



**HAL**  
open science

# Semi-active guiding systems in surgery. A two-dof prototype of the passive arm with dynamic constraints (PADyC)

Jocelyne Troccaz, Yves Delnondedieu

► **To cite this version:**

Jocelyne Troccaz, Yves Delnondedieu. Semi-active guiding systems in surgery. A two-dof prototype of the passive arm with dynamic constraints (PADyC). *Mechatronics*, 1996, 6 (4), pp.399-421. 10.1016/0957-4158(96)00003-7 . hal-03692401

**HAL Id: hal-03692401**

**<https://hal.science/hal-03692401>**

Submitted on 9 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Semi-active guiding systems in surgery.  
A two degrees of freedom prototype of the  
Passive Arm with Dynamic Constraints (PADyC)

Jocelyne Troccaz, Yves Delnondedieu  
TIMC/IMAG Laboratory - Faculté de Médecine (IAB)  
Domaine de la Merci - 38706 La Tronche cedex - France

**Abstract**

Guiding systems for surgical applications range from passive systems to active ones depending on the level of autonomy left to the surgeon. Passive systems provide the surgeon with suitable information to compare the executed strategy with a planned one whilst active systems autonomously perform a part of the surgical strategy. Depending on the selected technology, guiding systems may be more or less accurate, more or less safe, more or less user-friendly. Actuated robots are generally very accurate but raise safety and ergonomics issues; passive systems (optical localizers, motorless encoded arms, ...) are well-suited to navigation but their use is difficult for executing complex surgical strategies. We have designed a new type of mechanical guiding system based on the use of dynamic constraints; this passive arm is actuated by the surgeon who, keeps, therefore, fully involved in the execution of the surgical strategy. At each instant, the motions proposed by the operator are "filtered" with respect to the task, before being transmitted to the arm. This system seems a good answer to the guided execution of potentially complex strategies in an accurate and ergonomical way. In this paper, we present the principles of the PADyC arm and describe experiments already done with a two degrees of freedom prototype.

**1 Introduction:**

Guiding systems for MRCAS may be passive, semi-active or active, depending on the level of autonomy they leave to the surgeon. Passive systems give the surgeon a feedback of what he is currently doing; semi-active systems guide the surgeon (visually or mechanically) towards the planned surgical strategy; active systems perform by themselves the planned strategy. Mechanical guiding is done either by passive arms or by actuated robots. There is no doubt that the still

limited use of robots in medicine and surgery is partly due to safety problems. Systems for computed aided surgery must be safe both in their software and in their hardware components. In this work, we are mainly concerned with hardware problems. Using redundant internal and/or external sensors is part of the solution to verify whether a robot correctly executes a planned task. Nevertheless, actuated robots may execute unexpected motions very fast in case of hardware breakdown, and the reaction time of the control system may lead to serious problems. For example, one can imagine a surgical tool such as a knife or a laser at the end of a robot, that would be positioned very close to a structure that must not be damaged at any price (e.g. an important vessel, a nerve, or a region in the brain, among others). Even for a slow robot having a maximal speed of 1cm/s, if one joint is damaged, an authorized unexpected motion of 1mm would correspond to a maximal detection and reaction time of 0.1s which is difficult to achieve.

Passive arms that have no actuators and for which the trajectory is imposed by an operator do not present such a risk. All passive systems, mechanical or optical, are very useful for providing information about where a tool is (for instance, a surgeon moves his tool and just looks for its position with respect to MRI data on a screen [9]). Unfortunately, they are much less helpful for reaching a pre-computed position. In particular, moving a tool by means of a mechanical passive system towards a position is very difficult because the mechanical structure naturally promotes some directions of motion. A possible solution to achieving a final position may consist in moving the axes one by one (e.g. stereotactic frame or articulated decoupled systems [11]). Note that for the mechanical passive systems, a good accuracy in positioning is reached at the expense of a tedious use of fine adjustment mechanical devices such as verniers.

Optical passive systems (see [1]) make it easier to manipulate the surgical tool and to reach a pre-computed target (point or line), although delivering an up-to-six-dimensional information to the operator on a computer screen, as it is classically done, is from an ergonomic point of view very open to criticism. One drawback of optical systems comes from the difficulty of keeping the tool in the correct position; this may demand from the surgeon a particular attention during a long time (for drilling a hole for example). The other drawback is related to the fact that the accuracy of such a system is upper bounded by the manual precision of the surgeon.

## 2 Related works on safety in surgical robotics:

In the literature, different solutions have been devised for dealing with the safety problem in different surgical applications for which mechanical guiding systems are used.

[7] presents a semi-active system holding a linear guide for stereotactic neurosurgery. It makes use of a modified actuated industrial robot: high reduction



gear ratios allow for both very slow motions and irreversibility. Two phases can be considered: when the robot comes towards a planned target and when it is at the specified location. During the motion phase, the low speed reduces overheating and makes it easier to stop the robot in case of potential unexpected motions; nevertheless, such a motion cannot be executed without a person paying attention to what is going on and close to the panic button. During the second phase, the irreversibility permits the power to be cut off and the brakes locked on before introducing a needle in the guide. Moreover, software tests have been added to continuously control the encoder data. Since we expect no motion, the variation of the joint variables are tested. Any motion triggers an alarm. The surgeon can remove the needle from the brain, put the power on and ask for a new positioning, keeping close to the emergency button.

[8] presents an active system for total hip replacement. The robot is used as a machining tool to make the bone cavity as close as possible to the implant shape. In this system, the robot control subsystem performs extensive safety checking and monitors the cutter force to make sure that the robot does not exert excessive force on the patient [5]. External optical sensors have also been tested as a redundant control system in order to control the robot trajectory, but their benefit has been limited by the visual inaccessibility of the internal surgical field.

[3] makes use of a hardware device in order to constrain the possible motions of a mechanical guiding system for transurethral resection of the prostate. An active PUMA robot has been used for prototyping the system before the design of a passive manipulation aid. The major drawback of such an approach lies in the fact that one mechanical device has to be designed for each possible task. Moreover, for some complex tasks, such a design is not always possible. [4] presents a similar approach in which the task-dependant hardware is replaced by a programmable robot whose possible force-controlled motions are constrained by the task. The operator applies a force to the end-effector in a desired direction of motion. The motion is executed by the robot if it is compatible with a planned strategy. Force controlled robots make a better cooperation with the surgeon possible, and their use may add some safety features; nevertheless, since they are actuated, their potential problems are the same as for position-controlled robots.

Using actuated devices raises obvious safety issues since risks are not entirely eliminated except for very specific cases. As a consequence, [10] proposes another system based on the use of an improved passive mechanical arm. Computer controlled brakes allow for the automatic setting of partial torque (10 to 20 percent) when the arm is within 1mm of the target and for the full torque setting when the target is reached (0.2 mm). Using such active brakes together with simple sonar feedback, facilitates the operator's positioning task.

In summary, using passive aids is riskless but it is necessary to augment the capabilities of such systems to make them really mechanically semi-active in order to guide the surgeon in a better way for his positioning tasks. Furthermore



they should enable both to execute a complex pre-computed trajectory and to restrain the possible motions to a pre-defined sub-region of the workspace. The new type of arm that we propose in that paper is intended to provide such a help.

Section 3 describes the basic principles in the design and programming of the PADYC arm. Section 4 presents the model underlying this approach. Section 5 describes the current system and experiments are presented in section 6. Further developments are discussed in section 7 before concluding.

### 3 Principles of the PADyC:

#### 3.1 Basic architecture:

In order to cope with these safety problems and because the present major medical need is not concerned with fully autonomous robots, we are currently designing a new type of mechanical guiding system, named "passive arm with dynamic constraints". Such a system is intended to combine the advantages of both passive and active systems: the complexity of tasks potentially achieved by an active system with the limited risk involved by a passive one.

The system is a non-actuated arm which has to be moved by an operator. Contrary to a standard passive arm, the operator will be helped by the system itself to perform the planned task. The basic idea consists in dynamically constraining, at each instant, the direction of possible motions of each joint, depending on the task to be performed. In other words, the motions proposed by the operator are filtered by a mechanical system before being transmitted to the arm (cf. figure 1).

For each joint, the classical actuator is replaced by a system that allows for four possible functions:

- **F1:** the joint can be moved in the + and - directions
- **F2:** the joint can be moved in the + direction but cannot be moved in the - direction
- **F3:** the joint can be moved in the - direction but cannot be moved in the + direction
- **F4:** the joint cannot be moved

In order to realize these functions, each axis is equipped with two freewheels authorizing motion in opposite directions, and with a mechanical system that allows each of the freewheels to be clutched independently. The four functions are obtained by a coordination of the two clutching systems associated with the two freewheels. Such a constrained axis is integrated in a classical revolutive or prismatic joint assembly including encoders and reducers.

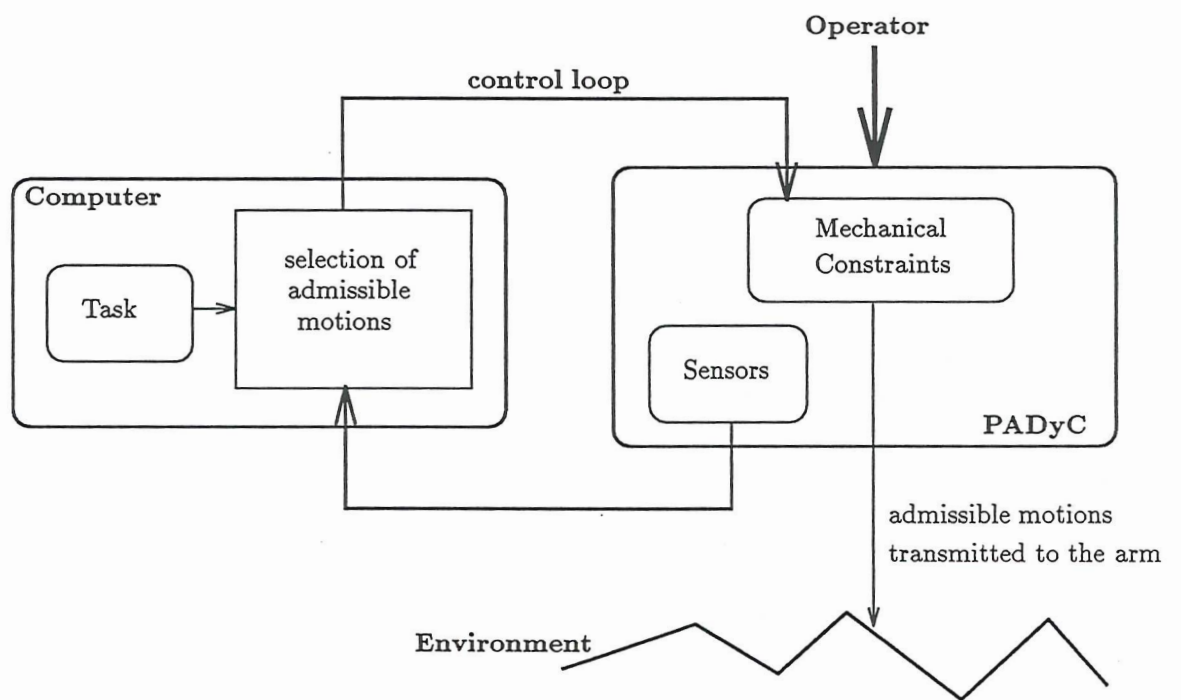


Figure 1: PADyC's Principle

Many technologies are available for realizing a constrained joint. Our design is presented in section 5.1.

Because there is no actuator, the system cannot move on its own; the only energy is the one provided by the operator. However the arm can physically guide the operator, since the possible motions are dynamically constrained in order to execute a pre-computed task. The constraints may be more or less mandatory depending on the expected accuracy of the task.

## **3.2 Interacting with PADyC:**

The whole set of joints are computer-controlled, in real time, so that the operator can use any of these four modes. He can:

1. freely move the arm
2. reach a pre-defined position
3. follow a pre-defined trajectory
4. stay in a pre-defined region

### **3.2.1 "Free" mode:**

In this case, the arm behaves as a classical navigator. All the freewheels are declutched, so that the arm can be used as a 6D position sensor. The operator can use such a device to move and track the position of an instrument or of a sensor. For other clinical domains for which learning is difficult, "teaching by showing" can be very useful. A possible use of the arm consists in recording a free sequence of motions executed by an experienced operator. Then, a novice operator can "play" again the recorded task in a constrained mode (typically the trajectory mode, see further).

### **3.2.2 "Position" mode:**

The problem is to reach a goal position corresponding to a given configuration of the arm. Such a mode is very useful for many clinical applications which require a linear action of the surgical instrument (drill, needle, electrode, trocar, etc...). That is the case for example in stereotactic neurosurgery or in orthopaedics. In these cases, the arm may be used to accurately position a mechanical guide. When that position is reached, all the freewheels are clutched in order to keep that position stable. The surgeon may then manually execute the surgical act.

### **3.2.3 "Trajectory" mode:**

In this mode, a goal (position or set of positions) has to be reached while taking care of the executed trajectory. Such a mode is useful either when the surgical



gesture is complex (for example in plastic and reconstructive surgery), when the access towards the surgical goal is delicate because of the surrounding anatomic structures, or when the workspace of the arm is reduced because of objects in the neighborhood.

#### **3.2.4 "Region" mode:**

In this mode, one or several control points on the surgical tool or on the arm are constrained to stay in a pre-defined region. Inside the region, any motion is free whilst some of them are forbidden on the border. This mode is particularly suitable for resection operations, or osteotomies. It also enables anatomic obstacles to be defined so that they can be avoided; this is particularly interesting when the surgical procedures are too complex to be fully automated.

## **4 Modelling:**

A task is generally defined in cartesian space from multi-modal data of the patient. We assume that the rigid transform relating the model of the patient to the model of the arm is known, thanks to calibration procedures (see. [7] for instance). The task may be described as constraints to be applied to a mathematical object; this is the case if we are only interested in the position and orientation of the tip of a surgical tool. The task may also be defined as constraints to be applied to a physical object: this is the case if we want to keep a surgical tool or the robot itself in a certain region. A basic issue of the PADyC lies in the ability to map the task into the joint space of the robot in order to determine, at each instant, the constraints to be applied to each joint. This is a difficult problem that has been extensively studied in the field of collision-free path planning [6] and for which some solutions have been proposed and implemented.

### **4.1 Task definition:**

#### **4.1.1 Position:**

A goal position may be defined in cartesian space or in joint space. In the former case, a posture (position and orientation) of the end-effector is transformed into a finite set of goal configurations of the arm. We will consider that one configuration is selected as the goal. If the robot is not in the same type of configuration, it has to be moved first to an intermediate position where reconfiguring is possible.

#### 4.1.2 Trajectory:

Learned trajectories are represented as discrete sets of recorded configurations. Such a set may be interpolated, using for example spline functions, in order to get a continuous representation of the trajectory in joint space. The trajectory may also be defined in cartesian space. A corridor is associated with a trajectory. The executed trajectory must remain inside that corridor. The thinner the corridor is, the more accurate the execution of the trajectory will be. If the trajectory and its corridor have been defined in cartesian space, they have to be mapped into joint space.

#### 4.1.3 Region:

A region is defined in the cartesian space. It represents authorized positions in space; as stated above, these positions may concern the tip of an instrument, the instrument or the robot itself. The region may have an arbitrary shape. In fact, the difference between trajectories and regions is thin because of the notion of a corridor associated with the trajectories. The difference lies in the fact that the progress along the trajectory restrains the set of accessible configurations in the corridor, whilst every configuration of a region is accessible even if it has already been visited.

Classical algorithms coming from Computer Aided Manufacturing could be used to transform a region in a trajectory when a "machining" procedure is involved; that is the case for example, to prepare a bone cavity, before the insertion of a prosthesis.

### 4.2 Constraint modelling:

For the sake of simplicity, we will consider, in the following, an arm with  $n$  joints and  $n$  degrees of freedom (d.o.f.).

The control of the arm consists in applying to each joint, in real-time, the constraint corresponding to the current configuration of the arm. Being given a model of the task, to each configuration  $Q_i = (q_i^1, q_i^2, \dots, q_i^n)$  of the arm, must be associated a constraint vector describing the clutching status of each of the  $2n$  freewheels  $C(Q_i) = C_i = (c_1^-, c_1^+, c_2^-, c_2^+, \dots, c_n^-, c_n^+)$ .  $c_j^{-/+} = 0$  (resp.  $c_j^{-/+} = 1$ ) if the negative or positive freewheel is unclutched (resp. clutched); in other words if the negative or positive rotations are authorized (resp. forbidden).

### 4.3 Execution of a task:

The constraint vector characterizes the possible evolution of the arm configuration as a function of time. It corresponds to the subdivision of the joint space in subspaces; let us call them "generalized quadrants". For each configuration, there exists  $4^n$  such subspaces for a  $n$  joints arm (each joint can be in four

states). The smallest generalized quadrant is a set containing only the current configuration whilst the largest one is the whole joint space. The constraint vector defines the generalized quadrant in which the motion has to take place.

Configurations and constraints may be rewritten as functions of time  $Q(t)$  and  $C(t)$ . Let us consider the instant  $t_i$  where the  $i^{th}$  constraint becomes effective on the arm and the sample time  $\Delta t$ , corresponds to the time necessary to determine and execute the next constraint vector.

In order to clarify, let us consider a simple 2 joints, 2 d.o.f. arm. At instant  $t$ , the arm is at the configuration  $Q(t) = (q_1, q_2)$ .  $C(Q(t)) = C(t) = (1, 1, 1, 1)$  corresponds to clutching all the freewheels, therefore forbidding any motion of the arm; in that case  $Q(t + \Delta t) = Q(t)$ .  $C(t) = (1, 0, 0, 1)$  insures that  $Q(t + \Delta t)$  is in the dashed region.  $C(t) = (1, 1, 1, 0)$  insures that  $Q(t + \Delta t)$  is on the bold half straight line. Figure 2 shows a possible configuration at  $t + \Delta t$  for  $C(t) = (1, 0, 0, 1)$ .

The constraint also defines the type of trajectory that will be executed within that generalized quadrant; for  $C(t) = (1, 0, 0, 1)$  (see figure 2),  $q_1$  monotonously increases whilst  $q_2$  monotonously decreases between  $t$  and  $t + \Delta t$ . More generally, we have:

$$\begin{aligned} c_j^+(t_i) = 0 \wedge c_j^-(t_i) = 1 &\implies \frac{dq_j}{dt}(t) \geq 0 \forall t \in [t_i, t_i + \Delta t] \\ c_j^+(t_i) = 1 \wedge c_j^-(t_i) = 0 &\implies \frac{dq_j}{dt}(t) \leq 0 \forall t \in [t_i, t_i + \Delta t] \\ c_j^+(t_i) = 1 \wedge c_j^-(t_i) = 1 &\implies \frac{dq_j}{dt}(t) = 0 \forall t \in [t_i, t_i + \Delta t] \end{aligned}$$

Therefore, all the trajectories between  $t$  and  $t + \Delta t$  are not possible, as is shown on figure 3: the plain trajectory leading to  $Q(t + \Delta t)$  is possible whilst the dashed one leading to  $Q'(t + \Delta t)$  is not because it does not fulfill the inequalities associated with the constraints.

Because the operator determines the speed of the arm, the controllability of the system has to be discussed. If this speed is sufficiently high relative to the sample time  $\Delta t$ , the system will not be able to make the operator converge towards a goal position. Additional computer controlled brakes may be a solution to that problem (cf. [12]); a better solution will be presented in the next section.

In any of the four modes, all the joints are not necessarily managed in the same way. For example, let us consider a SCARA architecture arm with a wrist having three revolute degrees of freedom : for a particular task, the position of the end-effector can be fixed by clutching the two freewheels of each of the first three joints, whilst the last three ones can be dynamically constrained in order to control the orientation of the effector.



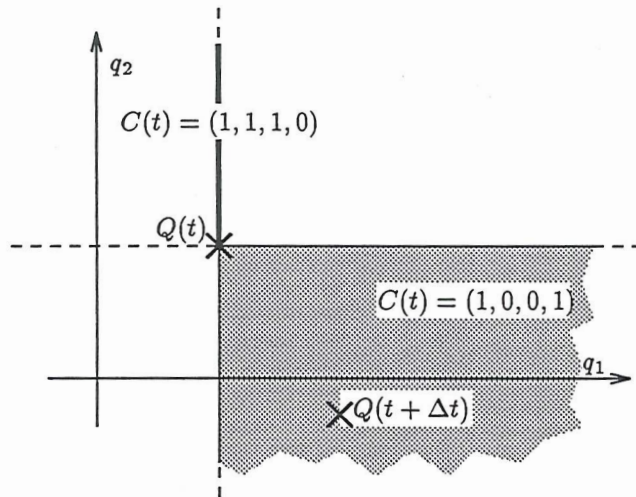


Figure 2: Some examples of constraints.

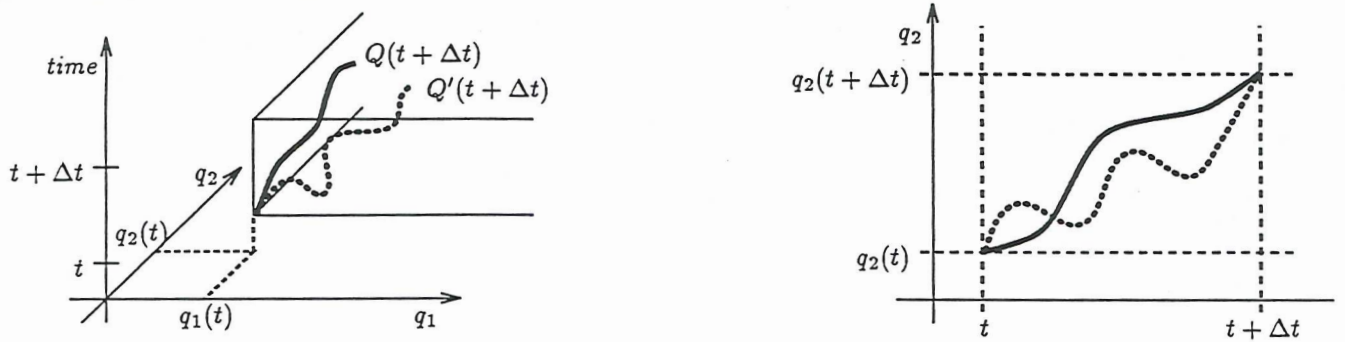


Figure 3: Configuration seen as a function of time.

## 5 Current system:

### 5.1 Architecture:

As stated above, an important feature of the PADyC lies in the choice of a clutching system for the freewheels. We have selected electrical motors to play that role. One motor is associated with each freewheel; depending on the speed of the motor with respect to the user's speed, the motion will be blocked or not. Each motor rotates in the direction that cannot actuate the joint and in that direction only. In other words, it cannot move the external part of the freewheel normally moved by the user (see figure 4). Therefore, no motion is induced by the motors. One advantage of this design lies in the fact that the motors can also play the role of brakes, since the speed in one direction is limited by the motor allowing motion in that direction. If  $\omega_i^+$  and  $\omega_i^-$  are the angular speeds of the two motors associated to joint  $i$  and  $\omega_{user}$  is the angular speed provided by the user, the four functions may be obtained as follows:

- F1:  $\omega_i^+ \neq 0, \omega_i^- \neq 0$
- F2:  $\omega_i^+ \neq 0, \omega_i^- = 0$
- F3:  $\omega_i^+ = 0, \omega_i^- \neq 0$
- F4:  $\omega_i^+ = 0, \omega_i^- = 0$

In fact, in that design, the freewheels are naturally clutched; when a motor rotates at joint  $i$ , the corresponding freewheel is unclutched and the motion is allowed as long as  $|\omega_{user}| < |\omega_i^{-/+}|$ . Thus, because of the mechanical design,  $\omega_{user}$  is bounded. To make the system fully controllable, the user's maximal speed can be servoed as a function of the distance to the goal. This will be detailed in the following section.

We have developed three prototypes; the first one, built with mechanical components from a toy construction set, was dedicated to the validation of the clutching mechanism using motors; the second one allowed the basic speed control of a single joint to be studied. The current prototype is a two degrees of freedom planar arm the purpose of which is to experiment with the different modes and their associated control laws in the framework of a simple joint space. As can be seen on figure 5, the planar arm is composed of two revolute joints. The current end-effector is a pen allowing the basic capabilities of the system to be demonstrated by drawing.

The system architecture is as follows (see figure 6). A PC-AT compatible computer is used for developing the application and interacting with the user. This computer includes a Galil DMC-1060 motion controller which manages the speed control of the motors, the readings of the joint encoders and the control of the four amplifiers associated with the motors. In fact, the DMC-1060 converts the velocity profile into a position trajectory where a new position

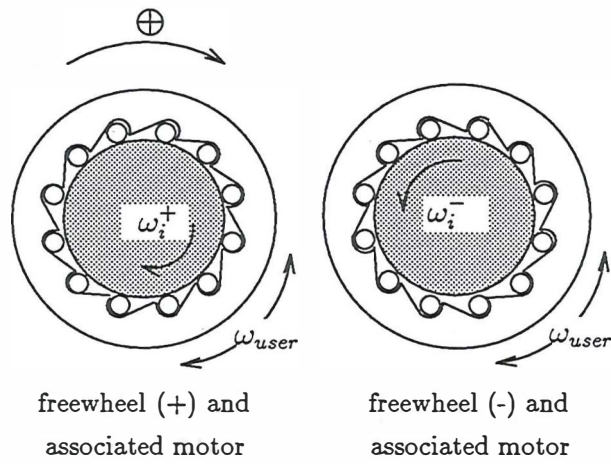


Figure 4: The clutching mechanism.

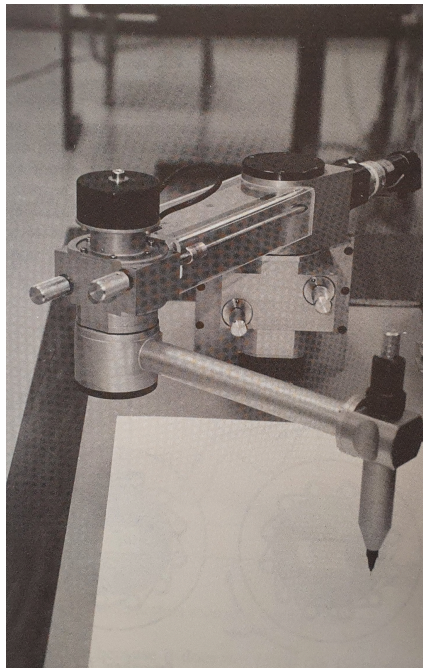


Figure 5: The two d.o.f. prototype.



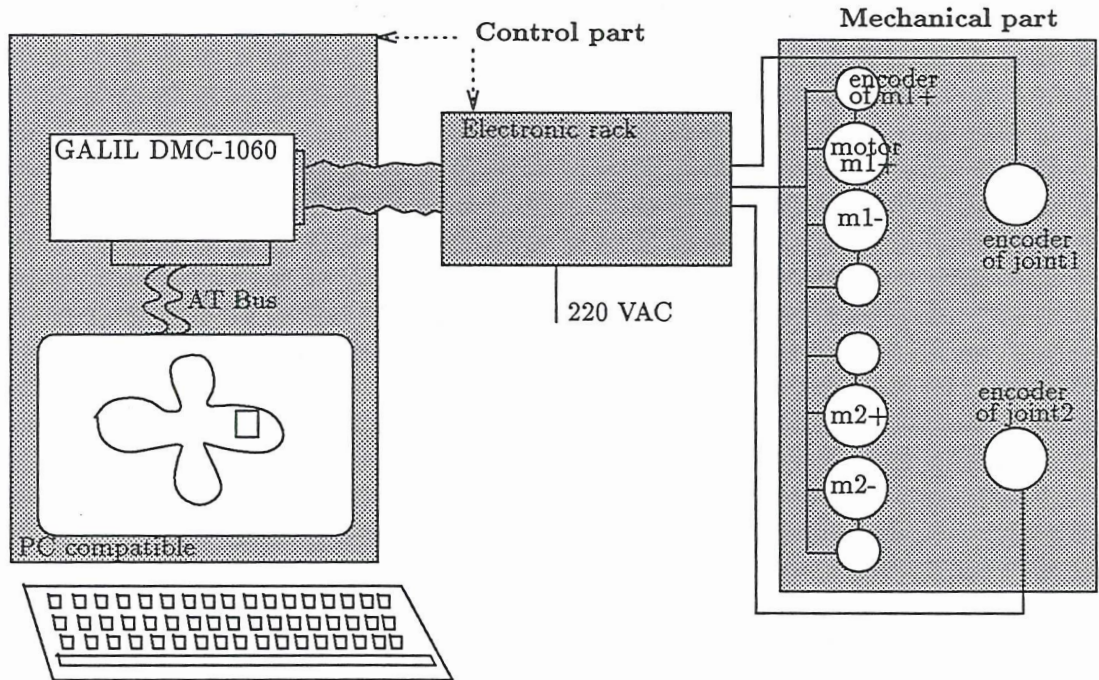


Figure 6: Hardware of the two d.o.f. prototype

target is generated every sample period of the controller (1 ms). This sample period must not be confused with the sample period  $\Delta t$  of the global control loop of the system (cf. figure 1). An additional rack contains the power part of the system, the amplifiers and includes the interface to the encoders. The joints are equipped with off-the-shelf freewheels, and M.T.L. ME50 incremental encoders (10000 lines, i.e. 40000 points, per revolution). The selected motors are Maxon DC motors including reducers and HP encoders. In order to prevent the user from back driving the motors in blocked configurations, worm screws have been added to the mechanical design of the joints.

When PADYC is powered off, no motion is possible since all the freewheels are clutched. In the case of a power failure, the worms screws may be actuated manually in order to enable the manual displacement of the robot out of the workspace.

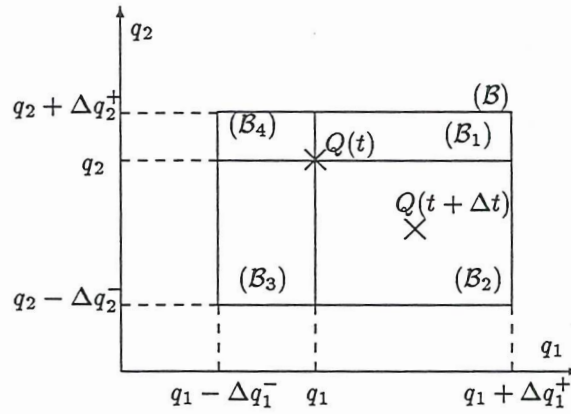


Figure 7: Region of admissible motions during  $\Delta t$ .

## 5.2 Programming modes:

To use the clutching system, the control of the joints consists in computing, in real-time, the speed of the four motors. Because the angular velocity is bounded, the generalized quadrants may be reduced to generalized rectangles (cf. figure 7). The maximum angular displacements are  $\Delta q_i^{-/+}(t) = \omega_i^{-/+}(t)\Delta t$  where  $\omega_i^{-/+}(t)$  is the negative or positive angular velocity of joint  $i$  at instant  $t$ .

In the four modes described below, we have to compute, at each time, the most judicious authorized subset  $\mathcal{B}_i$  of the rectangular region  $\mathcal{B}$  around the current configuration  $Q(t)$  and to set the authorized maximum angular velocity of each joint.

### 5.2.1 "Free" mode:

In this case, the authorized maximum velocity of each joint is constant and chosen sufficiently high to give the operator the feeling of free motion. The arm behaves as a classical passive one unless the user tries to move too fast. He is then constrained to stay below the pre-defined limit of the rotational speed.

### 5.2.2 "Position" mode:

The problem is to reach with accuracy a goal position  $Q_f$  corresponding to some configuration  $(q_f^1, q_f^2)$  of the two d.o.f. arm. The trajectory is not defined, so we can choose only to allow a motion toward the goal in joint space. At instant  $t$ , we select the constraint corresponding to the smallest generalized rectangle that contains the segment  $[Q(t), Q_f]$  (cf. figure 8).

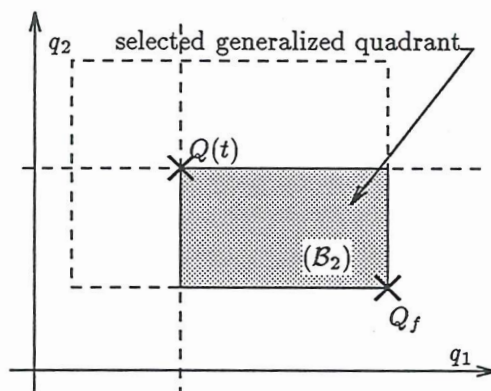


Figure 8: Position mode.

The joint velocity in this quadrant was chosen as follows. Let us consider a non degenerated case. The position  $Q(t + \Delta t)$  at the next sample time must be in the rectangle defined by its opposite corners  $Q(t)$  and  $Q_f$ . The authorized speed of both axes is proportional to the error distance  $q_f^i - q^i(t)$ .

This simple algorithm must be modified in practice, because the speed is so slow close to the goal that the user is no longer able to feel the authorized direction of motion. We have to define a low threshold. With that threshold, the configuration  $Q(t)$  could exceed the goal position by a known maximum joint error value which is the threshold speed multiplied by the sample time. In practice, the threshold is chosen sufficiently small to induce an error inferior to the resolution of the encoders; that keeps the system controllable.

### 5.2.3 "Trajectory" mode:

A trajectory could be considered as a sequence of positions close to each other to be successively reached in the "position" mode. Because of the control described above, we were afraid of the non-continuous feeling the user would have at each intermediate goal position, and hence that strategy was not used.

Let us consider the trajectory as a continuous curve associated with its corridor. Several methods have been tested to compute the motors' speeds. One consists in computing the projection of the current position  $Q(t)$  on the trajectory. A motion vector  $\vec{V}$  is determined as a linear combination of the normal vector  $\vec{N}$  and of the tangent vector  $\vec{T}$  both computed at this point, depending on the distance of the current position to the trajectory:  $\vec{V} = a(d) \cdot \vec{N} + (1 - a(d)) \cdot \vec{T}$ . When the current position  $Q(t)$  is close to the trajectory, one prefers to follow it ( $a(d) \simeq 0$ ) whilst a gradient displacement is favoured ( $a(d) \simeq 1$ ) when  $Q(t)$  is



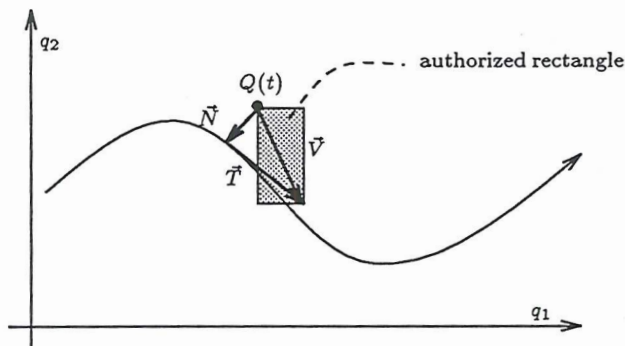


Figure 9: A first algorithm for the trajectory mode.

far from the trajectory. The selected generalized rectangle is then the smallest one defined by the current position  $Q(t)$  and the desired motion vector  $\vec{V}$  (cf. figure 9).

Because the motion can take place anywhere in the generalized rectangle, and has no reason to follow the vector  $\vec{V}$ , this method does not allow easy control of the maximum distance to the trajectory. In practice, the results are not satisfactory, because the feeling of the main trajectory direction is difficult.

A better method consists in searching for the largest rectangular region  $B_i$  which is completely included in the corridor associated with the trajectory and which forces the right direction. This rectangle can easily be computed if we consider a first order approximation of the trajectory as in figure 10. One of its corners is  $Q(t)$ . The motors' speed is deduced from the rectangle dimensions.

The implementation of this method gives satisfactory results: the feeling of the trajectory is far better than that corresponding to the previous approach.

#### 5.2.4 "Region" mode:

In the current implementation of this mode, a point of the surgical tool (the tip for example) is constrained to stay in a predefined region. Let us call that point the control point. To achieve a good tactile feedback, the motion should be as free as possible inside the region and the possible speed in the direction of the border should decrease as the control point moves near to that border.

Let us consider a region defined in configuration space, as shown in figure 11. As in the previous mode, the problem is to find an authorized rectangle around the current position. But in this case, we should avoid favouring the motion in a particular direction.

Obviously, the choice of the authorized rectangle is not unique. Our choice is to maximise the area of the possible position at the next sample time without

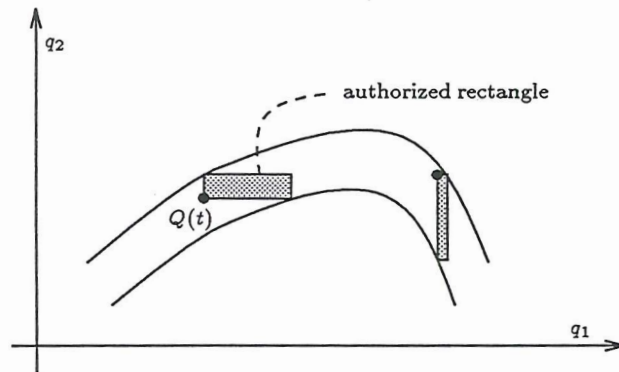


Figure 10: A better algorithm for the trajectory mode.

favouring any particular quadrant. For each of the four quadrants, the rectangle with the maximum area is computed. This rectangle is limited by the quadrant and by the region's border. Then, for each direction of rotation, the smallest side of the rectangles is kept to form the final authorized generalized rectangle as drawn in figure 11. At the border of the region, the rectangle may degenerate since some of the  $B_i$  may vanish. Given the final rectangle size and the sample time, the motor speeds are then computed.

### 5.2.5 Control loop of the system

As shortly presented above, the control loop is the following:  
begin Control loop:

1. Read the arm configuration
2. Until the task is performed:
  - (a) determine the corresponding constraints (generally pre-computed using the algorithms presented in the previous sections),
  - (b) send the orders to the motors,
  - (c) read the arm configuration,
  - (d) return to step 2

end Control loop.

It is clear that some time is necessary to make the constraint fully effective i.e. to obtain the computed speed of the motors; from experiments we determined that the maximum delay to stop one motor at full deceleration is 15ms. Because of that delay, the actual position when the constraints are effective is no longer the

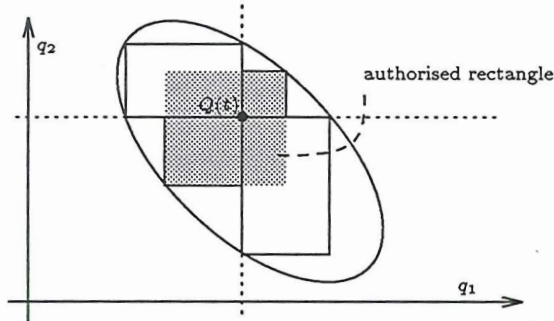


Figure 11: Algorithm for the region mode.

read position and the use of the given algorithms could result in the setting of a constraint no longer compatible with the current position. This could lead to authorize a motion which is no more compatible with the task (cf. figure 12). To overcome that problem, the authorized speeds at a given iteration are computed taking into account the speeds computed at the previous iterations. They are reduced accordingly.

## 6 Experiments:

### 6.1 Experiments

Some examples of tasks realized with the prototype are presented. In spite of the lack of accuracy of the system (see next section), we think that these preliminary experiments fully demonstrate the potentiality of the system.

#### 6.1.1 "Position" mode:

Three trajectories in cartesian space to achieve the same goal position are shown on figure 13.

#### 6.1.2 "Trajectory" mode:

The visual display of a cartesian circular trajectory in joint space is shown in figure 13.a: the planned trajectory and its corridor, the already executed trajectory, and the current rectangle  $\mathcal{B}_i$  are drawn. The resulting cartesian trajectory is shown in figure 13.b.

Figure 15 shows a trajectory recorded in "free" mode and the actual trajectory followed by the PADyC's user.

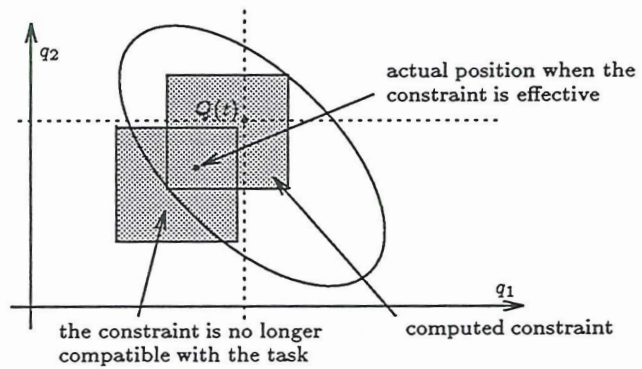


Figure 12: Potential problems arising from response time

Figure 13: "Position" mode example.



Figure 14: "Trajectory" mode example — (a) The visual display and (b) the actual trajectory.

Figure 15: "Trajectory" mode example — (a) Recorded and (b) played back trajectories.

Figure 16: "Region" mode example — The authorized rectangle  $\mathcal{B}$  computed.

### 6.1.3 "Region" mode:

Figure 16 shows the authorized rectangle  $\mathcal{B}$  computed in real-time for a circular cartesian region.

Figure 17 shows samples drawn in a "four-leaf clover" like region.

## 6.2 Mechanical characteristics of the current prototype:

Each link is 250mm long. The axes of the prototype have been designed to resist a force of 3DaN applied at the extremity of the arm. Applying an excessive force would result in damaging the axes. Because of the freewheel technology, no friction can be measured: a force of 0.01N is sufficient to move the arm in an authorized direction. No inertia is detectable.

The major drawback of the current prototype comes an insufficient accuracy at the end-effector. In straight configuration the error may attain a few millimeters. This error can be split in two parts: some backlash and some elastic flexibility. This comes from different sources:

- no 1 flexion in the axes between the freewheel and the link,
- no 2 flexion in the freewheels,
- no 3 backlash coming from the worm screws,
- no 4 hysteresis of the freewheels.

Some experiments have been performed to evaluate the contribution of each of them to the global error. The results are the following:

Figure 17: “Region” mode example — “four-leaf clover” like region.

<i>erreurtype</i>	$F = 1DaN$	$F = 2DaN$
<i>no1</i>	<i>2mm</i>	<i>3.5mm</i>
<i>no2</i>	<i>3mm</i>	<i>6.5mm</i>
<i>no3</i>	<i>2.5mm</i>	<i>2.5mm</i>
<i>no4</i>	<i>2.5mm</i>	<i>2.5mm</i>

These unaccuracies certainly comes from use of off-the-shelf components. We think that a better mechanical design of the arm and the choice of different materials especially for the freewheels would significantly reduce these inaccuracies.

## 7 Discussion:

As mentioned above, the current prototype is not very stiff: in some configurations the displacement of the tool when the joints are blocked may attain some millimeters. This makes it difficult to feel the constrained directions at low speeds. To overcome that problem, so far two strategies have been used. Looking at the visual display significantly improved the realization of tasks especially at low speeds (for example, for following the border of a region). In these low speeds cases, the other very efficient strategy consists in gently shaking the end-effector; the arm “filters” these random small displacements to keep only those permitted by the task. We must add that the visual display used in conjunction with a classical passive arm makes the execution of a task clearly more difficult than it is with our prototype even used with no display. We think that a better mechanical design and manufacturing would make the visual display unnecessary.

The algorithms presented in the previous section have to be generalized in  $nD$  for a  $n$  joints arm; computing generalized rectangles for any shapes of regions turns out to be very difficult. This is why we are currently investigating new algorithms which avoid the need to compute a region in the joint space (cf. [2] for a similar approach in path planning). Concerning the architecture of a six axes robot, many architectures may be selected depending on the type of interventions to be performed (workspace, preferential motions, etc.). In all the cases, the mechanics of the robot has to compensate for gravitational effects. In parallel to this design task, we are studying different solutions using parallel structures and a redundant one in the PADyC framework from an algorithmic and control point of view.

All the experiments were performed in a single type of configuration of the arm (lefty or righty). Particular attention has to be given to the management of singularities by the system. We have also to model the transient states of the system.

## 8 Conclusion:

We have designed a new type of robot, the purpose of which is to cooperate with the surgeon to realize a pre-planned strategy. Such a system, basically passive, induces very limited risks as compared to actuated robots. Giving back to the surgeon a visual and tactile perception of a planned strategy provides him with a valuable physical guidance and a close control of the task execution. Two international patents have been taken on that system. We have built a two degrees of freedom prototype which has proved to be very user-friendly for the execution of different tasks including reaching a position, following a trajectory or staying in a given region. A lot remains to be done before having such a robot fully operational in an operating theater. Nevertheless, we are very optimistic about that issue from the preliminary results we have obtained using that prototype. Unfortunately, as claimed by that system, visual feedback alone is very limited to sense physical objects in space: therefore, using the prototype would be the only proof that would really convince the reader!

## Acknowledgements :

This research has been partly supported by the Université Joseph Fourier of Grenoble by a grant (Fonds d'Incitation au Transfert de Technologies). We want to thank Dr Stéphane Lavallée for his strong involvement in the very principles of the PADyC and Mr Leszek Lisowski for his design of the control rack of the system.



## References

- [1] - L. Adams, A. Knepper, W. Krybus, D. Meyer-Ebrecht, G. Pfeifer, R. Ruger, Witte: "Orientation aid for head and neck surgeons", *Innov. Tech. Biol. Med.*, Vol 13, no 4, 1992
- [2] - J. Barraquand, J.C.Latombe: "Robot motion planning: a distributed representation approach", Technical Report, STAN-CS-89-1257, Stanford University, May 1989
- [3] - B.L. Davies, R.D. Hibberd, A. Timoney, and J.E.A. Wickham: "A surgeon robot for prostatectomies", *Proc. of 2nd Int. Conference on Robotics in Medicine*, Newcastle, UK, 1989
- [4] - B.L. Davies, S.C. Ho, R.D. Hibberd: "The use of force control in robot assisted knee surgery", *Proceedings of the 1st International Symposium on Medical Robotics and Computer Assisted Surgery*, pp258-262, Pittsburgh, 1994
- [5] - P. Kazanzides, J. Zuhars, B. Mittelsstadt, R.H. Taylor: "Force sensing and control for a surgical robot", *IEEE Conference on Robotics and Automation*, pp 612-617, Nice, 1992
- [6] - J.C. Latombe: "Robot motion planning", *The Kluwer International Series in Engineering and Computer Science*, Kluwer Academic Publishers, 1991
- [7] - S. Lavallée, J. Troccaz, L. Gaborit, A.L. Benabid and D. Hoffmann: "Image guided operating robot: a clinical application in stereotactic neurosurgery", *IEEE Conf. on Robotics and Automation*, Nice, May 1992
- [8] - B.D. Mittelstadt, P. Kazanzides, J. Zuhars, B. Williamson, P. Cain, F. Smith, W.L. Bargar: "The evolution of a surgical robot from prototype to human clinical use", *Proceedings of the 1st International Symposium on Medical Robotics and Computer Assisted Surgery*, pp36-41, Pittsburgh, 1994
- [9] - R. Mosges, G. Schlondorff, L. Klimek, D. Meyer-Ebrecht, W. Krybus, L. Adams: "Computer assisted surgery, an innovative surgical technique in clinical routine", *Computer Assisted Radiology, CAR'89*, pp 413-415, Berlin, June 1989
- [10] - R.H. Taylor et al.: "A model-based optimal planning and execution system with active sensing and passive manipulation for augmentation of human-precision in computer-integrated surgery", *Proc. 1991 Int. Symposium on Experimental Robotics*, Toulouse, June 1991
- [11] - R.H. Taylor et al.: "Augmentation of human precision in computer integrated surgery", *Innov. Tech. Biol. Med.*, Vol 13, no 4, 1992

- [12] - J. Troccaz, S. Lavallée, E. Hellion: "PADyC: A Passive Arm with Dynamic Constraints", Proc. of ICAR'93, pp361-366, Tokyo, 1993 .

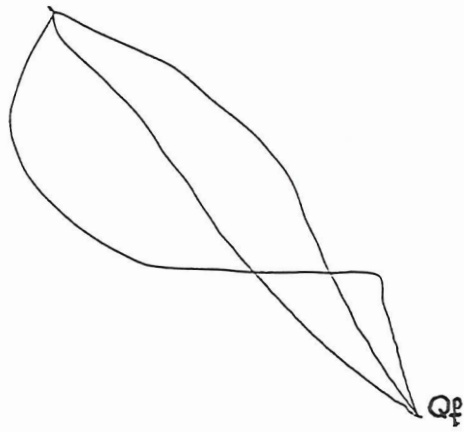


Fig. 12. "Position" mode example.

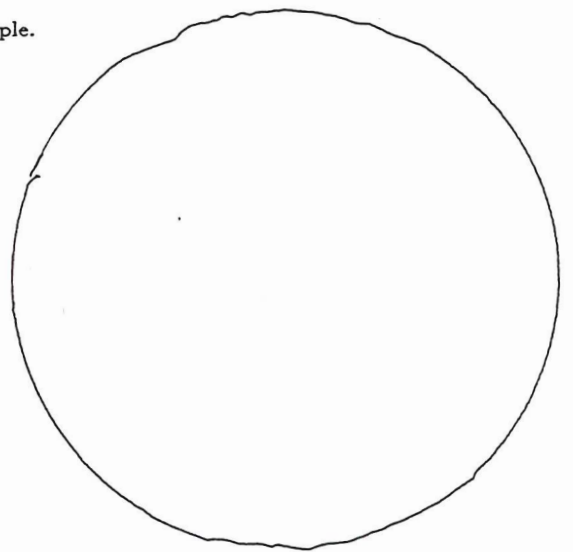
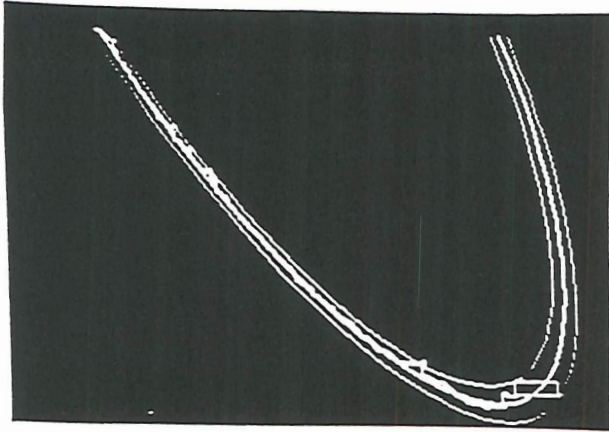


Fig. 13. "Trajectory" mode example — (a) The visual display and (b) the actual trajectory.



Fig. 14. "Trajectory" mode example — (a) Recorded and (b) played back trajectories.

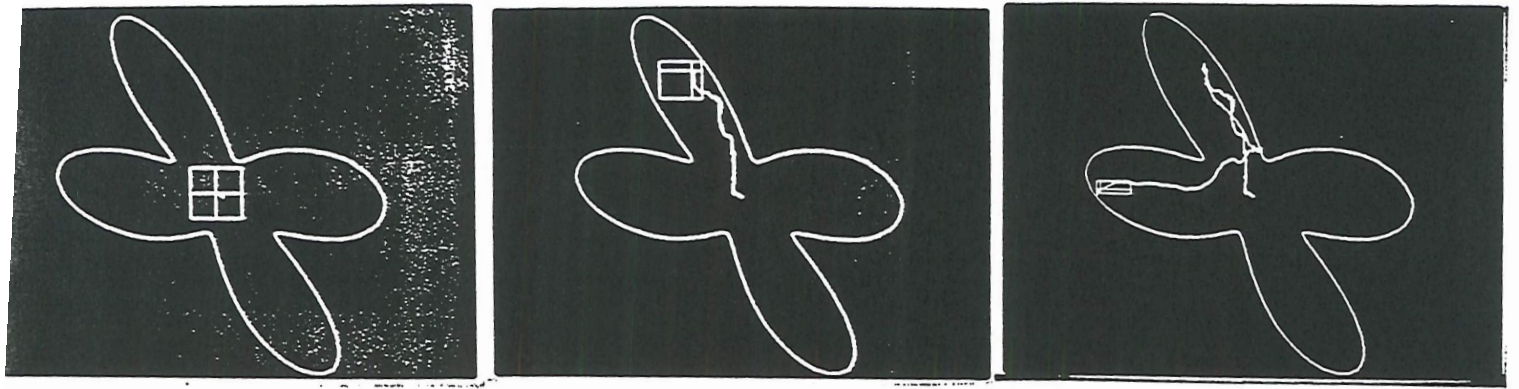


Fig. 15. "Region" mode example — The authorized rectangle  $B$  computed.

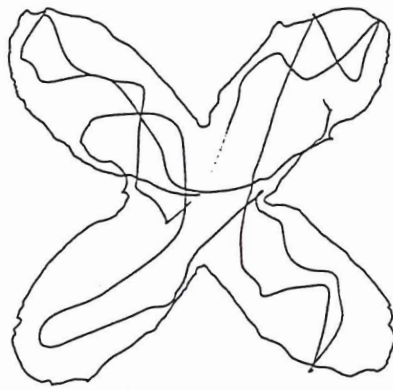


Fig. 16. "Region" mode example — "four-leaf clover" like region.

is drawn in a "four-leaf clover" like region.