2H} and the method to find the intersection between L_{2H} and L_{2s} are very primitive features. For example, having only one L_{2H} does not take into account that many actions are transferrable to many parts of the body. The same final resulting motion of the fingers may involve the fingers alone as well as the whole arm.

But that program must be understood as a starting point. It is an attempt to apply at an early stage, at the gear-and-bar level, that common idea in Artificial Intelligence, that our activities even those which seem simple, are controlled through a hierarchy of different mechanisms.

In the organization of the program, the ordinary three-dimensional space, or the five-dimensional space of the hand orientation-and-position are considered as simply fiberspace where L_{2s} is the base and where L_{2H} is the fiber over any element of L_{2s}.

It is not worthwhile to go further now. I would like to think my program is a starting point to put into practise abstract ideas such as those by Greene on control [5].

Image - As already said, the displacements of the joints are measured either by linear or by rotational potentiometers. For convenience, the parameters of the program are angles. A configuration of the arm can be given by the values of these angles $(\theta_0,\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6,\theta_7)$ or by the corresponding values of the potentiometers.

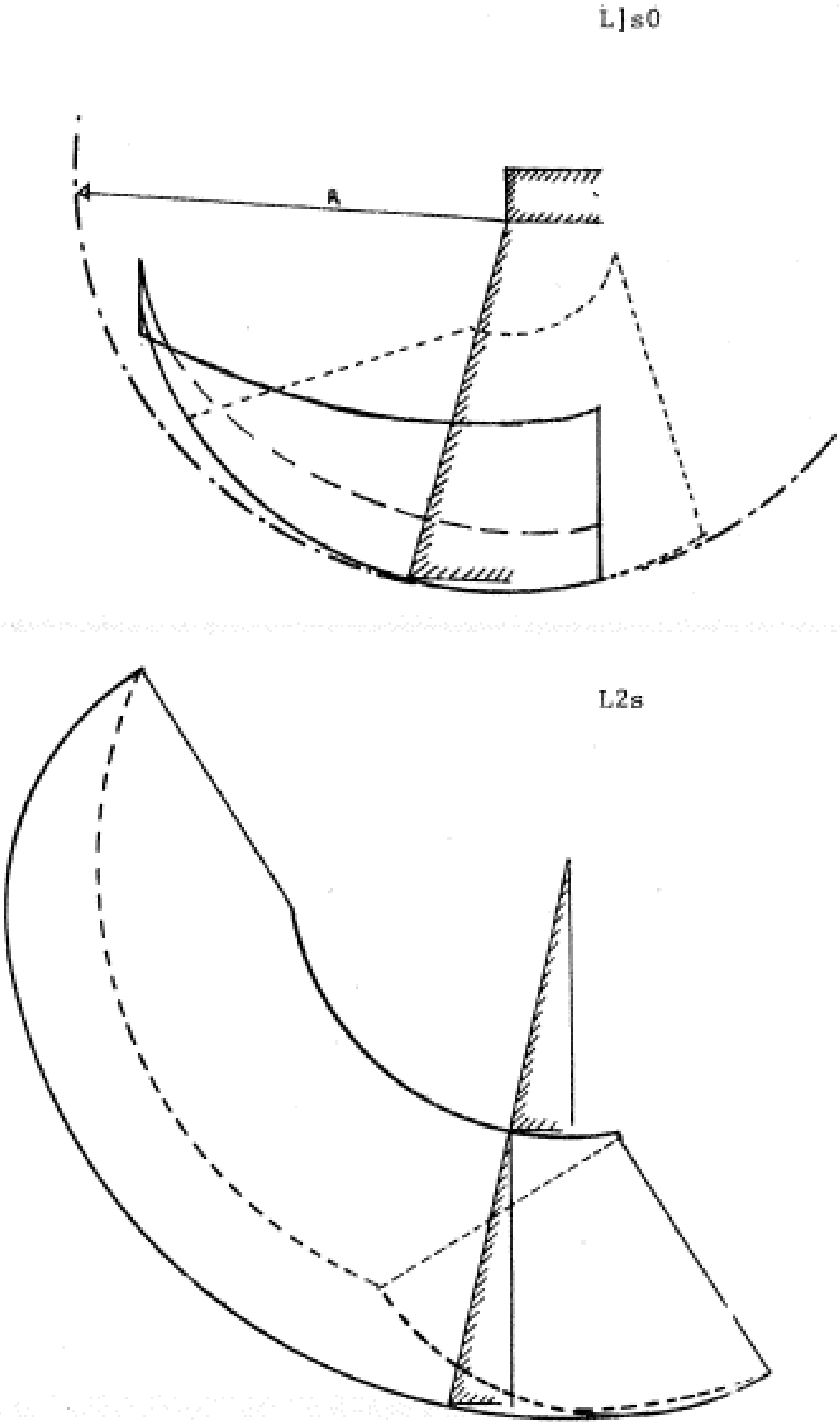
One of the main tasks of the program is to find the intersection between L_{2s} and L_{2H}. A point of L_{2H} belongs to that intersection if there exist three values of θ_0, θ_1, θ_2 within their domains such that the corresponding L_2 and the point of L_{2H} coincide.

Description of L_{2s}: L_{2s} is the volume generated by L_{1s} rotating about the axis of the shoulder. L_{1s} is the surface generated by C_2 when the two perameters of the first whole joint vary. L_{1s} is a portion of a torus. L_{1s0} corresond to the value 0 of θ.

M is the given point. M_1 is the intersection of the circle, to which M belongs and which has the same axis as the shoulder, and of L_{1s0}. The difference between the arguments of Sm and SM1 in the plane of the circle is θ_0.

θ_1 and θ_2 are the parameters of M1 on L_{1s0}; they can be found if one realizes that M1 can be obtained from the point (0,0,0) by displacements of the joint about two perpendicular axes, displacements of amplitude θ_1 and θ_2.

The only difficulty is that L_{1s0} is not a simple surface, to take the intersection with a circle in space. In a first stage, L_{1s0} was approximated by a portion of sphere, cetnered in C_1 and tangent with L_{1s0} on its lowest point (with respect to the shoulder). Then the approximation was tested using a function called IMAGE, whose three arguments, used in a first step to generate the three cartesian coordinates of M, had to be returned as intact as possible as values of



L1s0

L2s

Figure 10. 1.

θ_0, θ_1, θ_2 found through the foregoing method (TMI).

It was soon quickly found that the output values of $\theta_1$and θ_2 were independent of the input (and output) value of θ_0. So the method was tested through IMAGE (0.7 θ_1 θ_2) with 400 couples of values of θ_1 and θ_2, these angles varying between 1.4 and -0.5 Rd.

Both approximations on θ_1 and θ_2 are corrected with linear formulas, which center the approximations on the input values with a dispersion not exceeding 7,5% of the domain amplitude, and whose mean value is between 4 and 3% of that amplitude.

In a very early stage, the method was tested on an arm with centered joints. IMAGE was able to return the values of θ_0, θ_1, and θ_2 without any change on the first five significant digits. My opinion is that the present results are satisfying. Essentially because any imprecision made in an early stage of the program is corrected in the following stages.

Remark: There is a region on L_{1s0} where each point corresponds to two set: of values (θ_1, θ_2). So one may, and this is normal, obtain two sets of values (θ_0, θ_1, θ_2) for one point M. This normal ambiguity may or not disappear in the later stages.

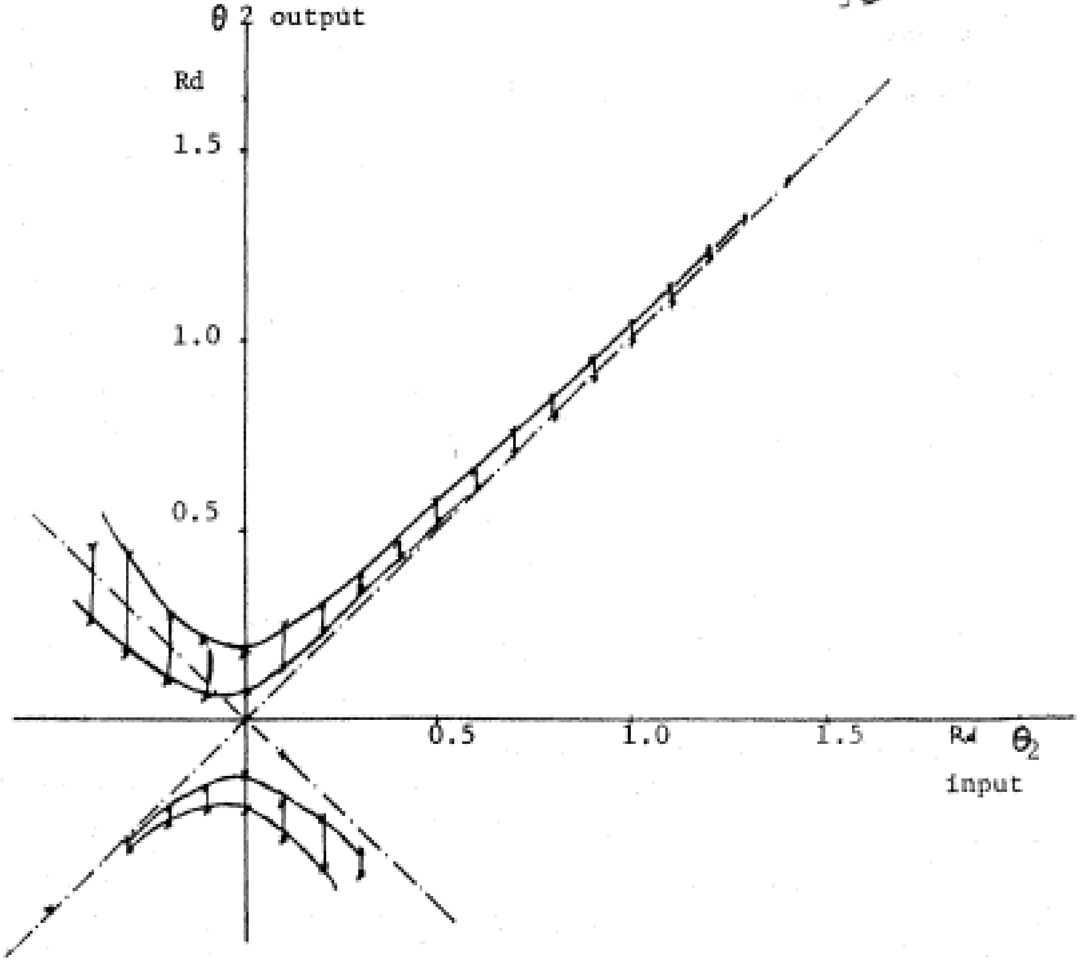
Details on IMAGE can be found in annex.

After these explanations on the weak point of the program, I will outline the main operations it performs.

<u>The program</u> - <u>step 0:</u> Once and for all the "table" describing L_{2H} was computed. It is a list of couples of triplets (xyz) (θ_5 θ_6 θ_7) which represent the cartesian coordinates in a system attached to the

hand of the position of C_2 for the value θ_5 θ_6 θ_7 of the three last angle parameters.

<u>step 1:</u> The data is the position and the orientation of the hand.



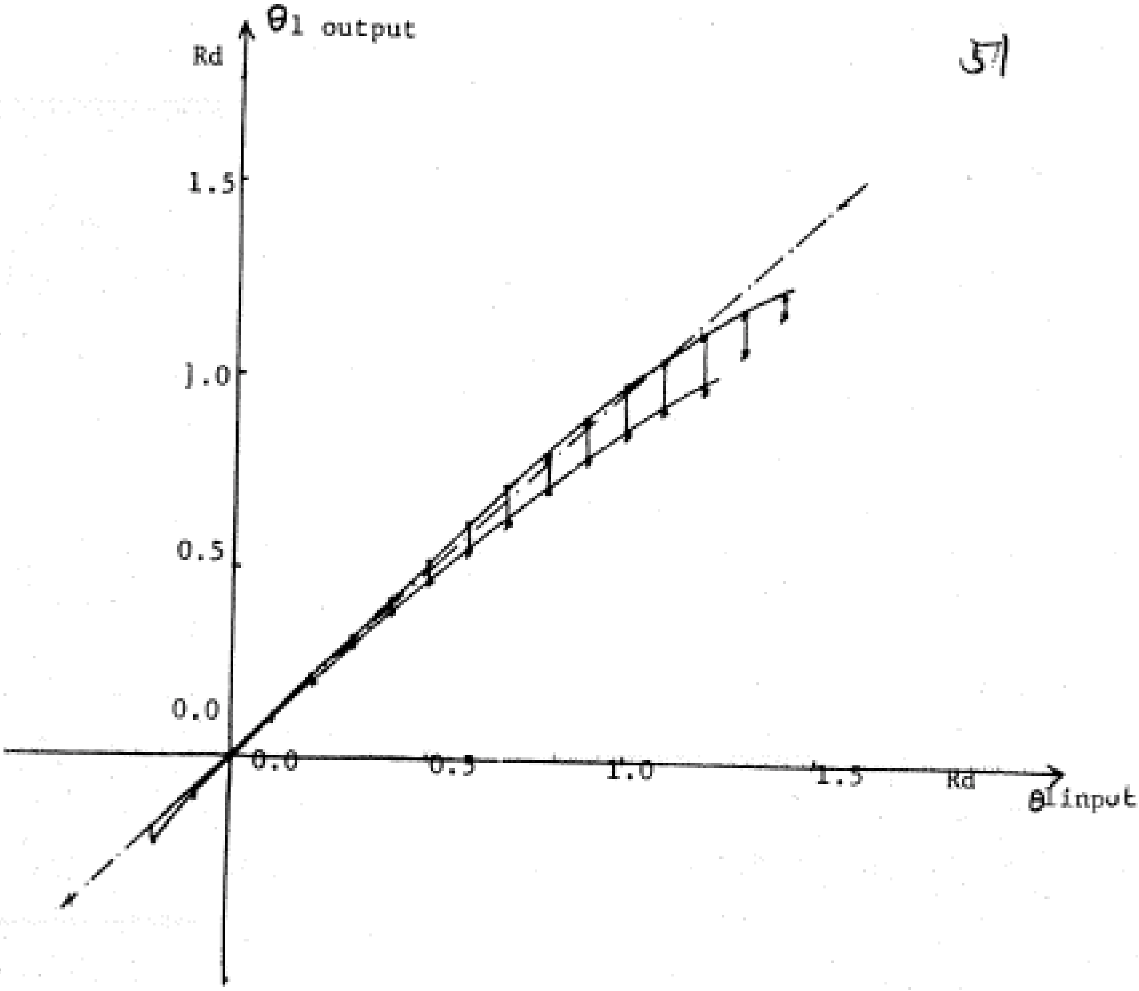
Correction applied for θ 2 output > 0

$$x = \frac{y - 0.0704}{0.949}$$

Correction applied for θ 2 out_put $\not>$ 0

$$x = \frac{y + 0.167}{0.5166}$$

Figure 11.



Correction applied for θ_1 output $>$ 0.5

$$x = \frac{12 - 0.5}{14 - 0.5}(y - 0.5) + 0.5$$

Figure 12.

After computation of the coordinates of the points of L_{2H} in a system attached to the shoulder, step 1 performs a first elimination on those points.

step 2: Applies to each point the following procedure:

1) TMI which returns either IMPOSSIBLE 1 or a list of sextuplets ($\theta_0\theta_1\theta_2$ $\theta_5\theta_6\theta_7$).

- if the result is IMPOSSIBLE 1, the program looks at the following point or skip to step 3
- otherwise it goes to 2.

2) The positive result from TMI, tells that there is a presumption of solution. The physical conditions are checked by VERIF, which result may be IMPOSSIBLE2 or (θ_0 θ_1 θ_2 θ_5 θ_6 θ_7)

- if the result is IMPOSSIBLE2, the program looks at the following point or skip to step 3
- otherwise it goes to 3.

3) θ_0, θ_1, θ_2, θ_5, θ_6, θ_7 are fixed. The arm is then equivalent to a device made of two bars. θ_3 and θ_4 are computed such that the position of the hand coin cides with the position of the goal. The result is either (θ_0 θ_1 θ_2 θ_3 θ_4 θ_5 θ_6 θ_7) or IMPOSSIBLETH. The result is put in a list L. The program looks at the following point or skip to step 3.

step 3: Considers L. If L contains a list a number (at least) it is taken as a solution. If L does not contain any number, the program considers new points in the neighborhood of these for which either IMPOSSIBLETH, or IMPOSSIBLE2, or IMPOSSIBLE1 have been obtained. [R]. If such a research fails, the program adds new points to the list which enters step 2. If this fails it returns NO SOLUTION.

Results - The result may be expressed in Rd or in pot values. Since it is difficult to find a possible goal, a priori, a small program generates position and orientation of the hand corresponding to eight values θ_0, θ_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7. The result gives the data for the main program called RECHERCHE.

One may or may not obtain the values back since θ_5, θ_6, θ_7 are not necessarily values in the table representing L_{2H}.

The program works in a domain which is more restricted than by the physical limitations ($-0.2 < \theta < 1.2$). The results being generally less satisfying when one of the angles takes a limit value.

The number of lists obtained for a solution is valuable, oscillating between one and more than ten.

The validity of a solution may be tested on the arm (there is a small program which permits one to give manually certain values to the potentiometers of the arm).

Another way is to compute position and orientation of the hand from the obtained values.

The absolute error on the position of the hand is less than 2" on any axis, which represents 5% of the range of the possible motion. The error on the orientation is more difficult to appreciate.

Despite their number the solutions generally proposed for an initial goal are not very different.

I tend to believe this is due to the physical limitations of the arm, which makes the redundancy of the four bars less impressive than it should be.

[R] remark: the last version of the program builds lists at each step and applies the iteration process to the first list which contains anything less than NIL. The first elimination process is such that if the answer is NIL there is no solution.

So the program can answer NO SOLUTION very early and not only after the iteration process. The iteration process is not satisfactory at the present time. So, I have tried to modify slightly the table and the general search, in order to avoid iteration.

THI' domain is $\theta x \theta$, $-0.25 \le \theta \le 1.25$

It can be easily extended but with a predictable but not controllable loss of precision.

NESSAI' domain (without iteration) appears to be irregularly shaped for angle values between -0.25 and 0.25, but it seems to be dense enough for values between 0.25 and 1.25. The prosecution b VERIF seems to be a non-necessary refinement since NA is not very long.

The change of central table has shown that the initial list of proofs is an important contributing factor to the success of the method without iteration. A greater density of points for the values of $0_o 0_1, 0_2$ between -0.25, 0.25 may be the solution to skip any iteration.

Conclusion

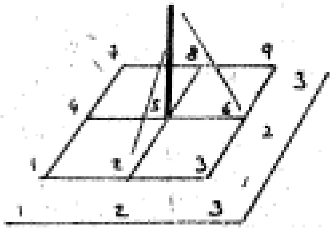
At the present time the program has not been tested in real time. For that purpose it will have to be rewritten into a faster lower language. Certainly at that point new improvements will be found. One may for example wish to have two modes of search, slow and fast. In the faster mode, the search can be ended once the first configuration has been found. Another possible way to speed up the process is to skip a few steps at the beginning of the program.

The manipulation of lists, as it is performed now, requires some storage. That storage is certainly the price to be paid for the simplicity of the computations on each element.

The essential merit of the program remains the ability of finding several configurations of the arm for one goal.

I propose the following way to describe the different solutions.

Each segment has nine discrete positions with respect to the ~~next~~ to the shoulder, only the first segment has three positions with respect to the shoulder. The hand is considered as a fifth segment with three positions.



A configuration of the arm is represented either by $(x_1\ x_2\ x_3\ x_4\ x_5)$

with $x_5, x_1 = 1,2,3$

$x_2, x_3, x_4 = 1,2,\ldots,9$

or by

(u_1,u_2,u_3,u_4) (v_2,v_3,v_4,v_5)

with

$$u_1,u_2,u_3,u_4,v_1,v_3,v_4,v_5 = 1,2,3$$

The two descriptions are equivalent:

$$x_1=u_1$$

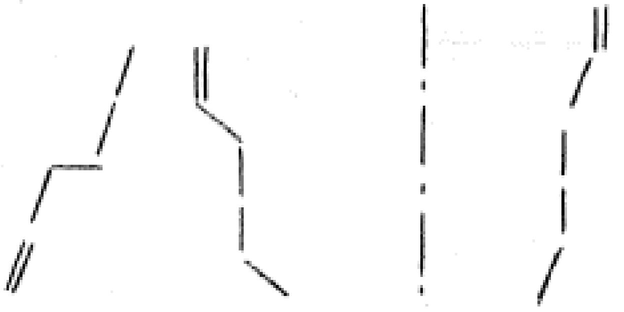
$$x_i=u_i+(v_i-1)3 \quad 2 \leq i \leq 4$$

$$x_s=v_s$$

The second description is more suitable for a graphical representation than the first one, one configuration of the arm can be represented by two linear figures to be drawn among 33.

Example:

(18243) ≡ (1,213) (312,3) ≡ (1,213) sym (132,1)



(1,213) (312,3) sym (132,1)

the two figures are not unlike the two projections of the arm onto two perpendicular planes.

The following use could be made of such representation:

- the goal is given by a point, the matrix orientation of the hand and some ideas about the configuration of the arm.

  These ideas may be expressed using the foregoing representation.

- The program on the new arm proposes different configurations for the goals (point, matrix orientation) which may be expressed with the same representation.
- The configurations are chosen which have equivalent or close descriptions.

Remark: such a discrete representation suggests a new way to describe the new arm, as a path through a three-dimensional graph (or five-dimensional).

This may be useful later when planning the activities of the arm will become less abstract.

<u>Another arm?</u> - In Chapter II the protein chain offers a model for an arm with a large number of degrees of freedom.

Its usefulness still depends on an open question: Does there exist simple functions (like a distance) here for which the only extrema can be solutions?

If the answer is yes - that is what I had to believe but cannot prove it now - is adding new degrees of freedom, the way to get rid of the disturbing saddle-like points?

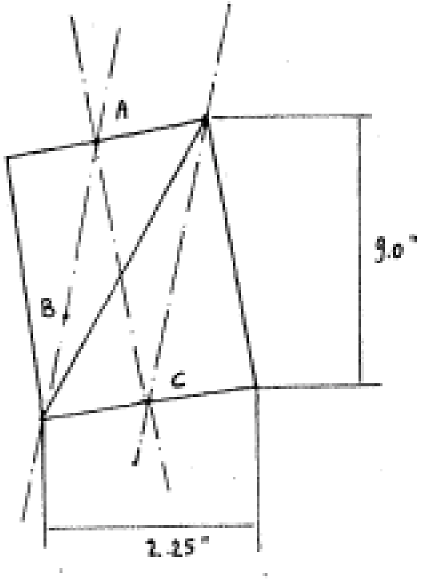
There are two directions of improvements which depends more on the mechanical engineer, but where gears, bar and programming should be studied at the same time.

The symmetry of the arm should make it describable in geometrical terms for any reasonably large number of degrees of freedom.

The physical limitations have to be pushed as far as possible (to gain here is exponential). It is not the easiest problem if one wants to have a device powerful enough with a reasonable size unless one decides to send a weak giant to the moon or to a space station.

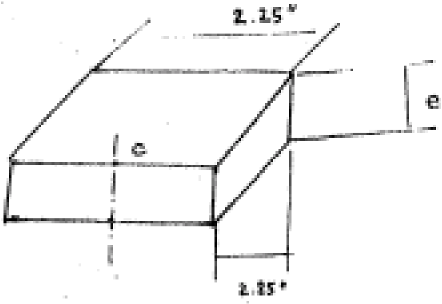
ANNEX I

Numerical data on the Arm

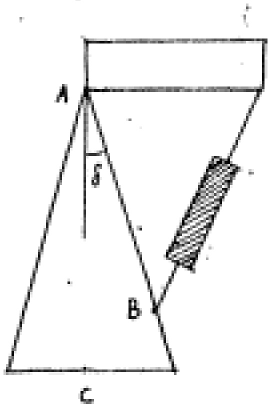


segment

plate



either 1.063" or 0.84375"
(joints 1 and 2)



δ= 7°88'2"

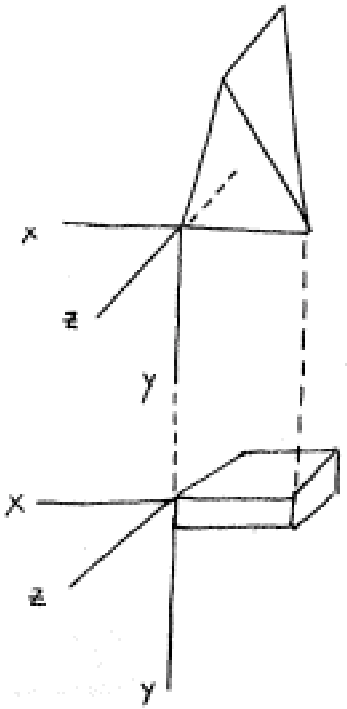
Cylinder length varies between

7.30" and 10.30" .

AB = 8.277"

ANNEX II

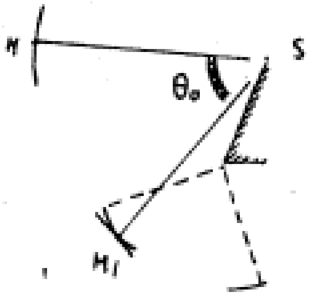
Here is outlined what three of the major features of the program do.



The coordinate axes were defined as they are on the figure, attached to a segment or to a plate. All vector coordinates are contravariant.

TMI

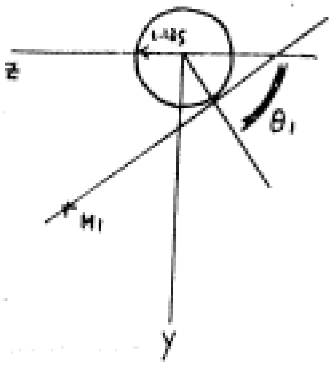
x,y,z are the coordinates of M in a system attached to the shoulder.S,M,M1 are in the same plane perpendicular to the z-axis of the first segment.If it exists θ_0 is the difference between the angle-coordinate of M and of M1 in the plane.The angle-coordinate of M1 is solution of the equation:



$$1.125 \sin \theta + 8.85829 \cos \theta = \frac{x^2 + y^2 + (z + 1.125)^2 - 22.78}{2\sqrt{x^2 + y^2}}$$

where

$$22.78 = (10.12503)^2 - [(1.125)^2 + (8.85829)^2]$$



10.12503 being the radius of the sphere approximating L_{1s0} .

y_1 ,z_1 being two of the coordinates of M,in a system attached to the first segment , θ is solution of the equation:

$y_1 \sin \theta - (1.125 + z_1) \cos \theta = 1.125$

x_2 , y_2 being two of the coordinates of M,in a system attached to the first plate, θ_2 is given by the expression:

$$\text{Arctg}\ \frac{x_2\ 9.9213 - y_2\ 1.125}{x_2\ 1.125 + y_2\ 9.9313}$$

So far TM_1 has been an approximate way of solving the following system of equqtions,where θ_0 θ_1:θ_2 are the unknown :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 & \cos \theta_0 & \sin \theta_0 \\ 0 & -\sin \theta_0 & \cos \theta_0 \\ 1 & 0 & 0 \end{bmatrix} \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 \\ 0 & \sin \theta_1 & \cos \theta_1 \end{bmatrix} \left\{ \begin{bmatrix} 0 & \cos \theta_2 & \sin \theta_2 \\ 0 & -\sin \theta_2 & \cos \theta_2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1.125 \\ 9.9213 \\ -1.125 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 0.06 \\ 0.0 \end{bmatrix} \right\} \begin{bmatrix} 1.125 \\ 9.9213 \\ -1.25 \end{bmatrix} \right\}$$

The corrections made on θ_2 and θ_1 are given in the text.

It appears that any input value θ_{2i} gives two output values θ θ_{o}'.One is very close to θ_{2i}, the other is more or less close to $-\theta_{2i}$.This means when we try to guess θ_{2i} from θ_{2o} we have to consider only the part of the figures close to the line

$$\theta_{2i} - \theta_{2o} = 0 ,$$

where by chance we have the best approximation for θ_{2i}.

VERIF

checks the following conditions:

1. Hinges X_2 and X_2' are,onthe same plate, perpendicular :

$$\vec{X}_2.\vec{X}_2' = 0$$

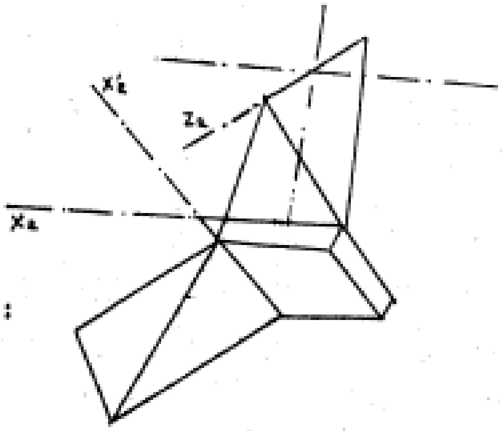
$$\begin{aligned}\vec{X}_2 =(&\cos\theta_0 \cos\theta_2 + \sin\theta_0 \cos\theta_1 \cos\theta_2 \;, \\ &\cos\theta_0 \cos\theta_1 \sin\theta_2 - \sin\theta_0 \cos\theta_2, \\ &\sin\theta_1 \sin\theta_2)\end{aligned}$$

X_2' is to the hand what X_2 is to the shoulder .If(A)is the matrix whose columns are the vectors defining the direction of the hand:

$$\vec{X}_2' = (A)\ \vec{X}_2$$

The condition is :

$$\vec{X}_2(A)\vec{X}_2 = 0 \ .$$



2. The plate can only rotate within limits:

$$-\sin 0.4 \leqslant \vec{Z}_2 \times \vec{X}_2'' \leqslant \sin 1.3 \ .$$

3.The third seg-ment can only rotate within limits about X_2' :

$$-\sin 0.4 \leqslant (\vec{X}_2 \times \vec{X}_2') \times \vec{Y}_2' \leqslant \sin 1,3 \ .$$

2 and 3 simply express the conditions;

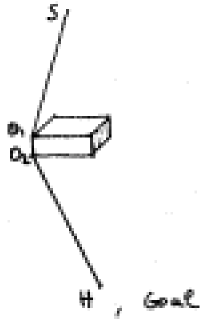
$$-0.4 \leqslant \theta_3 \leqslant 1.3$$

$$-0.4 \leqslant \theta 4 \leqslant 1.3 \quad .$$

NA

The arm is equivalent to atwo-bar linkage,whose segments are $SO_1$and O_2H.

$\overrightarrow{SO}_] + \overrightarrow{O_1H} = \overrightarrow{SG}$



where G is the goal.

$\overrightarrow{SO_1} = \overrightarrow{SO_1}(\theta_0,\theta_1,\theta_2)$

$$\overrightarrow{O_2H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & \sin\theta_3 & \cos\theta_3 \end{bmatrix} \left(\overrightarrow{O_1'O_2'} + \begin{bmatrix} \cos\theta_4 & \sin\theta_4 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \overrightarrow{O_2'H'} \right)$$

$\overrightarrow{O_1'O_2'} = (0,\ 0.84375, 0)$

$\overrightarrow{O_2'H} = \overrightarrow{O_2'H}\ (\theta_5,\theta_6,\theta_7)$

The coordinates of G in the system attached to the segment are x,y,z, the coordinates of H in the system attached to the third segmentξ,η,ζ.

θ_4 is a solution of the equation:

(1) $\quad x \cos\theta_4 + y \sin\theta_4 = \xi$

θ_3 is:

(2) $$\mathrm{Arctg}\,\frac{(1.063 + y\cos\theta_4 - x\sin\theta_4)\ \ -z\,\eta}{(1.063 + y\cos\theta_4 - x\sin\theta_4)\ \ +z\zeta}$$

NA solves (1) after having computed x,y,z,ξ,η,ζthen applies (2).

ANNEX III

BEELER'S PROGRAM

The goal is given by the coordinates of the point to reach x,y,z,two„orientation measures",functions of the coefficients of the matrix of orientation of the hand, M_1,M_2.

x,y,z,M_1,M_2 are approximated by close formulas of the following type:

$$x\ (p_1,\ldots,p_8) = x_0 + [\Delta p_1 \frac{\partial x}{\partial p_1} + \Delta p_2 \frac{\partial x}{\partial p_2} + \ldots + \quad \Delta p_8 \frac{\partial x}{\partial p_8}]$$

$$+ [\Delta q_{13} \frac{\partial^2 x}{\partial^+ p_1 \partial^- p_3} \quad + \Delta q_{35} \frac{\partial^2 x}{\partial^+ p_3 \partial^- p_5} \quad + \Delta q_{57} \frac{\partial^2 x}{\partial^+ p_5 \partial^- p_7}$$

$$+ \quad \Delta q_{24} \frac{\partial^2 x}{\partial^+ p_2 \partial^- p_4} \quad + \Delta q_{46} \frac{\partial^2 x}{\partial^+ p_4 \partial^- p_6} \quad + \Delta q_{68} \frac{\partial^2 x}{\partial^+ p_6 \partial^- p_8} \quad]$$

in which $\frac{\partial x}{\partial p_1}$ is an estimation of the partial derivative,and $\frac{\partial^2 x}{\partial^+ p_1 \partial^- p_3}$ is defined for p_1 increasing and p_3 decreasing.

The main part of the program computes,using Lagrange multipliers, p_1,' p_2, p_8, q_{13},..., q_{68}such

$$(\Delta p_1 + \Delta q_{13})^2 + (\Delta p_2 + \Delta q_{24})^2 + (\Delta p_3 + \Delta q_{35} - \Delta\ q_{13})^2 \ + (\Delta p_4 + \Delta q_{46} - \Delta q_{24})^2$$

$$(\Delta p_5 + \Delta q_{57} - \Delta q_{35})^2 + (\Delta p_6 + \Delta q_{68} - \Delta q_{46})^2 + (\Delta p_7 - \Delta q_{57})^2 + (\Delta p_8 - \Delta q_{68})^2$$

be minimum.

The result is a vector in the 8-dimensionnal parameter-space,each component being the amount by which a potentiometer has to be adjusted.
Various tests are then performed in order to see if the point in the parameter-space will move outside the physical limits.If so it may be brought back either onto the limit _at the next iteration the corresponding parameter will not vary_ or inside the limit at the closest point to the goal on a straight line defined by the starting position and the point outside.

The program iterates from the new position.

If the preceding method fails to bring the point closer to the goal a systematic step by step exploration (with decreasing steps eventually) is made about the starting point, in certain privileged directions.

ANNEX IV

A Small Program written in MIDAS,to use the potentiometers.

```
                TITLE ARM

                A=1
                B=2
                C=3
                T=10
001 000         D=T+1
                P=17

                TTYCHN==1
                IMXCHN==2
                OMXCHN==3

001 007 BEG:           MOVEI P,PDL-1
001 010                .OPEN TTYCHN,[1,,(SIXBIT \TTY\)]
                       .VALUE 0
001 011                .OPEN IMXCHN,[2,,(SIXBIT \IMX\)]
                       .VALUE 0
001 012                .OPEN OMXCHN,[3,,(SIXBIT \OMX\)]
                       .VALUE 0
001 003 LOOP1:         MOVSI A,-10
001 004 LOOP2:         MOVEI B,140(A)
001 006                MOVEM B,IBUF(A)
001 022                AOBJN A,LOOP2
001 005                MOVE A,[-10,,IBUF]
001 003                .IOT IMXCHN,A
001 003                MOVSI A,-10
001 004 LOOP3:         MOVEI B,50(A)
001 004                LSH B,12.
001 006                IOR B,IBUF(A)
001 006                MOVSM B,IBUF(A)
001 026                AOBJN A,LOOP3
001 005                MOVE A,[-10,,IBUF]
001 003                .IOT OMXCHN,A
001 003                MOVSI A,-10
001 004 LOOP4:         MOVEI B,50(A)
001 006                MOVEM B,IBUF(A)
001 035                AOBJN A,LOOP4
001 006                MOVE A,[-10,,IBUF]
001 003                .IOT IMXCHN,A
001 006                MOVEI T,14
001 055                PUSHJ P,TYO
001 003                MOVSI A,-10
001 006 LOOP5:         MOVE T,IBUF(A)
001 056                PUSHJ P,PRINT
001 052                PUSHJ P,CRLF
001 044                AOBJN A,LOOP5
001 003                MOVEI A,30.
001 003                .SLEEP A,
001 021                JRST LOOP1

001 006 CRLF:          MOVEI T,15
001 055                PUSHJ P,TYO
001 006                MOVEI T,12
001 005 TYO:           .IOT TTYCHN,T
001 003                POPJ P,

001 005 PRINT:         IDIVI T,6
001 007                HRLM D,(P)
001 003                SKIPE T
```

```
001       001 056            PUSHJ P,PRINT
002       001 000            HRRZ T,(P)
003       001 005            MOVEI T,"0(T)
004       001 055            JRST TYO
005
006                 ISDP:    BLOCK 10
007                 PDL:     BLOCK 20
010       001 014 END BEG
```

ANNEX VI

How to use the program on the ITS of the A.I. group at Project MAC.

The program is on the DEC tape JYA under the name MOVE NEWARK, the initial table is under thenname TABLE ARM. One has just to call a certain number of numerical functions, from the binary file LISP NUM 1 on the same tape.

The procedure is the following, in DDT

```
LISP$J
dev $L LISP NUM1  CR
$G
```

The machine then asks for allocation.

The interesting functions for an experiment are:

NRECH which takes as arguments the coordinates and the matrix orientation of the goal, and returns either the angle-values of the solutions or NO SOLUTION

TEST takes as arguments eight angle-values of the parameters, gives the same type of result as NRECH does.

NESSAI takes the same arguments as TEST, returns the solutions expressed in angle and potvalues, and coordinates and matrix orientation of the goal for each solution.

SOLUTION takes as argument the coordinates of a point in a system attached to the hand, and the angle-values of the parameters; returns coordinates and matrix orientation of the goal.

REFERENCES

1. Ernst, Heinrich A., MH-1, A Computer-Operated Mechanical Hand, Doctoral Thesis, MIT, December, 1961.

2. Favard, J., Cours D'Analyse De L'Ecole Polytechnique, 1962-1963, Ecole Polytechnique Ed.

3. Feigenbaum, E. A., and Feldman, J., Computers and Thought, McGraw-Hill, 1963.

4. Ferrell, W. R., "Remote Manipulation with Transmission Delay," IEEE Trans. Human Factors in Electronics, HFE-6 (1), September 1965, pp.24-32.

5. Greene, P. H., "New Problems in Adaptive Control," Chap. 18 of Computer and Information Sciences, ed. Wilcox, R. E., and J. T. Tom, Washington: Spartan Press, 1963.

6. Levinthal, C., Molecular MOdel-Building By Computer, Scientific American, June 1966, Vol. 214, No. 6, pp.42-52.

7. Levinthal, C., Barry, C. D., Zwich, M., Computer Graphics in Macromolecular Chemistry, MIT, Department of Biology, 1968.

8. Mergler, H.W., and P. W. Hammond, "A Path Optimization Scheme for a Numerically Controlled Remote Manipulator," 6th Annual Symposium of the IEEE Human Factors in Electronics Group, May, 1965.

9. Minsky, M., "Steps Toward Artificial Intelligence," Proc. IRE, Vol. 49, January 1961, pp.1-30.

10. Minsky, M., The New Arm, Device MA-3, Hardware Memo 4, Artificial Intelligence Project, May, 1966.

11. Minsky, M., and Papert, S., Research on Intelligent Automata, Status Reports, I and II, August 1966, September, 1967. MIT, Project MAC.

12. Minsky, M., and Selfridge, O., Learning in Random Nets, Proc. of the Fourth London Symposium on Information Theory, London, Butterworth.

13. Project MAC, Artificial Intelligence Memo Series, Massachusetts Institute of Technology.

14. Piaget, J., Etudes d'Epistemologie Genetique, P.U.F., Paris France.

15. Sheridan, T.B., "Use of Artificial Computation Loops within Human Loops for Remote Manipulation," unpublished Memo. Oct.28, 1964.

16. Tomovic, R., and R. Petrovic, "Target Approach in Biological Systems," IEEE Trans. Automatic Control, AC-10 (1), January, 1965, p.22.

17. Whitney, D.E., State Space Models of Remote Manipulation Tasks, DST 70283-5, Engineering Projects Laboratory, Department of Mechanical Engineering, MIT, January 1968.

18. Wooldridge, D.E., The Machinery of the Brain, McGraw-Hill Paperbacks, 1963.