HUMAN-INTERACTIVE SIMULATION AND DISPLAY OF AN UNDERWATER REMOTELY OPERATED VEHICLE

Thierry Royer

MITSG 85-29TN
November, 1985

MIT Sea Grant College Program
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Grant No.: NA84AA-D-00046
Project No.: R/V-7

This Sea Grant report is being published as part of the MIT Program's Technical Note series. The research was conducted by Thierry Royer in fulfillment of a Master of Science degree in MIT's Department of Ocean Engineering. The thesis supervisor was Thomas B. Sheridan, Professor of Mechanical Engineering and Applied Psychology.

Copies of this report may be ordered from

Sea Grant Information Center
MIT Sea Grant College Program
Room E38-320
Massachusetts Institute of Technology
77 Massachusetts Ave.
Cambridge, MA 02139

cost: $5.00

ABSTRACT

A simulator for an undersea ROV (remotely operated vehicle), if accurate enough, is an important alternative to the use of a real vehicle for the trainaing of human operators and the testing of new control methods.

The goal of this thesis is to provide a framework for future developments in underwater vehicle simulation. The simulator, implemented on a PDP 11/34, is comprised of four modules working in parallel: a dynamic model of the vehicle, a static model for the shape and tension of the tether, a graphics display, and an environment emulator. The graphics program represents a simplified environment with hidden face removal, with at least two images/second. The environment is composed of an offshore structure (4 columns and 4 transverse bars), a pipe-line, and a wellhead. Tne environment emulator detects and signals the collision with the elements of the environment.

The modular conception of this program allows easy change within the modules, and inexpensive implementation on a multi microcomputer system.

TABLE OF CONTENTS

LIST OF FIGURES.

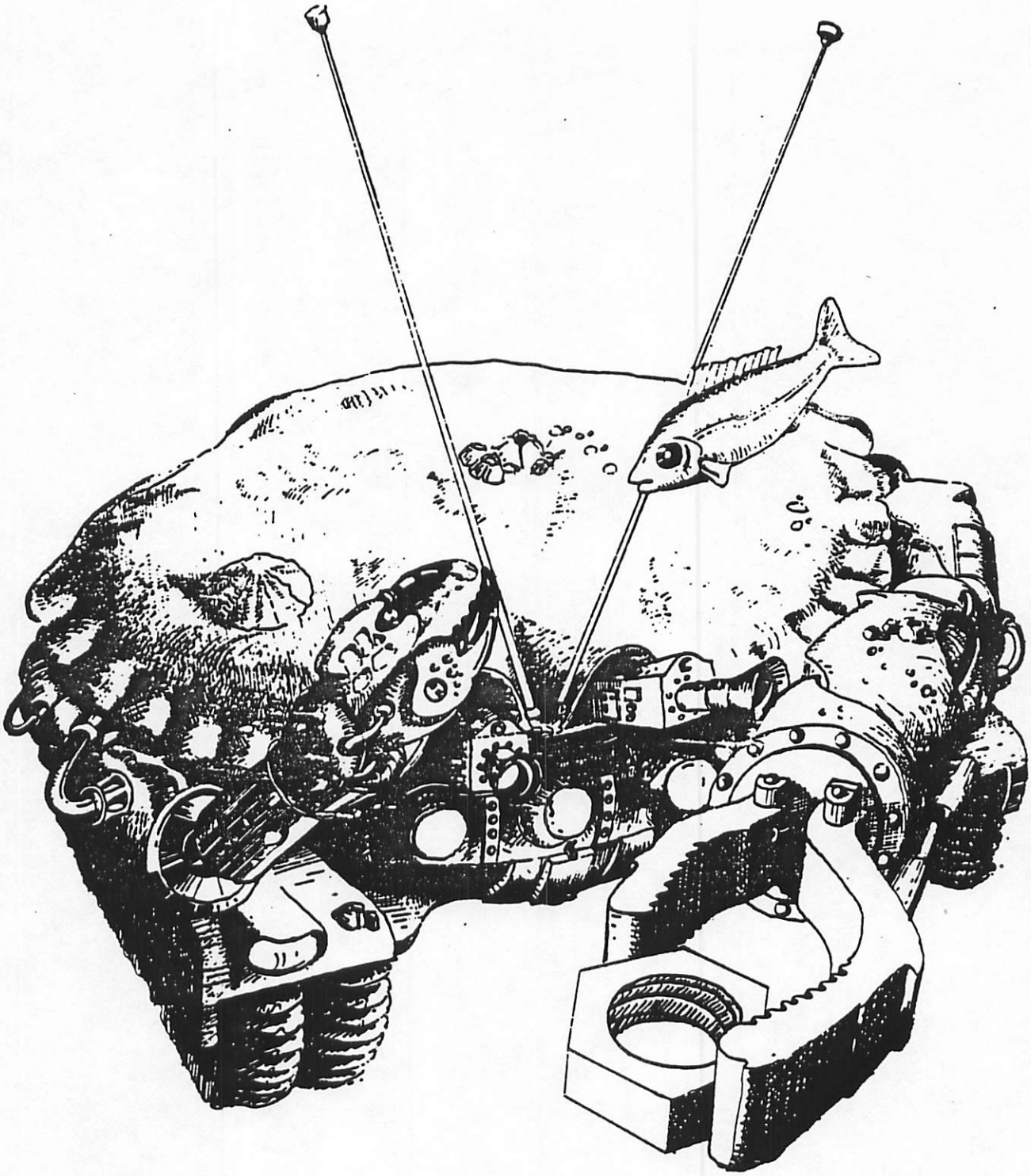


figure 1: Artist view of the new generation of ROV designed from simulators

CHAPTER 1

INTRODUCTION

The use of underwater remotely operated vehicles (ROV) in place of divers or manned submarines is becoming more widespread. The divers, though much more skilled than a remotely operated set - vehicle and manipulators-, tend to be replaced for economic and safety reasons, as well as for depth limitations. The high costs and long delays for manned submarine interventions preclude their use whenever a close human observer is not absolutely necessary.

For these reasons, the number of remotely operated vehicles is steadily increasing, and their specifications have to be improved in order to achieve the performance previously attained by divers.

As a consequence, ROV simulators have become an important alternative to the use of a real vehicle for the training of human operators and the testing of new engineering designs. An English consultant company has actually launched a survey among ROV users and builders to define the potential interest (April 1985).

Because the number of ROV's is increasing, new pilots will have to be trained (by 1990 an estimated 700 will be needed). This represents a considerable undertaking given that it takes about 100 hours for a pilot

to become proficient. Presently some companies are beginning to offer prepared seaspace for pilot training with real vehicles, independently of the weather or the sea conditions (e.g. NUTEC in a Norwegian fiord, or International Underwater Contractor in Houston, references 4, 14).

New control methods and new operating ideas such as 'supervisory control' proposed by Sheridan and Ferrell (ref. 11), will have to be developed, in order to enhance the capabilities of the vehicle and facilitate the task of the pilots.

The availability of a new generation of fast micro-computers will allow us to create reasonably accurate and economically efficient interactive simulators compared to the current operating cost of an ROV, utilized for teaching and testing.

The goal of my thesis is to provide a framework for future development in the interactive simulation of an underwater vehicle. The first utilizations should be primarily the design by another team of a heading and depth controller using sliding control, to be installed on the Man-Machine Systems Lab ROV, and secondly, a sensitization to the piloting problems for the team working on this ROV.

The simulator is implemented on a PDP 11/34, and is designed as 4 modules running in parallel: 1) a dynamic model of the vehicle, 2) a static model for the shape and tension of the tether (if needed), 3) a graphics display, and 4) an environment emulator. This modularity enables easy modification, test, and improvement of the different tasks.

The first module generates the 6 degree-of-freedom dynamics for real time use. Appropriate differential equations are currently available in the literature in general form.

For a tethered vehicle, an optional module can be used to compute in real time an estimate of the tether reactions depending on the positions of the ROV relative to the surface ship. The model for the umbilical is based on an iterative computation of the flow drag on rigid cylinder elements.

The original graphics program is able to represent a simplified environment with hidden face removal, perspective effect and shade, with a minimum of 2 images per second. The display is generated on a simple color screen without any specific hardware functions except polygon filling. The simulator world is composed of an offshore structure (4 columns and 4 transverse bars), a pipe-line, and a well head as a cube.

The environment emulator takes care of the interaction between the vehicle and the world by detecting and signalling collisions with the environment.

These 4 modules are orchestrated by a main routine. This routine can simulate the different problems inherent in information transmission between the ROV and the surface ship, such as delays for untethered vehicles. The display of the information can be modified to study its impact on piloting.

Finally, a future implementation of this kind of simulator on a multi-microcomputer system is suggested. Such an application would fit perfectly with the modular concept of this program. The graphic display presented proves the feasibility of such an implementation with an inexpensive graphic unit.

CHAPTER 2

A DYNAMIC MODEL FOR THE ROV

2.1 Introduction.

An essential part of a vehicle simulator is its dynamic simulation. The precision of the numerical model will result in a simulation reflecting closely the real motion of the vehicle. This precision is achieved when the forces utilized in the model represent precisely enough the forces really involved. The first part of this chapter summarizes both the method used to obtain the basic dynamic equations of a body moving with 6 degree-of-freedom and the various assumptions which simplify the final equations. Such equations are currently available in the literature (ref. 1). The second part gets into the details more specific to an ROV, and develops and explains the different choices and simplifications in the representation of the forces.

2.2 The 6 degree-of-freedom dynamics for a solid body.

The center of gravity G of a body is usually chosen as the reference point of its motion in a fixed system. In the case of a body underwater, it is more judicious to treat the dynamic equations by selecting a fixed point P in the body which might be different from G. Usually this point is taken as the center of the hydrodynamic forces application, if one can be defined. Hence hydrodynamic moments can be fairly simplified.

This point is in the same vertical plane of symmetry of the vehicle, as the center of gravity and the center of buoyancy.

The position of the ROV is thus represented by the coordinates X_P, Y_P, Z_P in a galilean system $(O,\hat{I},\hat{J},\hat{K})$. The axis $(O,\hat{Z})$ points downward (fig 2.1).

A local system $(P,\hat{i},\hat{j},\hat{k})$ linked to the body is defined from the symmetries of the vehicle, $(P,\hat{i})$ points forward, $(P,\hat{k})$ points downward.

The position and the orientation of this system, from the previous one, is determined by X_P, Y_P, Z_P and three angles ψ,ϕ,θ, respectively yaw, pitch, and roll (fig. 2.2).

The instantaneous speed and rotation vectors for the point P in the local system are represented as : $\hat{U} = u.\hat{i} + v.\hat{j} + w.\hat{k}$ and $\hat{\Omega} = p.\hat{i} + q.\hat{j} + r.\hat{k}$.

The equations of the dynamic give:

$$\Sigma(\text{external forces on body}) = m.\frac{d}{dt}(\hat{U} + \hat{\Omega}\times\hat{R}_G)$$

where $\hat{R}_G$ is the vector from P to G (center of gravity) $\hat{R}_G = x_G\hat{i} + z_G\hat{k}$ and m is the mass of the body. (y_G is supposed to be null).

Assuming that the axes $(\hat{I},\hat{j},\hat{k})$ are parallel to the inertia axes (this is, to some extent, usually the case for an ROV), we get:

$$\Sigma(\text{moments in G}) = \frac{d}{dt}(I'_x p\hat{i} + I'_y q\hat{j} + I'_z r\hat{k})$$

with $I'_x = I_x - m.z_G^2$ (I_x, I_y, I_z are the

$I'_y = I_y - m(x_G^2 + z_G^2)$ inertia relative to P)

$I'_z = I_z - m.x_G^2$ from Huyghens theorem

and $\Sigma(\text{moments in P}) = \Sigma(\text{moments in G}) + m\hat{R}_G \times \frac{d}{dt}(\hat{U}+\hat{\Omega}\times\hat{R}_G)$

using the notation $\frac{d}{dt}u = \dot{u}$ for (u,v,w,p,q,r), the final equations are:

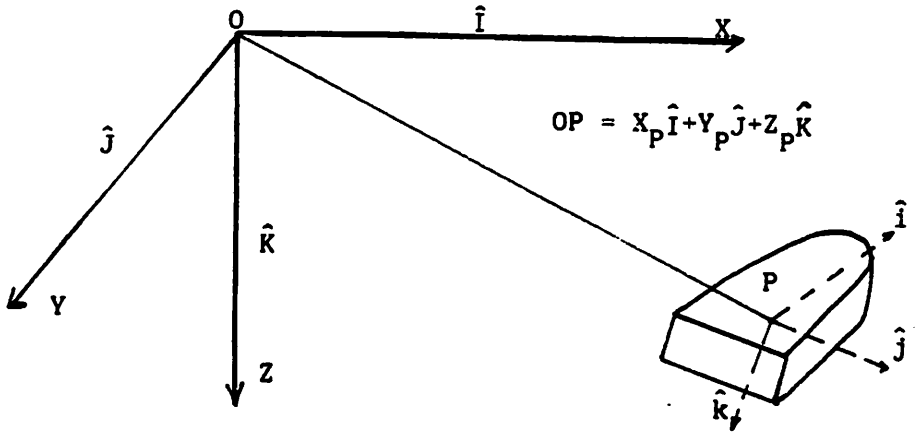


figure 2.1: The fixed system of reference.

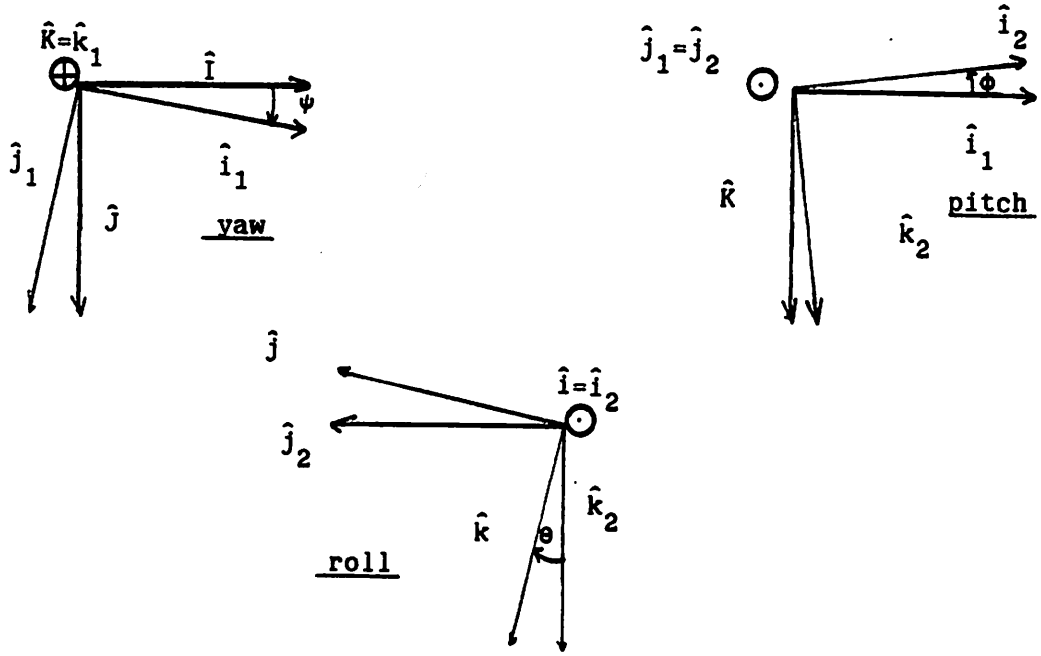


figure 2.2: Rotations to obtain the local system.

$$\begin{cases} F_x = m.[(\dot{u} - vr + wq) - x_G(q^2+r^2) + z_G(pr+\dot{q})] \\ F_y = m.[(\dot{v} + ur - wp) + x_G(pq+\dot{r}) + z_G(qr-\dot{p})] \\ F_z = m.[(\dot{w} - uq + vp) + x_G(pr-\dot{q}) - z_G(p^2+q^2)] \end{cases}$$

$$\begin{cases} M_x = I_x\dot{p} + (I_z-I_y)qr + m[- z_G(\dot{v}+ur-wp) - x_Gz_G(pq+\dot{v})] \\ M_y = I_y\dot{q} + (I_x-I_z)rp + m[z_G(\dot{u}+qw-rv) - x_G(\dot{w}+pv-qu) + x_Gz_G(p^2-r^2)] \\ M_z = I_z\dot{r} + (I_y-I_x)pq + m[x_G(\dot{v}+ru-pw) + x_Gz_G(rq-\dot{p})] \end{cases}$$

F_x, F_y, F_z are the sum of the external forces on the body, expressed in the system $(\hat{i},\hat{j},\hat{k})$,

and M_x, M_y, M_z are the sum of the moments around P in the same system.

The last set of equations is relative to the moment given by a development of the additional term: $R_G \times \sum$(forces on the body). This development is useful to elaborate a control strategy since the force on the body depend on the thrusters propulsion that we can control. However, for a simulation purpose, after computing the external forces it is valid and much simpler to use them directly in the equations for the moment.

Therefore we have:

$$\begin{cases} M_x + z_G.F_y = I'_x\dot{p} - I'_yqr + I'_z\dot{q}r \\ M_y + x_G.F_z - z_G.F_x = I'_xpr + I'_y\dot{q} - I'_zpr \\ M_z - x_G.F_y = -I'_xpq + I'_ypq + I'_z\dot{r} \end{cases}$$

(for more details see appendix A.1)

2.3 Axis transformations.

At this point some transformations are needed in order to go from one

system to the other.

(using $c\psi$ as $\cos(\psi)$ and $s\psi$ as $\sin(\psi)$, respectively for θ,ϕ)

$T(\psi,\phi,\theta)$ = matrix to pass form $(\hat{I},\hat{J},\hat{K})$ to $(\hat{i},\hat{j},\hat{k})$ is expressed as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c\psi.c\phi & s\psi.c\phi & -s\phi \\ c\psi.s\phi.s\theta-s\psi.c\theta & s\psi.s\phi.s\theta+c\psi.c\theta & c\phi.s\theta \\ c\psi.s\phi.c\theta+s\psi.s\theta & s\psi.c\theta.s\phi-c\psi.s\theta & c\phi.c\theta \end{bmatrix} . \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

This matrix and its inverse (transpose[$T(\psi,\phi,\theta)$]) are used to transform some of the forces in the local system, and compute the motion in the system $(\hat{I},\hat{J},\hat{K})$.

We have also as shown in figure 2.2: $\hat{\Omega} = \dot{\psi}\hat{K} + \dot{\phi}\hat{j}_1 + \dot{\theta}\hat{i}_2$.

Hence: $p = -\dot{\psi}.s\phi + \dot{\theta}$

$q = \dot{\psi}.c\phi.s\theta + \dot{\phi}.c\theta$

$r = \dot{\psi}.c\phi.c\theta - \dot{\phi}.s\theta$

Or: $\dot{\psi} = (q.s\theta + r.c\phi)/c\phi$

$\dot{\phi} = q.c\theta - r.s\theta$

$\dot{\theta} = p + \tan(\phi).(r.c\theta + q.s\theta)$

(see appendix A.2 for details)

2.4 Forces involved in the dynamic of a ROV.

The next step in the description of a model for a vehicle is to detail the external forces acting on the ROV. As explained in the introduction of this chapter, the accuracy of the model will be very dependent on their precise formulations. Several representations for underwater vehicles

exist in the literature (ref. 6, 7). These models are very complete and thus very expensive in time of computation for real time purposes. They also need a complete study in a towing tank to obtain all the coefficients utilized. They are open to question when these coefficients are not available or are inaccurate.

For these reasons, I chose to simplify the representation of the forces and only consider those we could fairly estimate or which have a measurable effect on the dynamics response of the ROV. Some simplifications can also be done in the representation, since for the particular case of an ROV no variable surface such as rudder exists. Hence the forces considered are :

a) the weight and the buoyancy,

b) "added mass" or the inertial force due to the acceleration of the fluid,

c) drag forces,

d) tether reactions,

e) thrusters.

The simplification made will be explained in each case.

2.4.a Weight and buoyancy.

Expressed in the local system,

the gravity force is: $\hat{P}_{gra} = mg.[\ -s\phi.\hat{i} + s\theta.c\phi.\hat{j} + c\theta.\hat{k}\ ]$

the buoyancy is : $\hat{F}_{buoy} = -\rho.Vol.[\ -s\phi.\hat{i} + s\theta.c\phi.\hat{j} + c\theta.\hat{k}\ ]$

ρ is the water density; Vol is the vehicle displacement.

For a neutral buoyancy vehicle $mg = \rho.Vol$ and $\hat{F}_{gra}+\hat{F}_{buoy} = \hat{0}$.

However, since the cable tends to pull the ROV toward the surface, the vehicle can be lested (mg > ρ.Vol) to compensate for this additional force, and thus spare the vertical thrusters.

Knowing the respective position of the center of gravity G and the center of buoyancy B from P ($\hat{R}_B = x_B\hat{i} + z_B\hat{k}$), we have for the moments:

$$\hat{M}_{gra} = mg.\{ -z_G.s\theta.c\phi.\hat{i} - (z_G.s\phi + x_G.c\theta)\hat{j} + x_G.s\theta.c\phi.\hat{k} \}$$

$$\hat{M}_{buoy} = -\rho.Vol.\{ -z_B.s\theta.c\phi\hat{i} - (z_B.s\phi + x_B.c\theta)\hat{j} + x_B.s\theta.c\phi.\hat{k} \}$$

2.4.b "Added mass".

The acceleration of the fluid by the acceleration of the vehicle creates an extra force proportional and opposite to the the motion directions.

Thus we have to consider the force:

$$[-m_{11}.\dot{u} ; -m_{22}.\dot{v} ; -m_{33}.\dot{w}]$$

and the moment : $[-I_{44}.\dot{p} ; -I_{55}.\dot{q} ; -I_{66}.\dot{r}]$

2.4.c The drag.

The drag will be dependent on the relative motion of the vehicle in the water. This motion can be different from the set (u,v,w,p,q,r) in the presence of current. Assuming the current velocity as VCx,VCy,VCz expressed in the fixed system ($\hat{I},\hat{J},\hat{K}$) the relative speed is then :

$$\begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} - T(\psi,\phi,\theta). \begin{bmatrix} VCx \\ VCy \\ VCz \end{bmatrix}$$

The drag is usually expressed as a force proportional to the square of the corresponding motion which gives a good approximation for speed important enough. For a low speed however, the drag effect is underestimated. By adding a term proportional to the motion, we can take care of the low speed drag (ref. 2).

Thus the force drag can be expressed as:

$$\begin{aligned}\hat{F}_{drag} = &(-C^1_{uu}.|u_r|.u_r - C^2_{uu}.u_r)\hat{i} \\ &+ (-C^1_{vv}.|v_r|.v_r - C^2_{vv}.v_r)\hat{j} \\ &+ (-C^1_{ww}.|w_r|.w_r - C^2_{ww}.w_r)\hat{k}\end{aligned}$$

This expression is a simplification. I did not consider any lift effect. For instance, a motion in $\hat{i}$ might create a force along $\hat{k}$, or a motion in $\hat{j}$ a force along $\hat{i}$. A rotation along any axis might also create a force. These effects have been neglected in the "drag" formulation. The drags have been supposed also to be independent. For example, the drag due to a motion in both $\hat{i}$ and $\hat{j}$ is supposed to be equal to the sum of the drag due to the independent motions. These lift effects are really negligible compared to other effects, such as tether or thrusters, since the shape of the vehicles modelled in this simulator are not especially designed to produce any portance.

If we assume that there exists a point in the ROV where moments due to (u,v,w) are null for any conditions, then we can select this point as the reference point P of the vehicle. This assumption is realistic, since the point of application of the hydrodynamic forces will not vary much because of the limited range of speed. Thus considering that particular point, we obtain for the drag moment, neglecting also the hydrodynamic coupling effects:

$$\hat{M}_{drag} = (-C^1_{pp}.|p|.p - C^2_{pp}.p)\hat{i} + (-C^1_{qq}.|q|.q - C^2_{qq}.q)\hat{j} + (-C^1_{rr}.|r|.r - C^2_{rr}.r)\hat{k}$$

(Note: - These simplifications do not suppress the cross-coupling effect between the motion.

- Also the position of the point P has to be found in the vehicle.)

2.4.d Tether reactions.

An approach to obtain the static force due to the tether in a given current is presented in Chapter 3. With this reaction in the fixed system, the calculation of the forces and moments in the local system is only a matter of transformation and cross product.

If $\hat{F}_{Tether} = F_{XT}\hat{I} + F_{YT}\hat{J} + F_{ZT}\hat{K}$ then in the local system we have:

$$[\ F_{xT}\ \ F_{yT}\ \ F_{zT}\ ]^T = T(\psi,\phi,\theta)\ .\ [\ F_{XT}\ \ F_{YT}\ \ F_{ZT}\ ]^T$$

With a tether fixed at T where $\hat{R}_T = x_T\hat{i} + z_T\hat{k}$ (the tether is fixed on the symmetry plane) the moment is :

$$\hat{M}_{Tether} = -z_T.F_{yT}\hat{i} + (z_T.F_{xT} - x_T.F_{zT})\hat{j} + x_T.F_{yT}\hat{k}$$

2.4.e Thruster actions.

The associated forces will obviously depend on their position, orientation, and power. In the configuration of the ROV modelled in this simulator (see Chapter 5), five thrusters are used and the force and moment are:

$$\hat{P}_{Thrusters} = (P_1+P_2)\hat{i} + P_3\hat{j} + (P_5+P_6)\hat{k}$$

$$\hat{M}_{Thrusters} = [\ -z_3.P_3 + (P_6-P_5)y_{56})]\hat{i} + [\ (P_2+P_1)z_{12} - (P_5+P_6)x_{56})]\hat{j} + [\ (P_1-P_2)y_{12} + P_3x_3]\hat{k}$$

P_i is the propulsion due to the thruster i, the x_{ij}, y_{kl}, z_{mn} are the useful coordinates of the thruster positions to get the moment (fig 2.3).

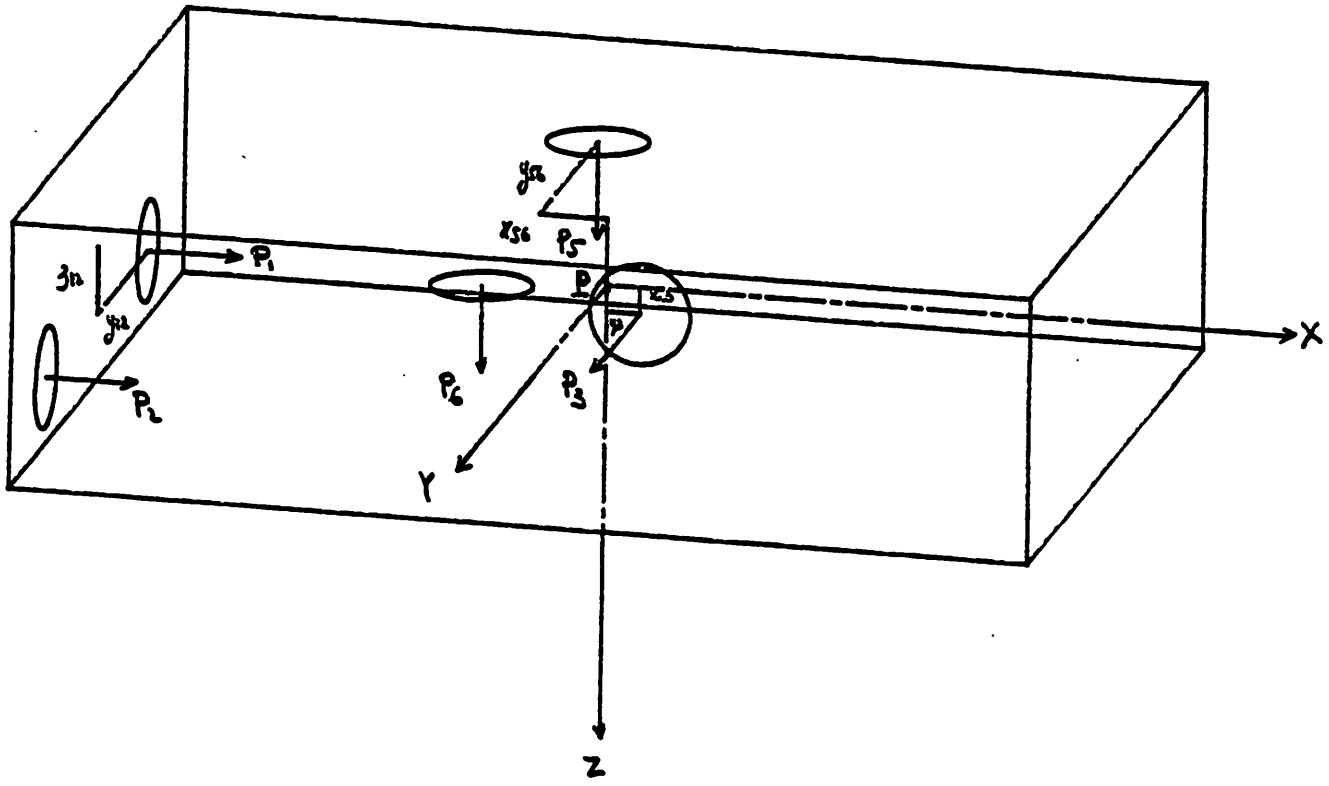


figure 2.3: Position of the thrusters in the ROV modelled.

The propulsion of a thruster is not exactly proportional to the propeller rotation N controlled by the pilot but has dynamics such as:

$$\dot{P}(t) = -\frac{P(t)}{\tau} + \alpha.|N|.N$$

τ is a time constant $\approx$ 0.5 second

α is a fixed coefficient.

As we can see the time constant is fairly small, allowing us to simplify the thruster dynamics to be proportional to the pilot command.

2.4.f General remark on the forces representations.

The purpose of the model in this thesis is to produce a reasonable estimate of the dynamics of an ROV. Underwater vehicle dynamics present usually more than 40 coefficients for modelling different hydrodynamic forces. As seen in the previous paragraphs, we hardly arrive at this number in the model presented. Several reasons might justify not worrying about the terms simplified.

As a first step, the coefficients utilized for the ROV modelled have been guessed from an extrapolation of the other ROV's characteristics, since no experimental data are available at the moment. Thus, some errors surely exist for these estimate. As we will see in the next chapter, forces and moments due to the tether are important and cannot be computed exactly, which is another source of error. Finally, the dynamics of a small underwater vehicle, such as an ROV, is subject to external unsteady fluid motion such as turbulences, that the more complete models can not take into account. These effects might surpass the precision given by some sophisticated simulations.

2.5 Integration consideration. The dynamic module of the simulator.

To obtain the motion of the vehicle the procedure is straightforward:

- The forces and moments are evaluated.

- for a (u,v,w,p,q,r) we have to resolve

$$(1)\quad \begin{cases} (m+m_{11})\dot{u} + m.z_G.\dot{q} = F_x + m.vr - m.wq + m.x_G(q^2+r^2) - m.z_G.pr \\ (m+m_{22})\dot{v} - m.z_G.\dot{p} + m.x_G.\dot{r} = F_y - m.ur + m.wp - m.x_G.pq - m.z_G.qr \\ (m+m_{33})\dot{w} - m.x_G.\dot{q} = F_z + m.uq - m.vp - m.x_G.pr + m.z_G(p^2+q^2) \end{cases}$$

and

$$(2)\quad \begin{cases} \dot{p} = [\, M_x + z_G.F_y + (I'_y - I'_z)qr]/(I'_x+I_{44}) \\ \dot{q} = [M_y + x_G.F_z - z_G.F_x + (I'_z - I'_x)pr]/(I'_y+I_{55}) \\ \dot{r} = [\, M_z - x_G.F_y + (I'_x - I'_y)pq]/(I'_z+I_{66}) \end{cases}$$

The forces and the moment are considered without the "added masses" already in the equations

To develop a control algorithm, we would have to develop the forces and moments functions of (u,v,w,p,q,r,X,Y,Z,θ,ϕ) to obtain a vectorial differential equation:

$$[\text{mass\_matrix}]*\dot{\underline{S}} = f_1(\underline{S}) \quad \text{development of the dynamic equations}$$
$$+ f_2(\underline{S}) \quad \text{Hydrodynamic forces without "added masses"}$$
$$+ g(P_1,P_2,P_3,P_4,P_5) \quad \text{Thrusters}$$
$$+ h(X,Y,Z) \quad \text{Cable}$$
$$+ k(\phi,\theta) \quad \text{hydrostatic forces}$$

with $\underline{S} = [\, u\ v\ w\ p\ q\ r\ ]^T$

and a mass matrix :

$$\begin{bmatrix} (m+m_{11}) & 0 & 0 & 0 & m.z_G & 0 \\ 0 & (m+m_{22}) & 0 & -m.z_G & 0 & m.x_G \\ 0 & 0 & (m+m_{33}) & 0 & -m.x_G & 0 \\ 0 & m.(z_G+z_G.x_G) & 0 & (I_x+I_{44}) & 0 & 0 \\ m.z_G & 0 & -m.x_G & 0 & (I_y+I_{55}) & 0 \\ 0 & m.x_G & 0 & -m.x_G.z_G & 0 & (I_z+I_{66}) \end{bmatrix}$$

The equations (1) & (2) can be integrated by several methods like Runge-Kutta or Newton.

Runge-Kutta order 4 is usually employed since it is more precise that Newton. It is also more time consuming : 4 evaluations are needed instead of one. Because the PDP 11/34 (see Chapter 5) served also for the other modules, and because I needed real-time dynamics, I chose the Newton method:

$$\underline{S}(t+\Delta t) = \underline{S}(t) + \dot{\underline{S}}(t).\Delta t.$$

The theoretical loss of precision with this method is compensated by a smaller step of integration, since the loop time is less.

The listing of this module is presented in Appendix B.

CHAPTER 3

STATIC FORCE DUE TO A TETHER IN A CURRENT.

3.1 Statement of the problem.

The research on untehered vehicles is nowadays very active, but only a few ROV's are operating without an umbilical cord. A tether, at the moment, is still the best manner to transmit underwater information (especially video) between the ROV and the surface ship and to provide power for the propulsion and the diverse tools.

While the tether is very convenient, the drag due to its presence is a serious drawback. Its reactions are important enough, so that some practical research has been conducted (ref. 5, 12) or soon will be conducted, e.g. Pr Triantafyllou with the Sea Grant College Program, in order to evaluate the forces involved and take them into account to estimate more precisely the propulsion needs. These forces are both static and dynamic.

The forces acting on a cable can be modelled for static or dynamic computations (ref. 1, 3, 8, 13) and the set of differential equations can be integrated from a set of initial conditions, such as the tension and the angle of the cable at one of its extremity. However, no formulation exists to obtain the tension and the angle of the extremities of a tether, given only the relative positions of its extremities and its length. From a set of differential equations, proposed by Antoine BLIEK (M.I.T. Ocean Eng. Dept.), I found an algorithm to give, in real time, the extremity tension, from the relative position of the extremities.

3.2 The differential equations for a cable.

The reference point is (O,x,y) (fig. 3.1). The starting point is A, and the angle α is negative.

At a point s on the cable: $Q(\alpha)$, $P(\alpha)$ are respectively the normal and tangential forces acting on the cable at s. These forces are due to: gravity, buoyancy, and current drag. At the equilibrium we have:

$$dT(s) = - P(\alpha)$$

$$T.d\alpha = - Q(\alpha)$$ where T(s) is the tension at s (fig. 3.2).

The gravity is $-\lambda.g.ds.\vec{j}$ with λ linear mass of the tether, and the buoyancy can be approximated by $\nu.g.ds.\vec{j}$ with ν linear volume of the tether. This is an approximation, since the previous force would be the buoyancy of an element 'ds' separate from the others. The hydrostatic pressure is also integrated with the element extremities (fig. 3.3).

The effect buoyancy-gravity is thus : $-\omega.\vec{j}$ with $\omega = (\lambda - \nu).g$

The tangential hydrodynamic force can be expressed as:

$$\frac{1}{2}\rho.C_f.D.|V_t|.V_t.ds$$

Where C_f is a friction coefficient $\approx 0.05\pi$

D is the cable diameter.

V_t is the tangential current speed, relative to the tether.

ρ is volumic mass of the fluid.

The normal hydrodynamic force is:

$$\frac{1}{2}\rho.C_n.D.|V_n|.V_n.ds$$

Where C_n is a normal drag coefficient ≈ 1.2 for speed current < 2m/s

V_n is the normal current speed, relative to the tether.

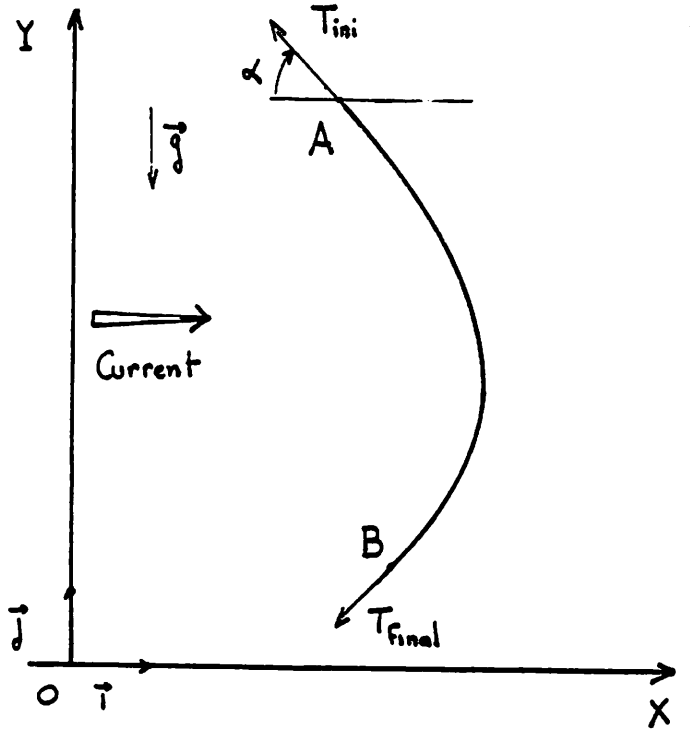


fig. 3.1: Axes system for the cable

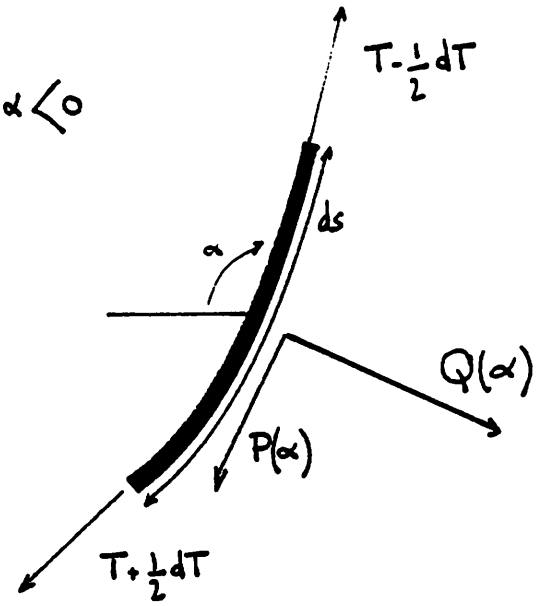


fig. 3.2: Forces on an element ds

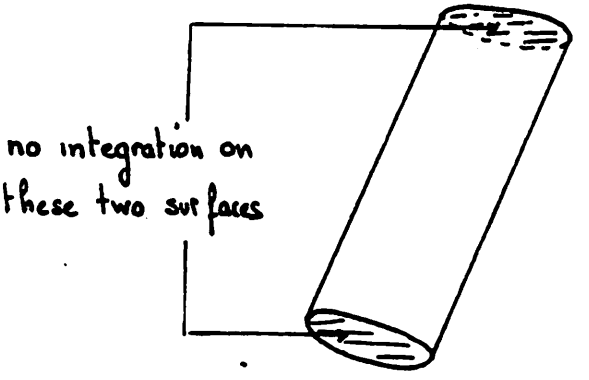


figure 3.3: Hydrostatic pressure on an element ds of the cable.

The differential equations become:

$$\frac{dT(s)}{ds} = \omega.\sin(\alpha) - \frac{1}{2}.\rho.C_f.D.|V_t|.V_t$$

$$T.\frac{d\alpha(s)}{ds} = \omega.\cos(\alpha) - \frac{1}{2}.\rho.C_n.D.|V_n|.V_n$$

and the deflection in x and y are $\frac{dx}{ds} = (1+e).\cos(\alpha)$ and $\frac{dy}{ds} = (1+e).\sin(\alpha)$, with $e = T(s)/E_c$ and E_c the elasticity module of the cable.

The speed current is usually horizontal $\hat{V}=V(y).\hat{i}$ and so: $V_t=V.\cos(\alpha)$ and $V_n = - V.\sin(\alpha)$

The integration of these equations will give from :

a) the position of the first extremity A, and

b) the orientation and the magnitude of the tension at that extremity,

the corresponding value for the other extremity B.

3.3 A simple 3 dimensional case.

These differential equations can be used after some modifications in 3 dimensions. The equations are however, more complicated.

When the cable is neutrally buoyant and when the speed varies only along one direction, for example $\hat{V} = V(y).\hat{i}$, then the problem becomes bidimensional. The plane to consider is the plane passing through the first extremity and defined by the tension vector at that point, and by the current vector (fig 3.4).

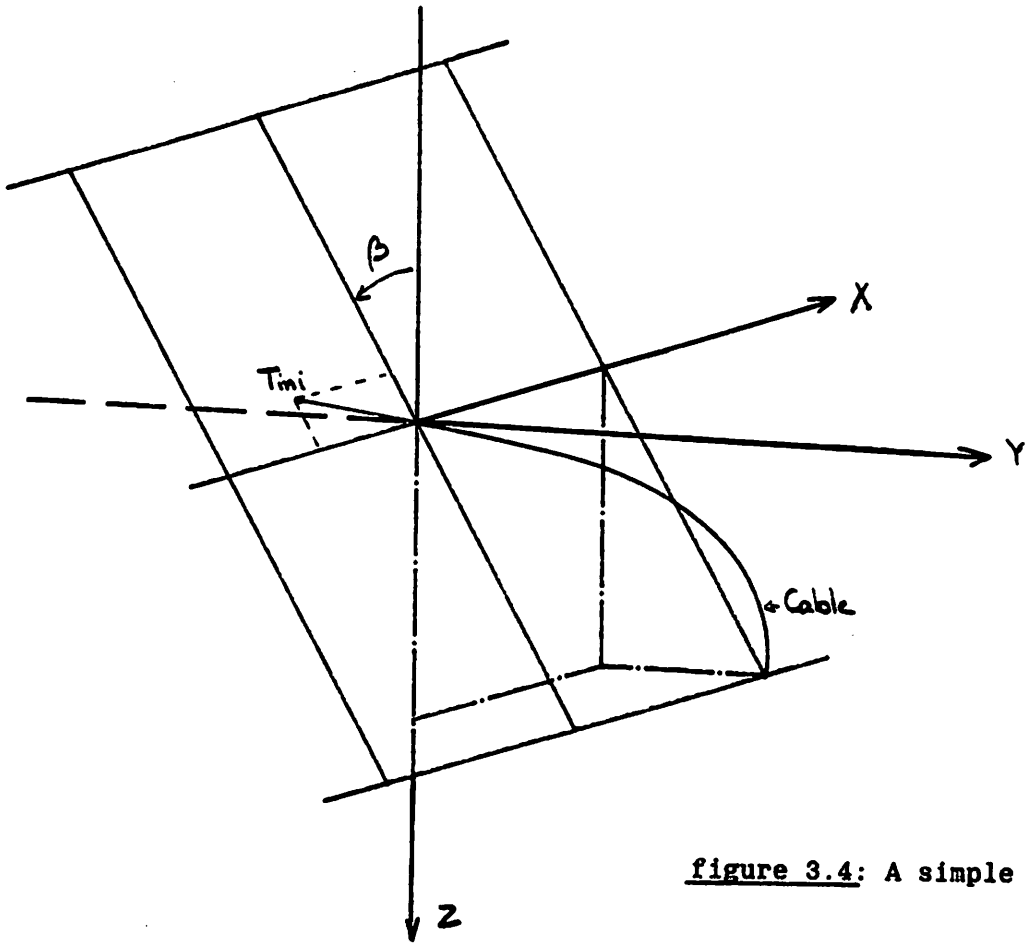


figure 3.4: A simple 3D case.

Figure 3.5 shows the shape variation of the cable with ω = -0.5 N, 0.0 N, and 0.5 N, for a speed current of 0.75 m/s, a diameter of 0.035 m, an initial tension of 200 N, and an initial angle of -20 degrees.

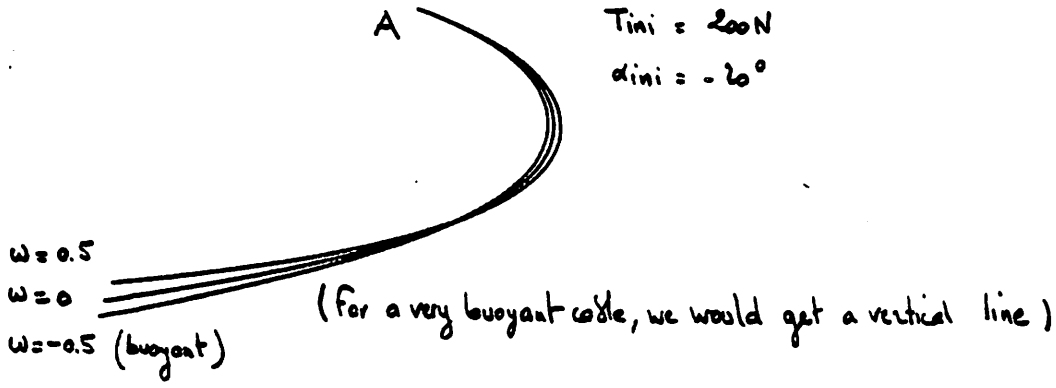


figure 3.5: Variation with ω

The application for the simulator follows:

The relative position of the ROV to the surface ship is Xr,Yr,Zr in $(\hat{I},\hat{J},\hat{K})$

The current is $\hat{V} = \cos(\beta)\hat{i} + \sin(\beta)\hat{j}$

$xr = \cos(\beta).Xr + \sin(\beta).Yr$ and $yr = -\sin(\beta).Xr + \cos(\beta).Yr$ give the relative position in the system defined with $\hat{v} = \hat{V}/V$, as: $(\hat{v},\hat{v}\times\hat{K},\hat{K})$.

To obtain the orientation γ of the bidimensional plane from this latter system, we have $\tan(\gamma)=yr/Zr$. The bidimensional problem has then as desired relative position:

$$X = xr$$

$$Y = \sqrt{(yr^2+Zr^2)}$$

γ and β will allow the final tension to retransform into $(\hat{I},\hat{J},\hat{K})$

For the simulator, in order to reduce the computation time, the Elasticity module of the cable is supposed infinite, and the current is also assumed constant in intensity.

3.4 Integration.

The differential equations are integrated on small elements with the Newton method. The step of integration is given by 1/50 of the cable length. A smaller step could be obviously chosen, but would use more computation time. The best alternative is to get a variable step, especially when the tension gets very small. To resolve the problem of small tension in special condition (small T_{ini} and small α_{ini}), the tension is low-limited to 5 N. For the same reason the angle is limited to -178 degrees. This avoids the cable going up, because of having too large

steps in special conditions. Usually, these problems do not occur. The step of 1/50 of the cable length is a good compromise between precision and computation time. Figure 3.6 shows the extremity differences between a 1/1000, 1/100, and 1/50 steps. The final tension at the end of the cable varies by 3 percent only, which is less than the precision we can expect with such a computation.

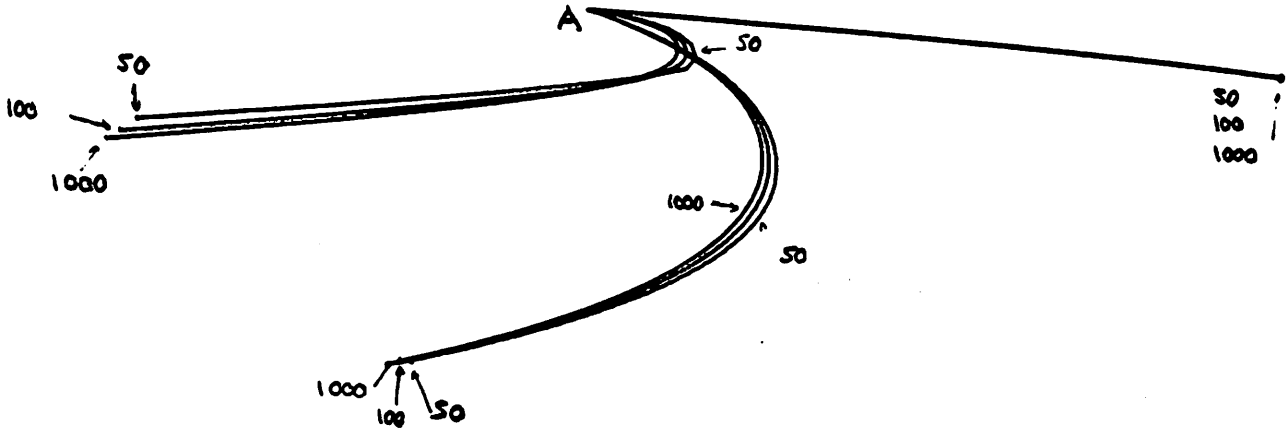


figure 3.6: Variation with the step.

3.5 The algorithm to get the desired deflection.

The objective is to compute the initial tension T_{ini} and its orientation α_{ini} in A, so that we achieve a desired deflection X,Y at the other extremity of the cable B.

When the end is reached then we know the tension at that extremity and its angle. Thus we know the reaction of the cable when the ROV is at that particular position.

Figure 3.7 displays the variation of the extremity obtained with

different α_{ini} for a given T_{ini}. Figure 3.8 shows the variation of the extremity for different T_{ini} for a given α_{ini}.

We can see that the variation in x with α_{ini} is monotonic, thus a simple algorithm would fit to get the best α_{ini} for a given T_{ini}. The best can be defined as the one minimizing the distance between the extremity desired and the one obtained. Thus the search for the α is a minimum search. The direction of progression in α is determined from the variation of distance to the target. In the algorithm developed, the number of points for the search is finite, and the set of points covers the useful range of variation for the α_{ini}. With the two last tries in α_{ini} ordered in the increasing direction, we can define a line which will give the position of the desired extremity relative to the one obtained. This position allows us to use a binary search for the tension (fig. 3.9). The algorithm is thus:

- X,Y is the desired extremity
- x,y are the extremity from calculation

1) -> $T_{ini} = T_1$ a starting tension

2) -> search for α_{ini} which minimizes $(x-X)^2+(y-Y)^2$

3) -> IF the distance is less than the tolerated error, a "solution" has been found

IFNOT order the two closest computed extremities with x increasing

x_1,y_1 & x_2,y_2 $(x_2 > x_1)$

compute $K = (y_2-y_1)*X - (x_2-x_1)*Y + (x_2 \cdot y_1 - x_1 \cdot y_2)$

IF $K > 0$ increase tension T_{ini}

IF $K < 0$ decrease tension T_{ini}

GO TO 2

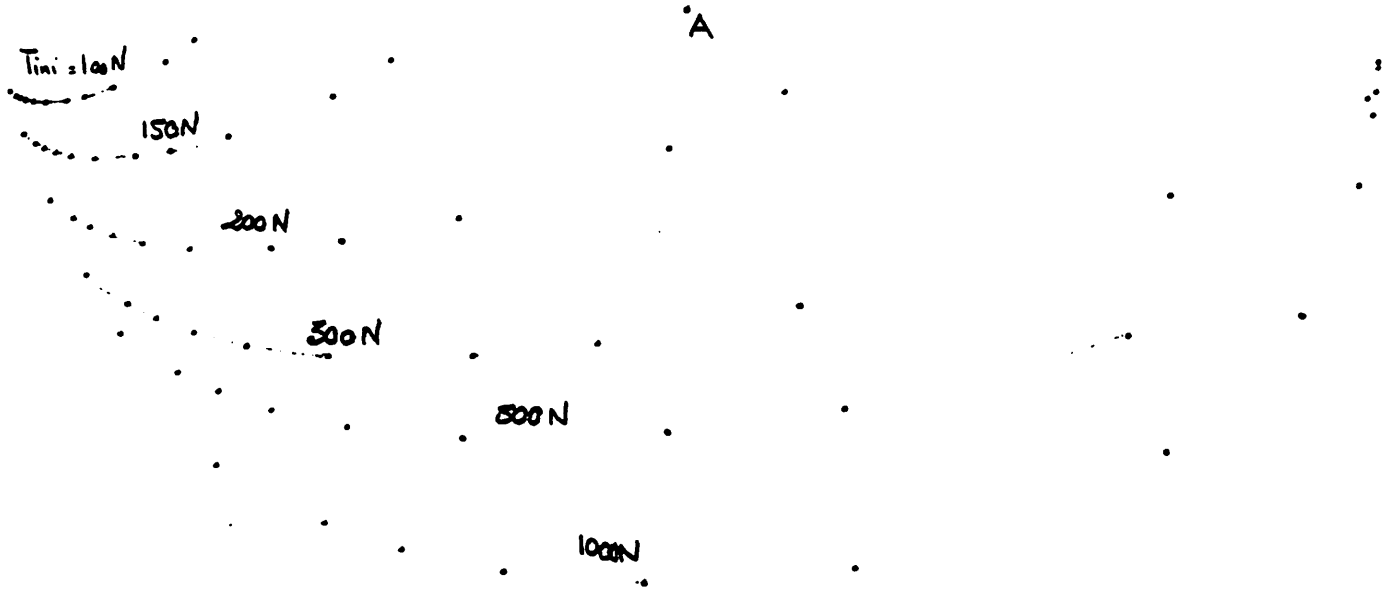


figure 3.7: Variation of the extremity B with α_{ini} for a given T_{ini}

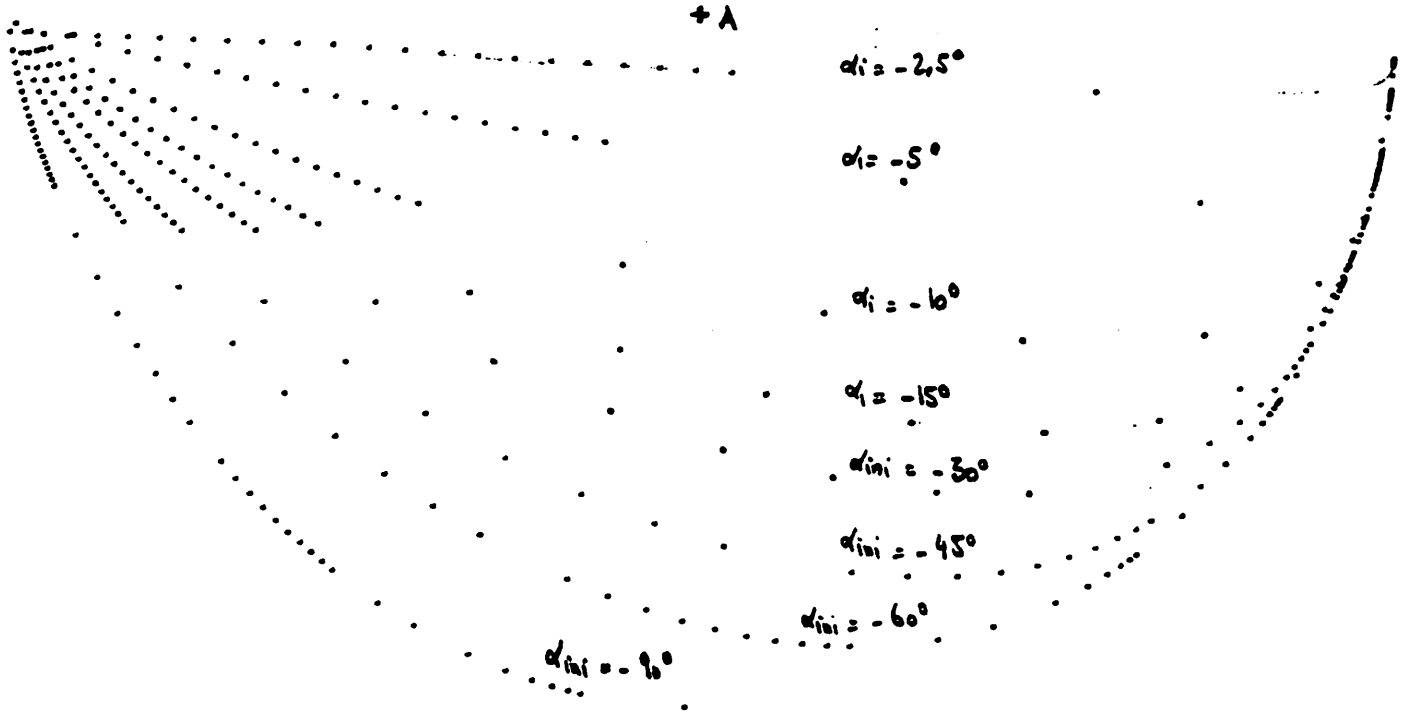


figure 3.8: Variation of the extremity B with T_{ini} for a given α_{ini}

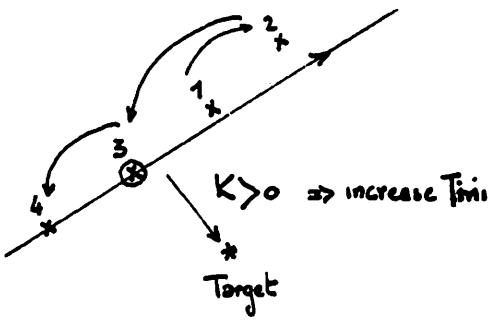


figure 3.9: Search for best angle and tension

The listing can be found in Appendix B

The tension modification can be done on a binary search principle. The precision will vary depending on the method to find the best α_{ini} for a T_{ini}. In the program the search in α is done for a discrete set, the step between two trials is varying depending on α since a same step variation do not give the same deflection variation with for instance, α=-5° and α=-30°.

The ROV is not moving very fast so a new search with the previous set converges rapidly. Two examples are given in figures 3.10 and 3.11.

3.6 Some numbers.

As I wrote in the previous chapter, the force from the cable can be important and varying. Here are some tension values obtained for a cable of 0.035 m of diameter, in a current of 0.75 m/s, for different relative

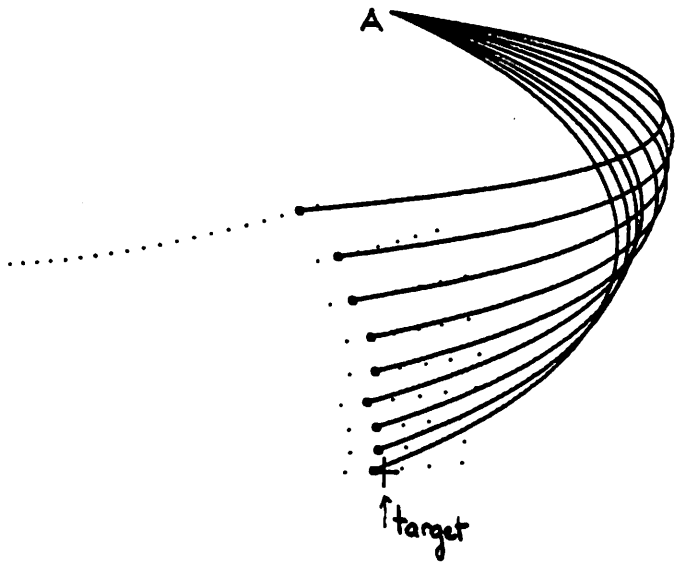


figure 3.10: Convergence from a far set of initial conditions

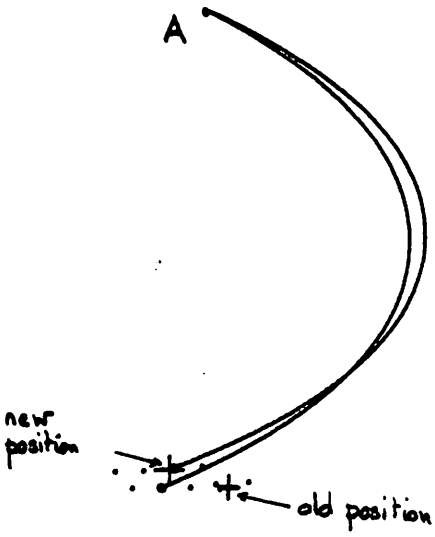


figure 3.11: Convergence from a close set of initial conditions

positions between the points A and B (i.e. the ship and the vehicle), and a cable length of 150m.

XA - XB = 0m ; YA - YB = 100m ; => Tension ≈ 450 N

XA - XB = -70m ; YA - YB = 70m ; => Tension ≈ 350N

XA - XB = -95m ; YA - YB = -10m ; => Tension ≈ 200 N

XA - XB = 95m ; YA - YB = -10m ; => Tension ≈ 45 N

CHAPTER 4

A "3-DIMENSIONAL" (3-D) GRAPHIC DISPLAY FOR THE SIMULATOR

4.1 Reasons for a "3D" display and reason for a computer solution.

The principal tool of ROV pilots, at the moment, is the video camera feed-back. Thus limiting a simulator to only the indicators of vehicle position would have simplified the software but created a unrealistic simulation relative to the information really available for an ROV system.

Hence the problem was to generate a "3D" display of an environment. "3D" here means the representation on a surface, of a 3-dimensional scene, the equivalent of a camera.

Two solutions exist in creating such images. The first one uses a TV camera and a scale model, whereas the second is developed from a computer.

The former presents some advantages, when the model is carefully built. A fiber optic camera mounted on a special support can generate very realistic images. Electronics can be added to limit the field of view, etc... Nevertheless the camera and its support, even small, limit the moves to environments without structure. The model is also very often untransportable and needs a specific room. This solution is mostly used for simulators of manned submarines, where the piloting cabins, as well as possible defects, are modelled.

The second solution is the method used for flight simulators and provides a sufficient display with high-level color screens, though not as

fine as does a scale model. These displays have built-in processors to take care of hidden face removal, shading, and perspective effect; they give very soft sequencing images. Their big drawback is mainly their high cost $80,000 (or more for the sophisticated ones). These displays are conceived for general purposes. Thus the idea of creating a lower-cost computer generated image, is to use a standard display and develop a specific software for the environment we want to represent. With a standard color display, an alternative can be also to generate a finite number of images of the environment before the run. The picture represents the environment watched by an outside observer. Then the vehicle is drawn as a wire frame moving on that image (ref. 10). The observer can take a finite number of places. This alternative represents a 3D scenery but is not what the pilots would get on a real system. The "best" solution is to show what the camera can see. Thus the images have to be regenerated at every step.

I developed such software. The computer solution was chosen because of the disadvantages of scale models (the Man-Machine Systems Lab has a camera support able to move with 5 degrees of freedom, reference 9). A standard graphic display is the constraint for budgetary reasons, since a simulator has to stay within a relative cost compared to an ROV. We can imagine that once the software has been developed for this kind of display and a specific environment, a specific board can be built to only run the display algorithms.

4.2 Software development.

Once the above decisions were made, two choices were available for the representation:

a) Wireframe.

This avoids the problem of hidden surface removal but cannot really simulate distance impression or shading.

b) Filled shapes.

This solution is in fact not much more time consuming that the previous one when the elements of the environment are well defined. It also facilitates the comprehension of the scenery.

These two options were bound to the utilization of either an available Megatek vector display (wireframe) or an available Lexidata raster display (filled shape). The Megatek had a 3D graphic library based on parallel projection and so, was not useful for 'Perspective 3D'. For both displays all the computation had to be done with the same algorithm. With an appropriate algorithm a filled shape is reduced mainly to 2, sometimes 1 or 3, surfaces. Eight points need to be computed to draw a parallepipedic wireframe. As a consequence the Lexidata was designated with a filled-shape representation. This display has a built-in function to fill convex polygons.

4.2.a Perspective effect.

The perspective effect is due to the projection through a point, on a given surface, of the points in the space (fig. 4.1).

The explanation of the method is directly applied to the simulation.

Assume that the camera is fixed to the ROV for the moment (no pan or tilt). A point M in the space is referred by its coordinates in a fixed system $(O,\hat{I},\hat{J},\hat{K})$ as (X,Y,Z). The local system, including the point and the plane of projection, is linked to the vehicle as $(P,\hat{i},\hat{j},\hat{k})$. Let the plane of projection be: x=a=constant (figure 4.2) and P be the point of projection (see chapter 2).

The coordinate of the point M in the ROV system are :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = T(\psi,\phi,\theta).\begin{bmatrix} X-X_P \\ Y-Y_P \\ Z-Z_P \end{bmatrix}$$

where (X_P,Y_P,Z_P) are the coordinate of P in the fixed system

The projected point is then given by:

$$\begin{cases} x' = a & \text{since it is in the plane of projection} \\ y' = y/x \\ z' = z/x \end{cases}$$

For our use this projection is very convenient. It conserves segments and convex surfaces. Its effect is, of course, that the farther away an object is the smaller it will be represented on the image.

But we need only to project a specific region of the space, the field of view of the camera. Thus the projection has to be executed after a 3D clipping.

4.2.b 3D clipping.

The clipping is the shortening of a segment or a surface relative to a given domain (fig. 4.3). Clipping is most often used for 2D, however its principle can be adapted to a 3-dimensional case.

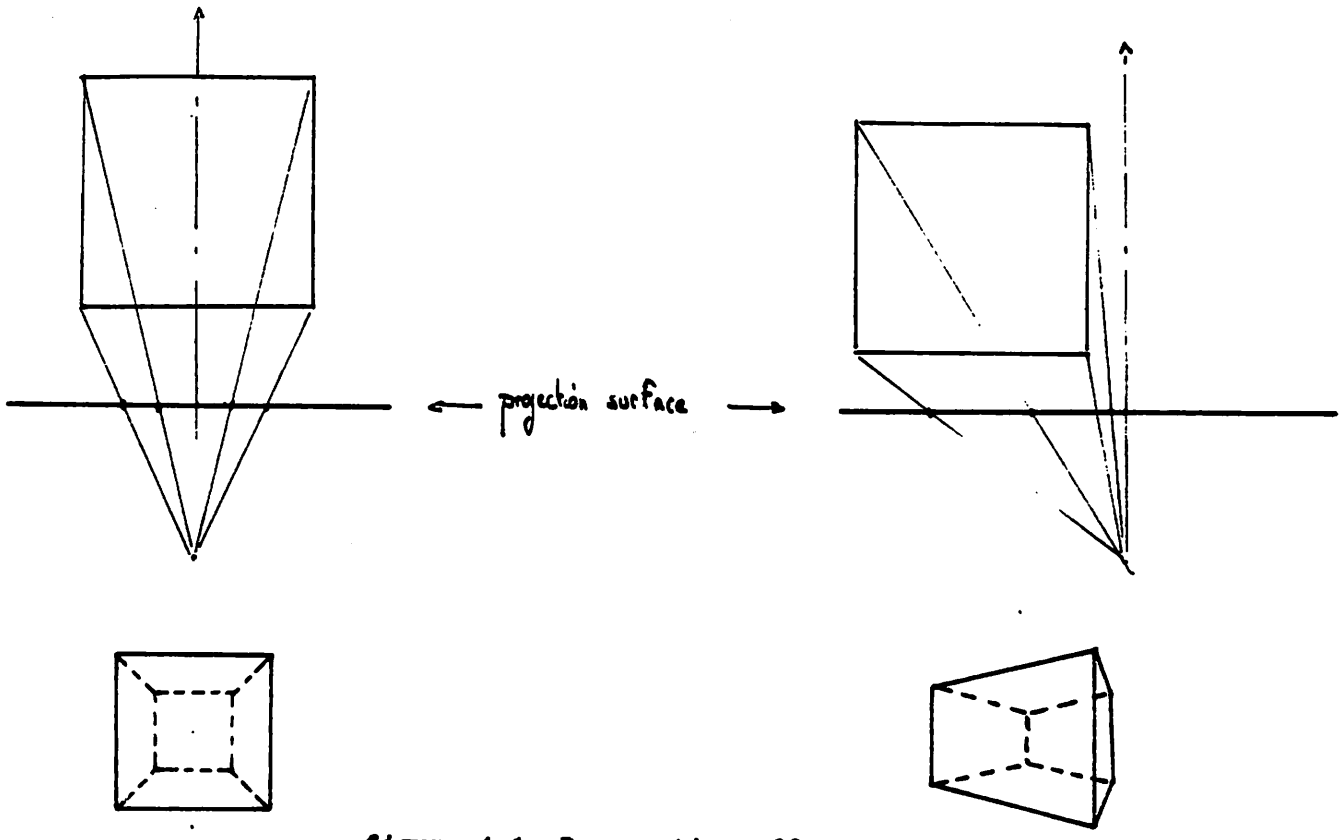


figure 4.1: Perspective effect.

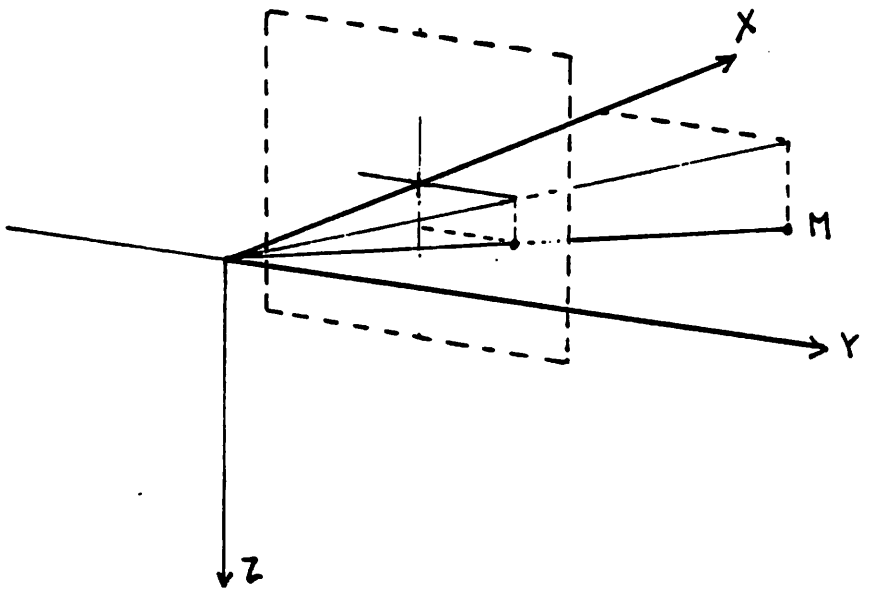


figure 4.2: Projection for perspective effect

In our case the spatial domain is limited by 4 planes whose characteristics are defined by the field of view of the camera or the ratio between the dimensions of the projection surface and its distance to the projection point (fig. 4.4).

With a projection surface: 2b×2b and a distance: a, the four planes can then be represented as:

PL1: $f_1(x,y,z) = b.x - a.y = 0$

PL2: $f_2(x,y,z) = b.x + a.z = 0$

PL3: $f_3(x,y,z) = b.x + a.y = 0$

PL4: $f_4(x,y,z) = b.x - a.z = 0$

These equations define 16 regions in the space. The one defined by all $f_i(x,y,z) > 0$ is the region along $x > 0$ (fig. 4.4). This is the region that we need.

The basic element to draw the filled shapes is a convex plane surface defined by 4 points in the space. A specific subroutine (POLY see listing) clips the polygon relative to the previous planes, to form the surface (which is also a polygon) inside the desired space. The surface can then be projected and its projection stays within the limit of the projection surface.

4.2.c Hidden face removal.

These three words in graphic software are synonimous of long time of computation and complex algorithm. This is indeed, a very difficult problem when dealing with volumes or surfaces of any shapes. In the simulation, I chose a simple world of 3 main elements: 4 columns linked by

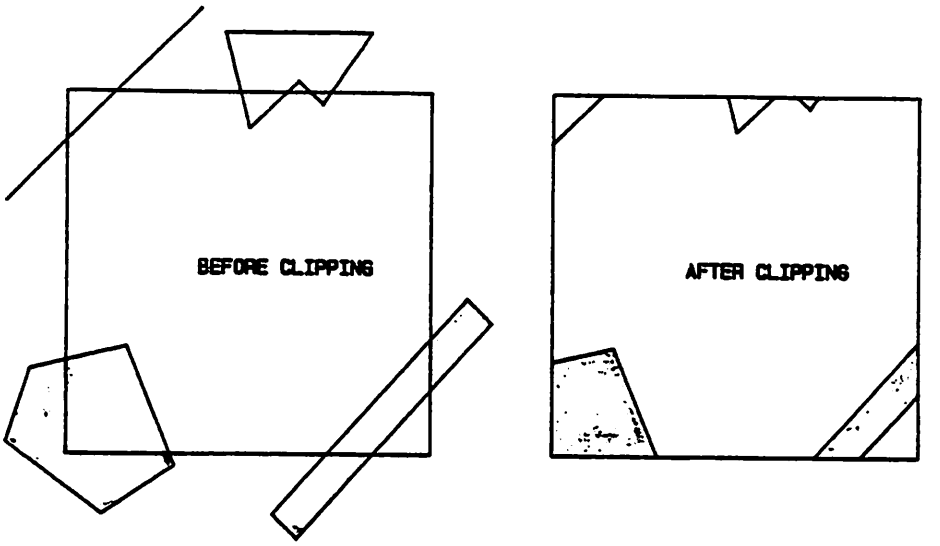


figure 4.3: Clipping.

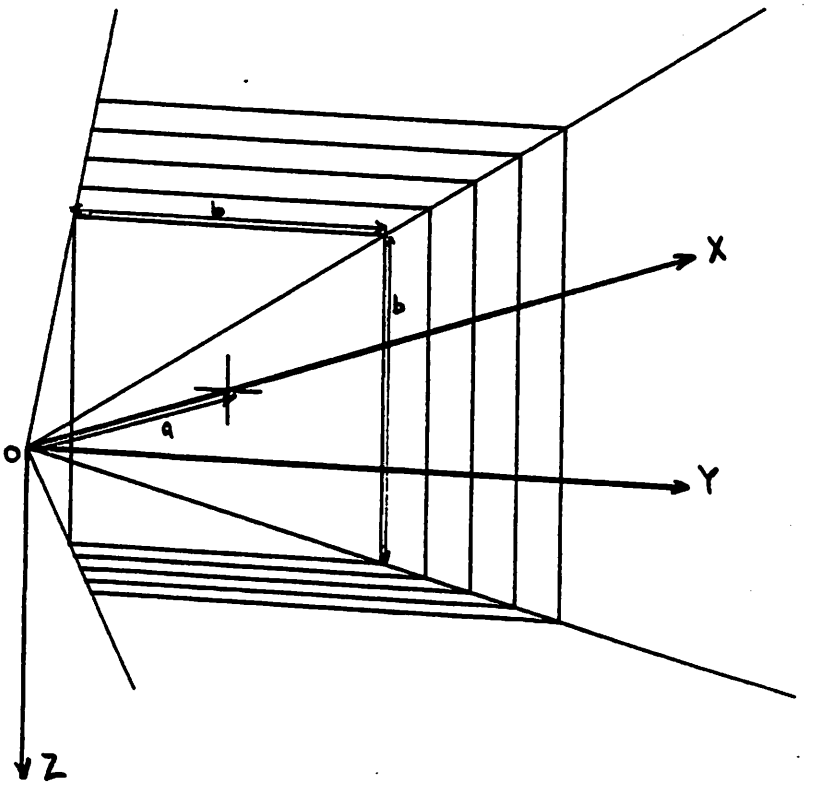


figure 4.4: Field of view.

4 bars for the offshore structure, 1 cube for the wellhead, and another bar for the pipe-line (fig 4.5). Each element is separate from the other, so that a first sort allows one to know which one has to be drawn in first. A specific algorithm for each element is then utilized to sort the faces.

With filled shapes hidden surface removal is greatly facilitated since the only information we need to know is which surfaces are behind, and which are in front. The one behind is obviously generated first, and the one following will cover the part of the previous one, that it hides, by filling.

To optimize the computation time, some surfaces, which are impossible to see are not displayed at all.

As I wrote before, the sorting of these surfaces, if not impossible, can be very time consuming for complex bodies.

The case of a cube is very easy. The region around the cube can be separated into 9 parts (fig. 4.6). When the "camera" is in a particular position, only the corresponding surfaces are clipped and projected to get the images (fig. 4.7). The top of the cube is displayed depending on the altitude.

For the offshore structure, since the body is symmetrical, a simplification can be made, with an adequate parameterization of the surface coordinates. The algorithm is developed on the same principle as the cube (fig. 4.8, 4.9 & 4.10).

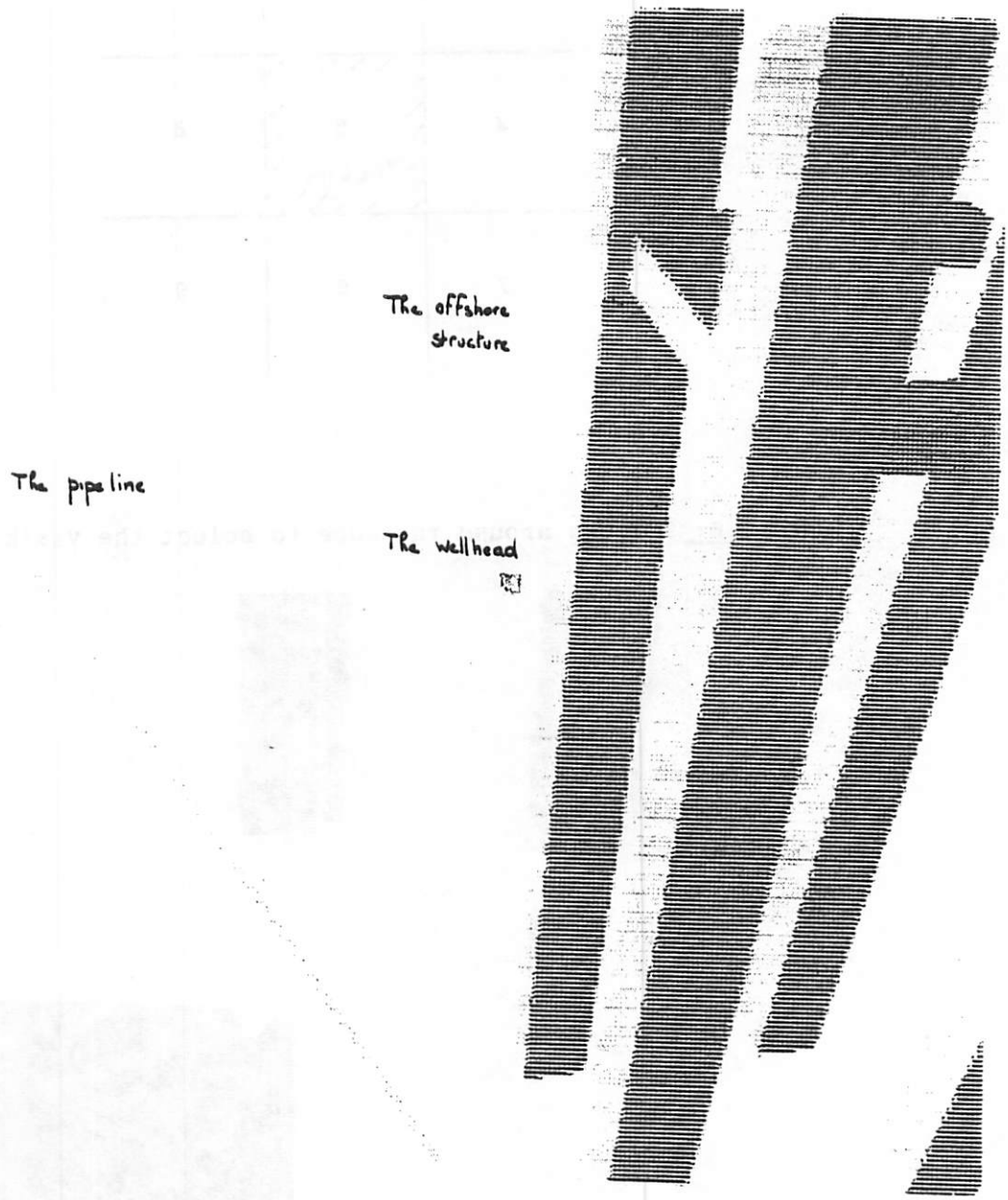


figure 4.5: The whole environment in perspective.

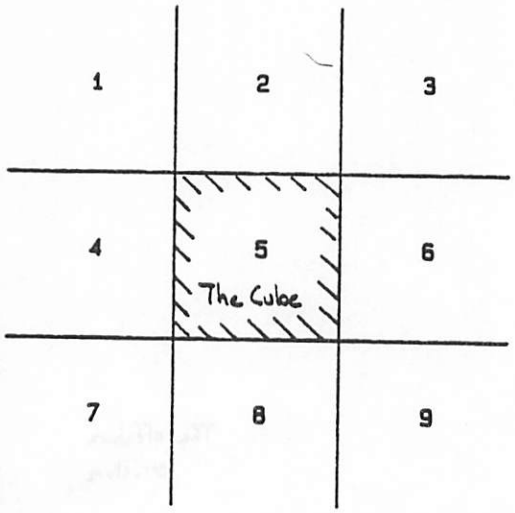
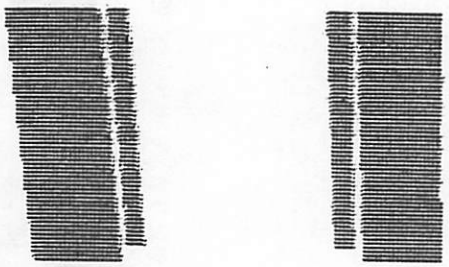


figure 4.6: 9 areas around the cube to select the visible faces.



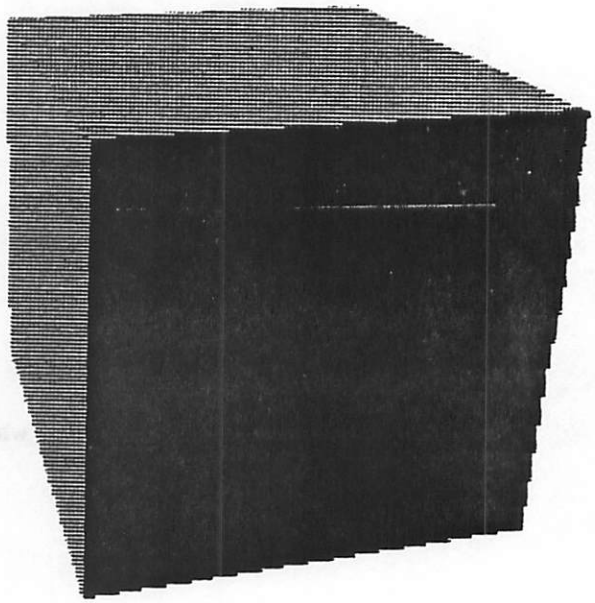


figure 4.7: Copy of the lexidata screen.
The cube in perspective.

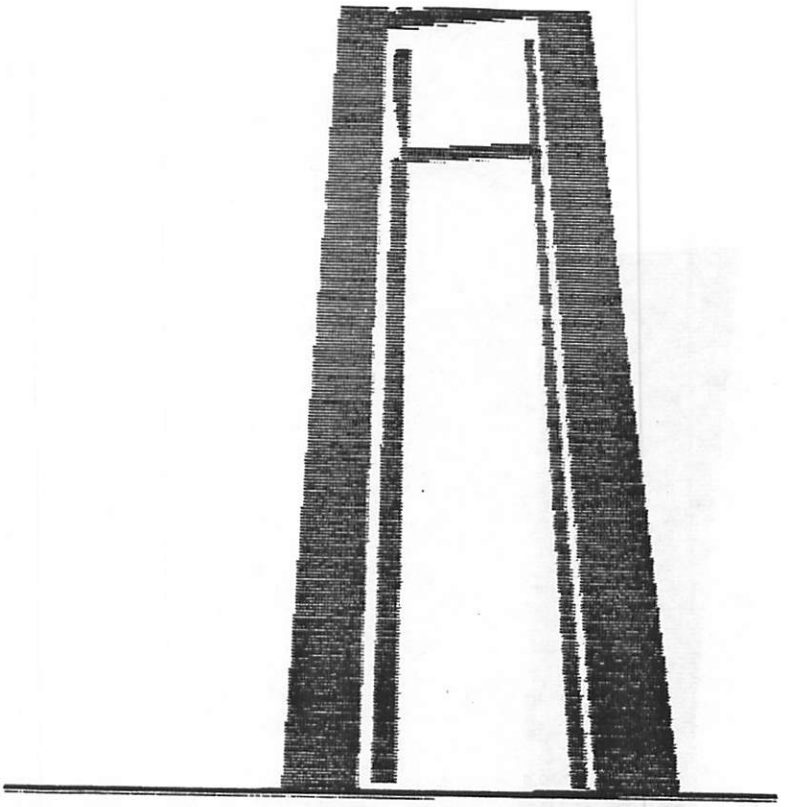


figure 4.8: View of the offshore structure.

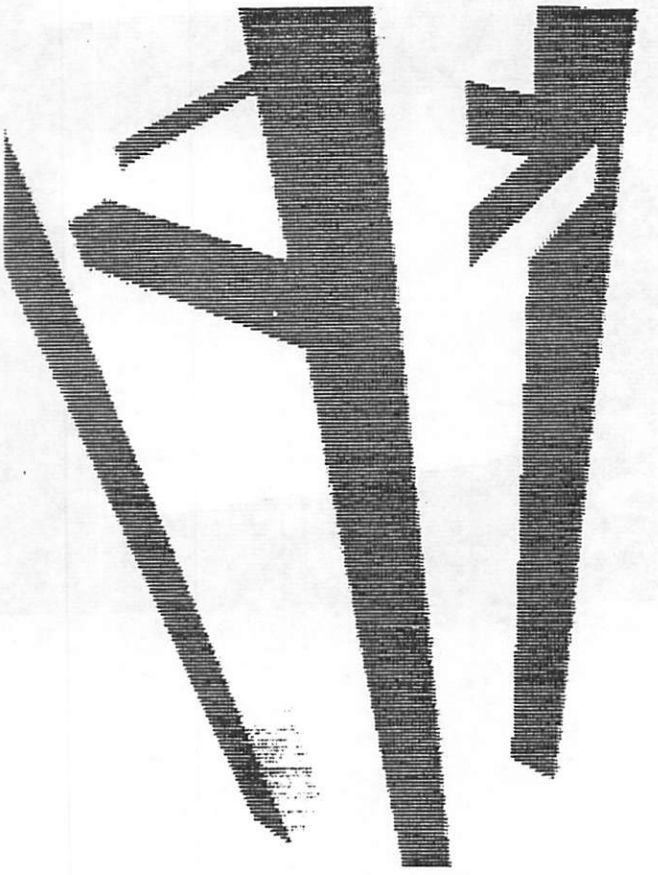


figure 4.9: View of the offshore structure.

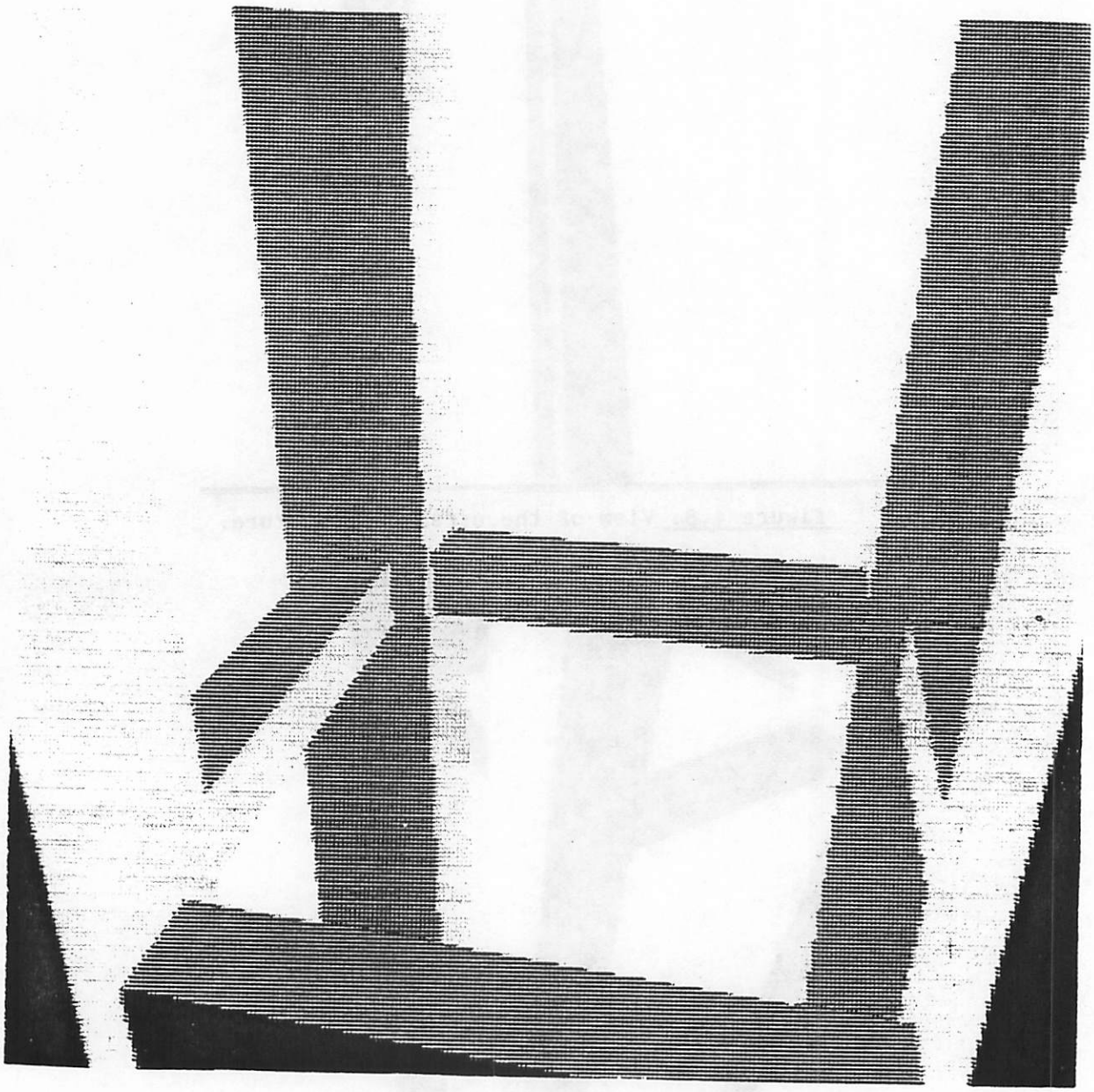


figure 4.10: View of the offshore structure.

4.2.d Shading.

Shading is helpfull for a filled-shape representation since it allows the surfaces to be differentiated visually. But there is in fact, an obligation to be able to differientate between one surface from another.

The surfaces of the shape are planes even after being clipped. After transformation of the coordinates into the camera system, it is thus possible to calculate an orientation and find the angle between the normal surface and the axis of the camera. The filling color level of the surface can then be adjusted to that angle. It is also possible to calculate the distance of the camera to the surface. The intensity of the color can be modified depending on that distance. These data combined simulate the light coming from the camera.

4.2.e Pan and Tilt for the "camera".

This is only a problem of system transformation. The camera is supposed to be at C:(x=1.5, y=0, z=0) in the ROV. The pan is a rotation β along an axis parallel to $\hat{k}$, passing through C. Then the tilt is a rotation α along the axis parallel to $\cos\beta.\hat{j}-\sin\beta.\hat{i}$. Thus a point M in $(O,\hat{I},\hat{J},\hat{K})$ is transformed in x_c,y_c,z_c coordinates in the camera system by:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} c\alpha & 0 & -s\alpha \\ 0 & 1 & 0 \\ s\alpha & 0 & c\alpha \end{bmatrix} . \begin{bmatrix} c\beta & s\beta & 0 \\ -s\beta & c\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} . \left[T(\psi,\phi,\theta) . \begin{bmatrix} X-X_p \\ Y-Y_p \\ Z-Z_p \end{bmatrix} - \begin{bmatrix} 1.5 \\ 0 \\ 0 \end{bmatrix} \right]$$

4.3 Improving the shapes and the representation. The capabilities.

The software written gives some good results. However, there are many ways to improve the images.

The Lexidata could use only 15 different grey levels (for more explanation on that subject see Appendix B). A larger number may improve the representation slightly.

The grey level was calculated for the whole surface. By spreading the surface in some smaller patches the effect of fading with distance would be more sensible and would also facilitate the approach very near a surface.

Finally, the square columns, bars or pipe-line, may be replaced by some more complex or more realistic shapes. The algorithm would not be more complicated but the computation time would increase in accordance with the number of points added, mostly for the clipping time. For only the offshore structure the clipping process takes .45 seconds per loop.

The software is implemented on a PDP 11/34. Its average time for a floating point multiplication is 0.02 ms (there are not only multiplications in the clipping program, but some array handling). The average time to display the environment in an unfavorable situation is about 0.75 s, when the whole environment appears and is clipped in some parts (time when only this module is running).

CHAPTER 5

THE ENVIRONMENT EMULATOR

5.1 The reasons and the objective.

A simulator has to reflect as well as possible the vehicle dynamics and its different operating modes. For an ROV, the pilot has to compensate for the cable, the current, the bottom level, the arm, and many other things. A good simulator should recreate all these annoying effects.

The conception of this ROV simulator is modular, and thus one module can have the specific task of complementing the dynamic model by detecting any interactions with the environment and computing the different forces and corresponding moments.

The detection is not so complicated, and is the only part written in that simulator. The computation of the reaction forces, for instance when the vehicle hits an element at a particular point, is much more complex and requires a great sophistication to simulate as it really is (fig 5.1).

The environment emulator could eventually include an an arm, detect if the arm is at the position to grab an object, and generate the change in the ROV dynamic and static due to the object grabbed.

When the vehicle enters or turns around a structure, the cable might rub some elements, where its shape and the extremity tension would obviously be modified. This is especially the case when the pilot, through inattention, makes a loop around a bar (fig. 5.2).

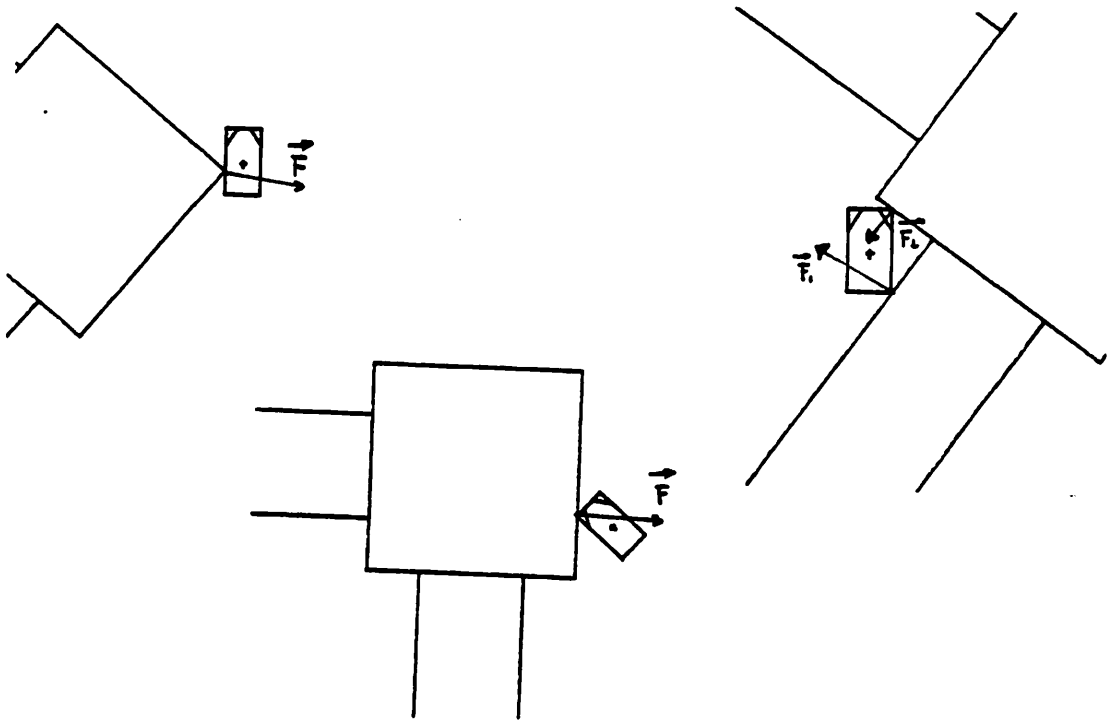


figure 5.1: Which forces apply at the collision point?

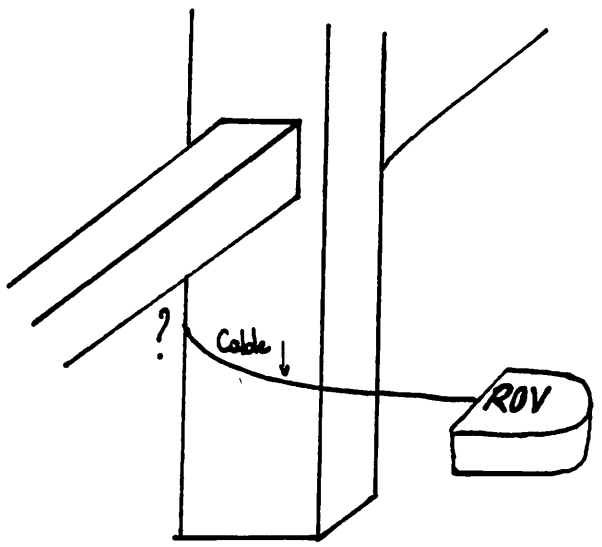


figure 5.2: The cable around an elememt.

5.2 Collision detection software.

The environment emulator of this simulator is limited compared to what should be done eventually. The program is only able to detect (and signal) any collision with an element of the environment. Hence, the vehicle is not stopped when it hits something. A simple algorithm (without knowing the force reactions) could have stopped it in (X,Y,Z) in a specific direction, but the yaw would have necessitated more computations. The best method in order to prevent the body from going through the structures would be still the force reactions which should be included in the dynamic model giving the motions. But this needs to be able to detect and compute the data fast enough, so that they are taken into account immediately by the system dynamic, and was not possible in this study.

For the detection, the ROV is now modelled as a rectangular solid and includes neither roll nor pitch. The program is detailed below for the collision with the cube representing the wellhead; the collisions with the other elements of the environment are very similar. These are the different steps:

- The altitude is checked, if not within the cube range: no collision
  if within the cube range: proceed further

Referring to the figure 5.3:

- If the center of the ROV is in area 1: no collision
- If its center is in area 3: collision
- If it is in area 2: need for more test:

  The position of the 4 points of the ROV are computed.

- if either one of them is within the cube surface: collision.
- if not proceed further

The 4 points of the ROV define 4 oriented lines $f_i(x,y)$. All the inside points of the ROV are so, that $f_i(x,y)$ are all positive. To know if one point of the cube is in the ROV, the $f_i(x,y)$ are calculated.

- if they are all positive: collision.
- if not: no collision.
- if no collision has been detected for the four corners of the cube then the ROV is safe.

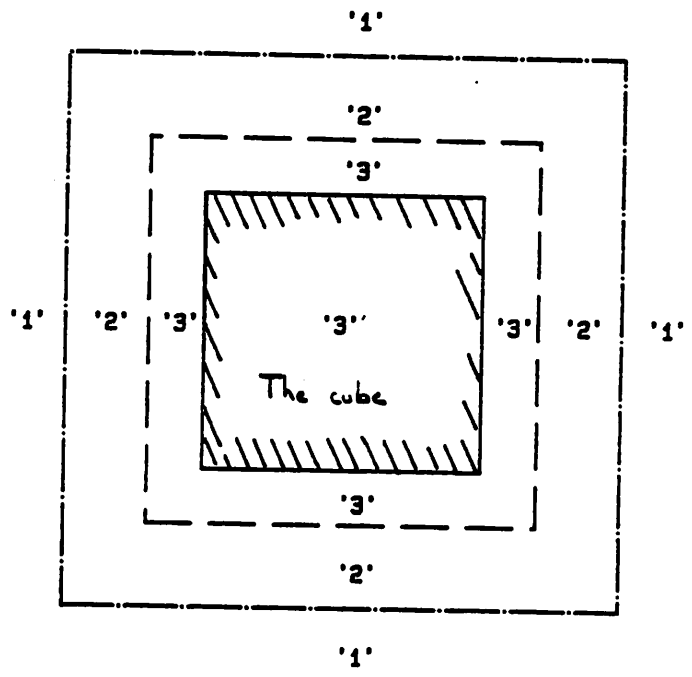


figure 5.3: Areas of collision around the cube (the wellhead).

The listing of this module can be found in Appendix B

CHAPTER 6

GENERAL CHARACTERISTICS, PILOTING THE SIMULATOR, THE LINK MODULE

6.1 General characteristics.

6.1.a The ROV modelled.

This simulation can be utilized for many ROV's, propelled and oriented by thrusters. Their characteristics have only to be specified in a data file. The ROV modelled for this study is the one being redesigned by the MMSL [Man-Machine Systems Lab] in cooperation with Sea Grant, originally from a RECON 5 built by Perry Oceanographic (fig. 6.1, ref. 15).

Some parameters were available like: mass, inertia, thruster positions. The other coefficients have been obtained by interpolation from other vehicles. We have:

- mass = 400 Kg (≈ 850 pds)
- Ix = 30 Kg.m^2
- Iy = 100 Kg.m^2
- Iz = 100 Kg.m^2
- distance center of gravity-buoyancy = 0.2 m

The added masses can be estimated by the mass or inertia multiplied by some coefficients (ref: 16):

- m11 = 250 Kg
- m22 = 400 Kg
- m33 = 400 Kg
- I44 = 20 Kg.m^2
- I55 = 100 Kg.m^2
- I66 = 100 Kg.m^2

The power of the thrusters have been measured statically on the real vehicle and we get:

- P1 = P2 = P3 = P5 = P6 ≈ 500 N at full power.

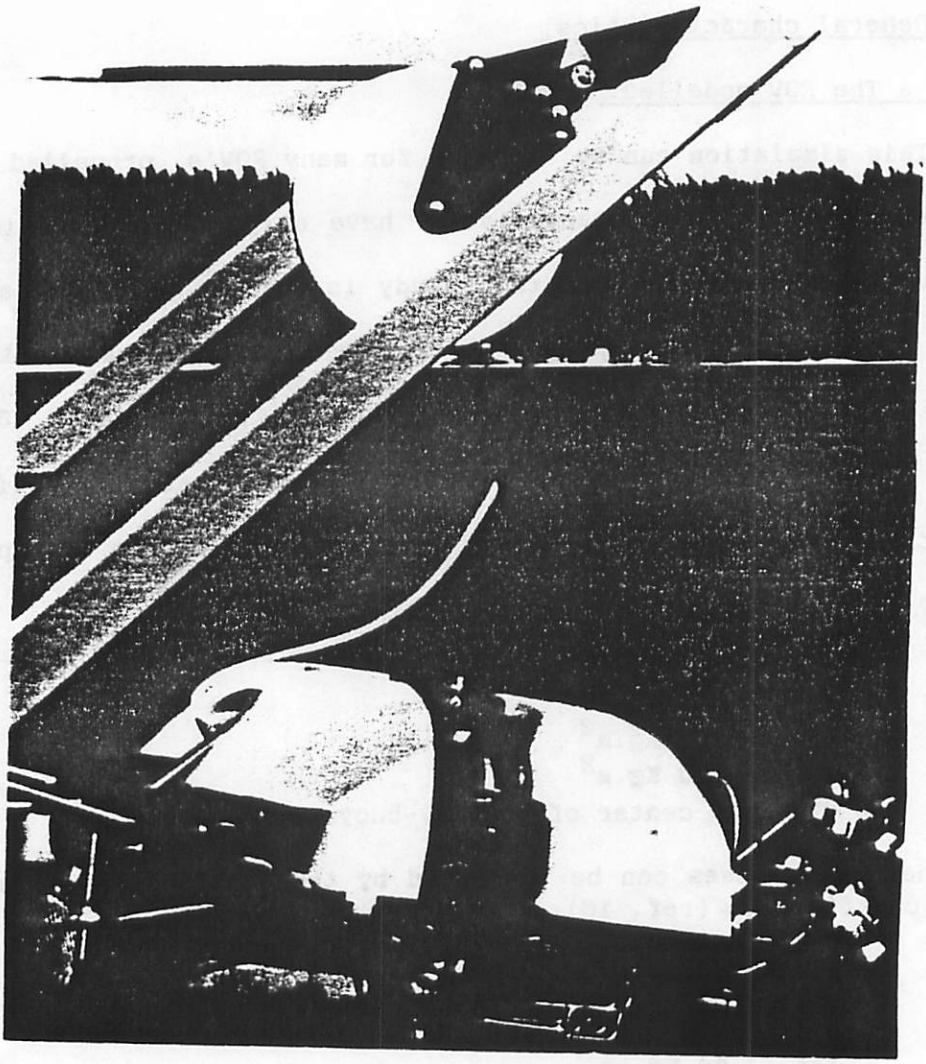


figure 6.1: SEA GRANT 1.

For an ROV the thruster power is generally limited by an hydraulic pump. This can be taken into account in the simulator.

The hydrodynamic coefficients were chosen so that the limit speed of the ROV does not exceed the "usual" speed of ROV's (in the simulator: 3.2 knots). These coefficients will have to be adjusted with sea trials or by tests in a towing tank:

$C_1uu = C_2uu = 150$
$C_1vv = C_2vv = 200$
$C_1ww = C_2ww = 200$
$C_1pp = C_2pp = 450$
$C_1qq = C_2qq = 700$
$C_1rr = C_2rr = 800$

For the collision detection the ROV is modelled as a parallepiped of dimensions: 3m × 1.6m × 0.8m .

The direction of the current can be also changed without any problems through the ROV data file.

6.1.b Implementation.

The simulator has been implemented on the PDP 11/34 of the Man-Machine Systems Lab. As presented before, it consists of four modules linked by a main program. These modules are independent. In order to run, the simulator needs the main and the dynamic modules. The others can be added depending on needs. This configuration allows easy test or modification. The different modules can be simplified, for instance by using 4 degree-of-freedom dynamics instead of the complete set, or by using a "3D" display with only one element.

When all the programs run together, the dynamic module runs at 20-30 Hz, and the others are sequenced by the main modules at a maximum of 3 Hz.

The "camera" frequency varies between 3 Hz and 1 Hz, depending on the scene. This organization, instead of a sequencing through the dynamic module, was chosen in order to allow the dynamic module to run at a reasonable frequency.

6.2 Piloting the simulator.

Figure 6.2 shows a general view of the installation. It consists of 2 displays, 1 small (3 inputs & spring return) joystick (the same one which will be used for the real ROV), a bigger analogic input board, and a keyboard. At the moment the terminal screen is not used (only when something wrong happens in the programs!).

The smaller joystick is utilized for the input of the longitudinal and transverse thrusters P1,P2,P3. The rotation of the head of the joystick enables the pilot to differentiate the longitudinal thrusters to rotate the vehicle. The command of the vertical thruster P5,P6 is accessible through a paddle on the larger input board.

On this board, the pilot can also command:

- The roll compensation (which differentiates the vertical propellers).
- The pan and tilt of the camera.
- The trims for the thrusters.
- The position of the surface ship (no dynamics).

Usually on real operation the cable is released at the order of the pilot or copilot. In the simulator, to lighten the task of the operator the length of the cable is proportional to the distance between the ROV and the surface ship. This coefficient can be modified interactively like the others. For instance, a coefficient equal to 1.1 will result in a tensed cable whereas a coefficient 2.0 will give a loose cable.

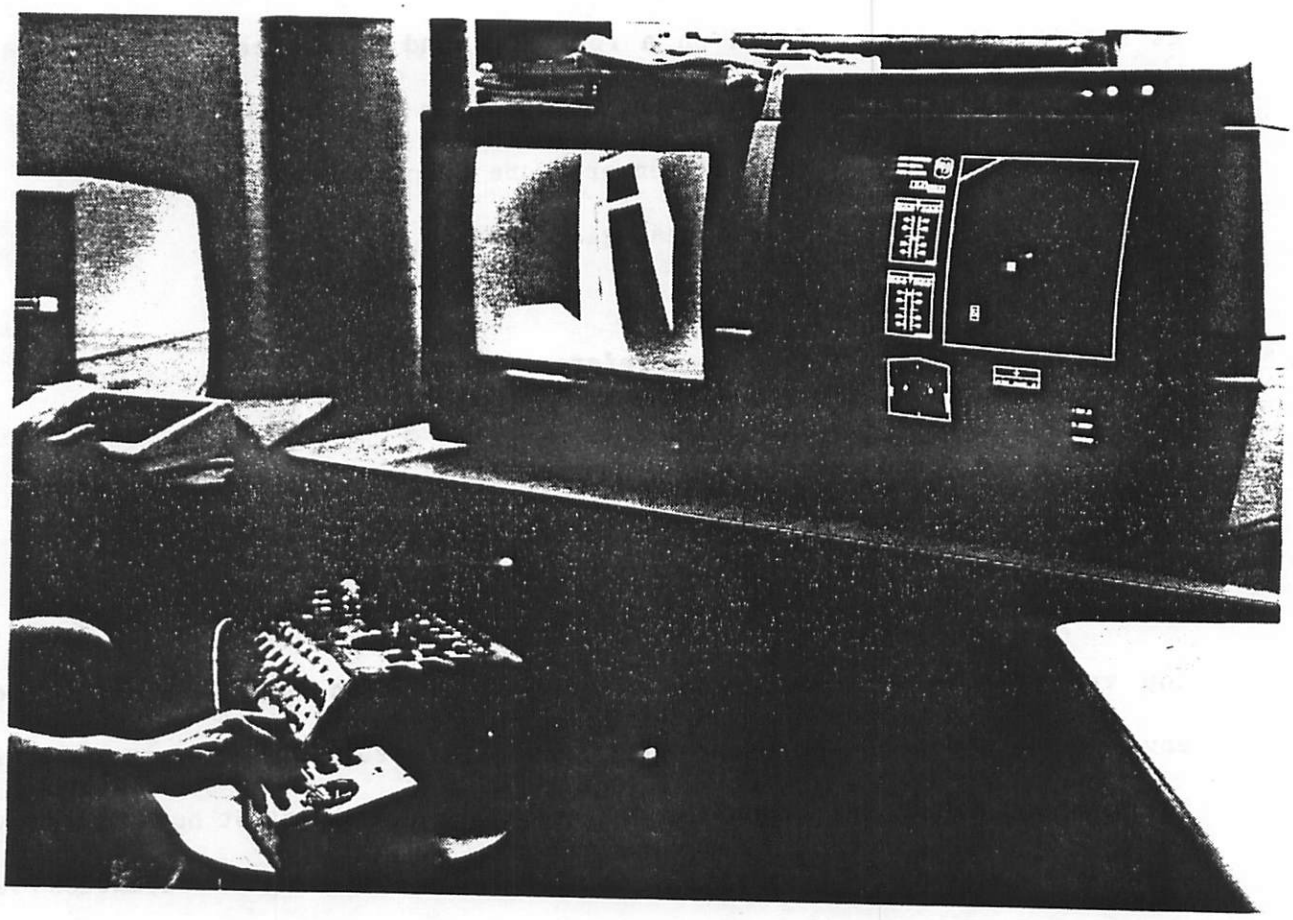


figure 6.2: Installation of the simulator.

These controls are completed by keys on the terminal keyboard. They are used to zoom the top view, select the auto-heading, the auto-positioning, or the auto-depth mode, or change the form of the top-view display (explanation below).

Two screens provide the outputs. A Lexidata color display (used in black & white see appendix B) serves for the "3D" image of the environment as seen by the camera fixed on a real ROV, and a Megatek vector display shows the different indicators and the top-view (fig. 6.3).

The information on that screen are more precisely (fig. 6.4):

- heading (in analogic form)
- roll and pitch (")
- depth
- altitude
- position relative to a reference system (fig 6.5)
- pan and tilt of the camera
- thrusters commands
- top view
- tension at the end of the cable
- cable coefficient (length/distance(ROV;ship))
- bell and crash lines when a collision occurs (fig 6.6)

Two forms of top-view are available (fig 6.3, 6.6). In the first one, the vehicle stays fixed in position and rotation on the screen. The environment moves in accordance and turns. This solution is very helpful for the command of the propellers, since the pilot does not have to worry about the motion of the joystick relatively to the ROV. The drawback is that the effect of the current on the screen will be also rotated. The second alternative is to let the vehicle pivot, thus the environment is only translated. This solution does not modify the current effect on the screen, but complicates the task of the pilot for the control of the thrusters. However, this inconvenience can be corrected if the joystick inputs take care of the ROV heading. Hence the pilot would not have to

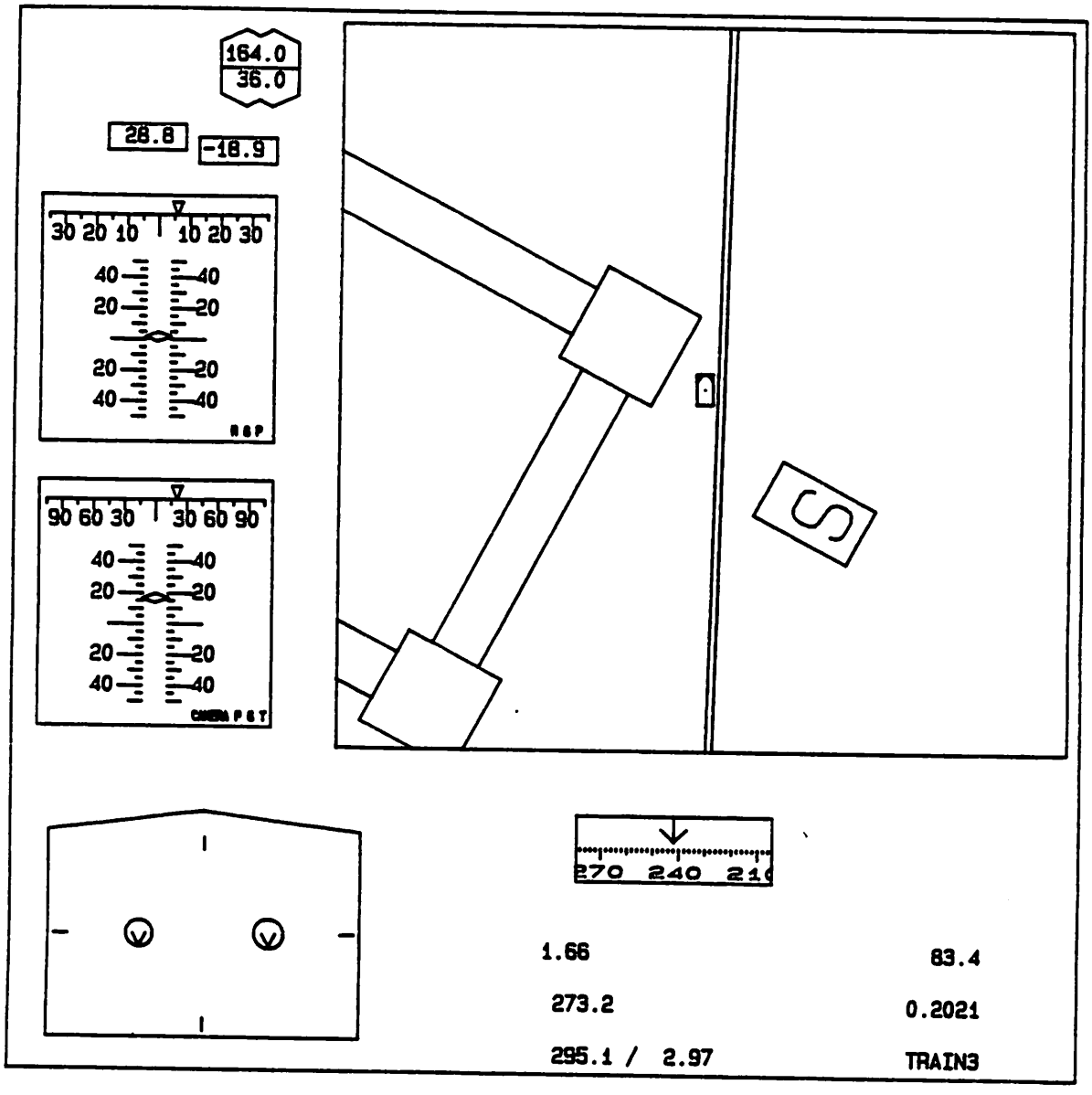


figure 6.3: The Megatek display.

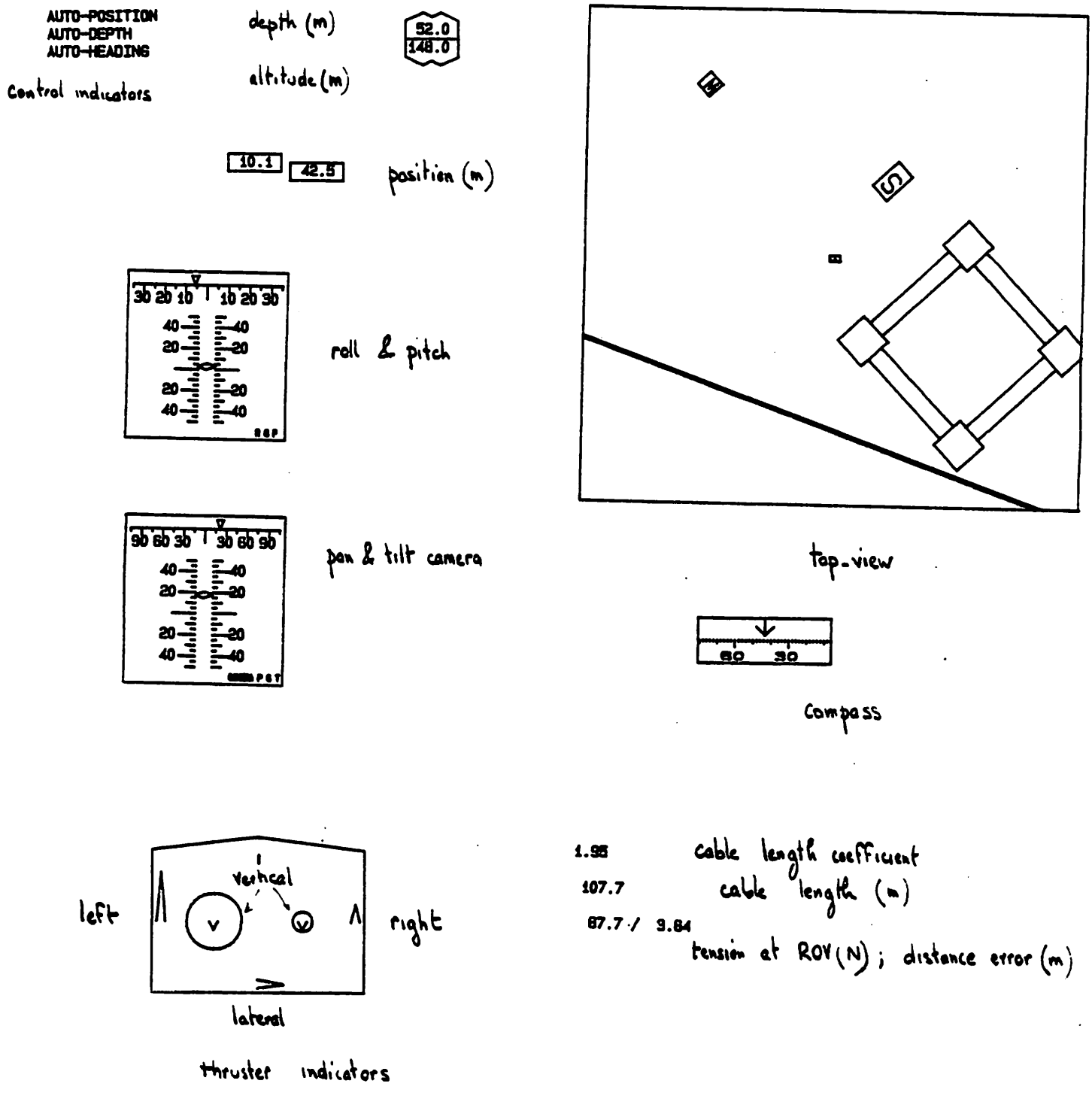


figure 6.4: Detailed view of the Megatek display.

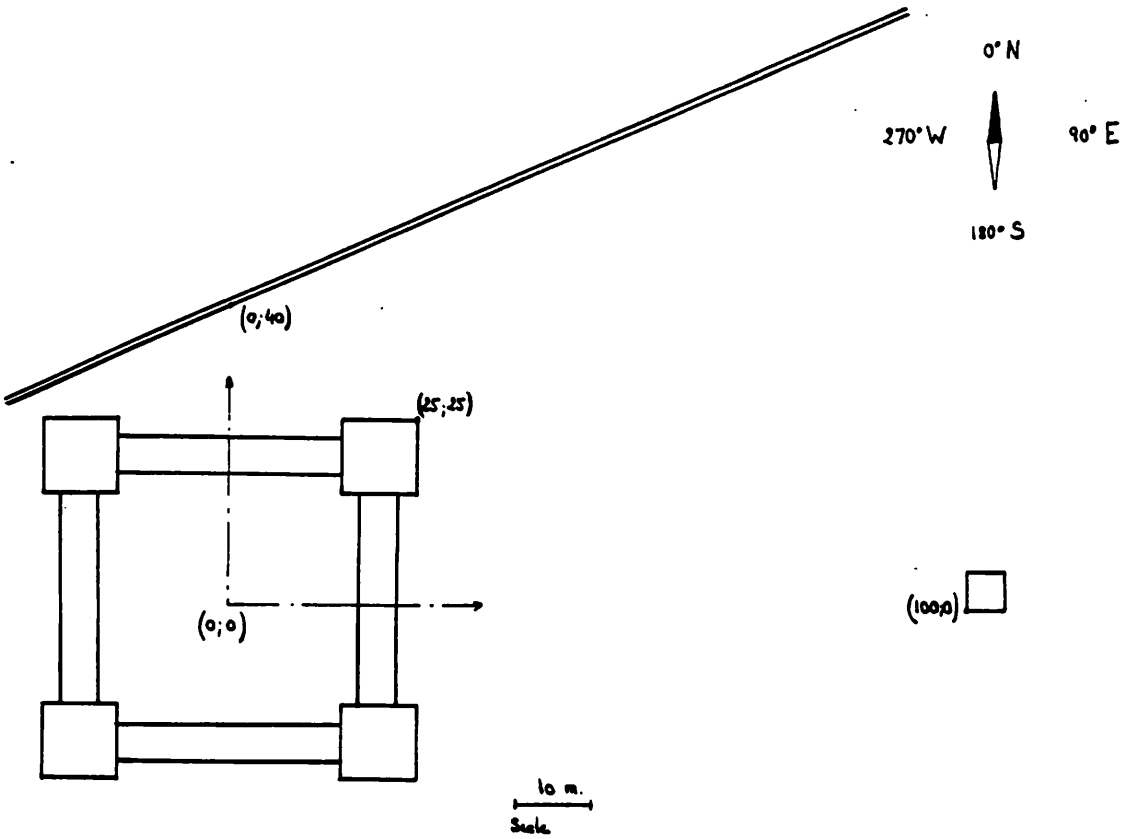


figure 6.5: Map of the environment.

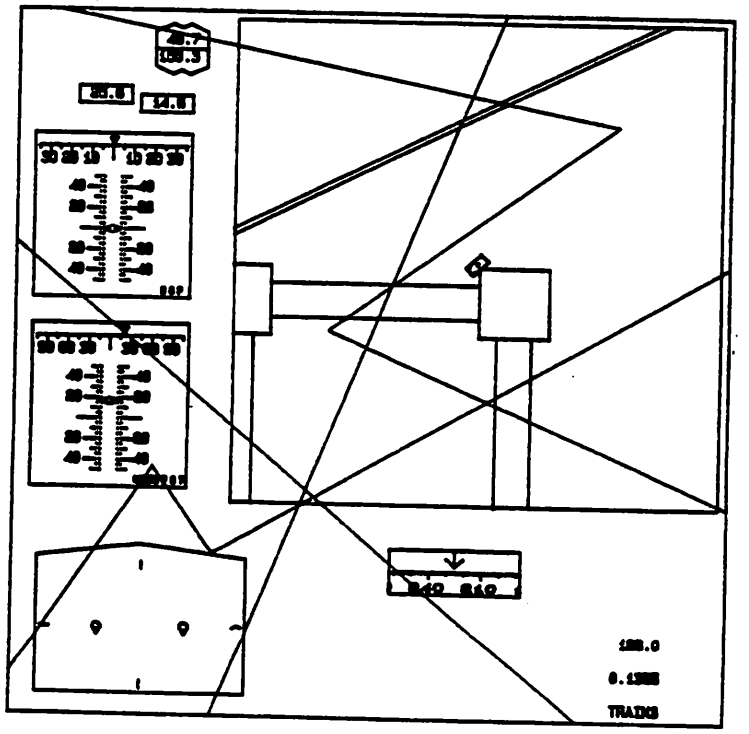


figure 6.6: The ROV hits an element of the environment.

worry about the correspondence of thrusters to joystick. This correction has not been implemeted on the simulator. This alternative might not really be needed when the pilot does not use any top-view and has only his camera information. On another hand, when the camera is panned the correlation between the pan angle and the joystick might again become helpful. Hence the vehicle would move towards the camera view. These considerations are related to Supervisory Control problems and can be tested on the simulator before being tested on a real ROV.

To my knowledge, this specific top-view display (either one of the alternative) is not very often used by ROV systems designers. The simulator takes advantage of the potential of the Megatek screen, nevertheless, a comparable solution could be generated on a microcomputer graphic display. (For example, there exists a flight-simulator on a personal computer which reproduces the same effect for its radar view.) Besides, the position of a vehicle in the water can be measured nowadays and the position of fixed elements in the "world" of the ROV (offshore structure, pipe-line, well-head, template, etc..) can be located with precision.

Finally, the reason for having that top-view in the simulator was not to reflect reality completely, but to compensate for the loss of information due to the 0.5-1.2 s of delay and the low frame-rate of the "3D" display. (The frame rate is determined by the time to generate the next image.) For experimental purposes, the top-view can be turned off or on at the keyboard.

6.3 The link module.

As explained in the previous pages, this module guides all the operations of the simulator. Except for the dynamic module which runs by itself, the others modules are sequenced, through event flags, by the link module.

This module takes care of the indicator displays, reads the joysticks the various potentiometers, and the keyboard.

The operator can select position, depth, or heading "control". Their respective effects are to stabilize the corresponding motions, but they are false controls since the program assigns numerically a value to the associated variable, without using any thrusters. A master's thesis in the MMSL is actually being done on the design of a "real" heading and depth control. Its implementation on the simulator has just begun. After being tested the control will then be installed on the Lab vehicle. The link module will be used to simulate the measurement available on an ROV, with or without noise. The thrusters will be set according to the controller.

Routines to collect data+ can be implemented in the different modules. However the best place is the link module, which is in direct relation with the operator through the keyboard and the analogic inputs. Hence, recording can be started or stopped easily for experiments (Chapter 7).

Modifications of the indicator displays can be done through the link module. Graphic calls are grouped in a specific subroutine and set up to allow any change..

The listings of the programs which constitute the link module are available in Appendix B.

CHAPTER 7

ILLUSTRATIVE RESULTS

7.1 Introduction.

Piloting an ROV is not an easy task. The pilot and his ROV have always to struggle with the current and tether reactions. The motion of the vehicle demands a lot of pilot experience to be predicted, even roughly, and it is said (ref. 4), that only one third of the professional pilots can really do a fair job.

Several people in the Lab including myself tried to "fly" the simulator, and previour observations correlated with the remarks just made above. We could command roughly the motion of the vehicle with the thrusters, for instance to go from one point to another, but a more precise task such as following a pipe-line or hovering at a fixed point in order to manipulate a valve with an arm, became much more complex.

It was thus interesting to see what were some different factors which were important in the human control of an ROV.

Two kinds of experiments were conductedinformally, both directly related to the two problems described previously:

- Following a pipe-line
- Staying at a fixed position.

For these experiments several conditions were tested, beginning with a simple case without current and cable, then only a current, and finally a current and a cable. The importance of the top-view display in the

simulator was evaluated. Several alternatives were also tested such as the help that auto-depth provided to the pilot, who then does not have to worry about this motion.

The subjects of the experiments were myself (since I had the most experience on the simulator!) and a novice subject (in order to learn about initial experience in ROV piloting

These experiments do not have the pretension to be yield statistical significance but only to illustrate some studies which could be done on the simulator.

7.1.a Pipe-line following.

In this experiment the pilot had to track a pipe. The different operating conditions were the following:

- Top-view and camera and:
  - no current, no cable
  - current of 1m/s from the south, no cable
  - current of 0.75 m/s from the south, a cable (characteristics as presented in Chapter 3), the ship stays nearly at the vertical of the vehicle, the depth is 100 meters.
- Only the "camera" and the same conditions as before.

The acquisition of the data was allowed to start at will, through the link module. These data consisted of the x,y,z, the heading of the vehicle, and the time. They were utilized after the run to give a sampled top-view motion as showed in figure 7.1.

7.1.b Hovering.

The previous conditions were also applied for this test. The same data were collected but were treated differently. The hovering situation is presented in figure 7.2. The goal was to fit the two triangles, the one on the ROV and the one in the cube. A complete fit occured when the triangle overlapped and when the altitude was 2.5 meters. The pilots were asked to stay outside the cube. From the data we can get the variation of the altitude, the variation of the (horizontal) distance between the triangles, and the variation of the heading. When the depth control was selected, only the distance was computed, since in this case the pilot is able to limit his heading correctly. Figure 3 is an example of the curves obtained.

When the vehicle stepped into the cube, the distance between the two triangles (which is > 0) was multiplied by -1 to signal the collisions on the curves (fig.7.3). For the camera test, a spot of 25cm $\times$ 25cm was displayed on one face of the cube, at the location were the triangle was supposed to be. This spot was the target of the pilot, who had to hover as close as possible to it.

7.2 Influence of the auto-depth control.

It is obvious that the number of parameters influencing the behavior of the pilot and the ROV are numerous. In the following figures (4, 5, 6, 7) we see the importance of overloading when the pilots had to take care

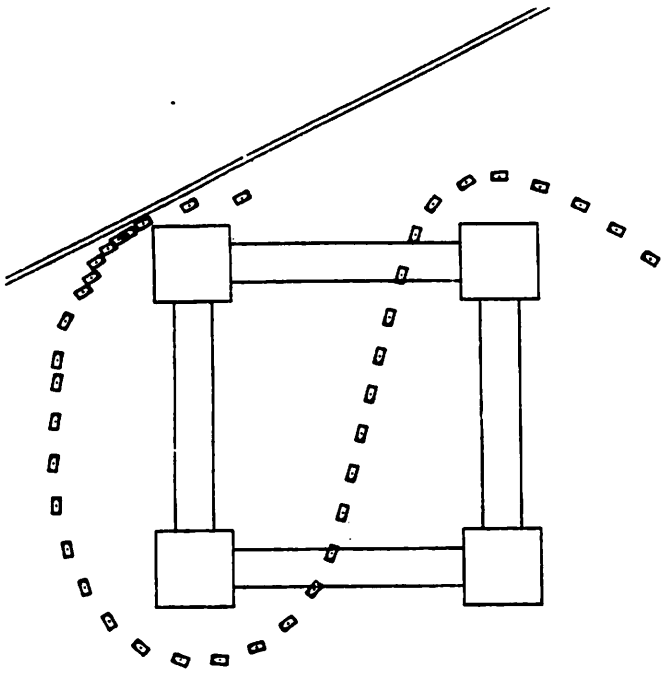


fig. 7.1: Path reconstitution.

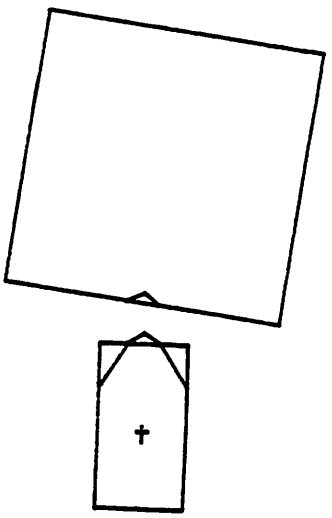


fig 7.2: Hovering situation.

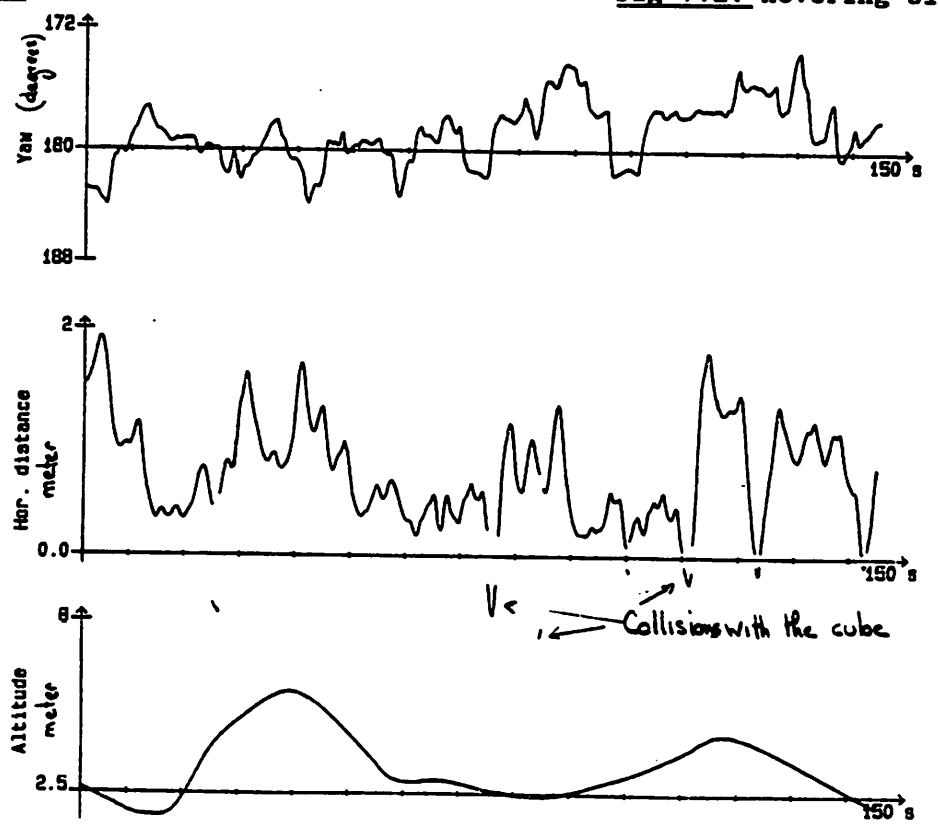


figure 7.3: The different curves obtained from an hovering situation.

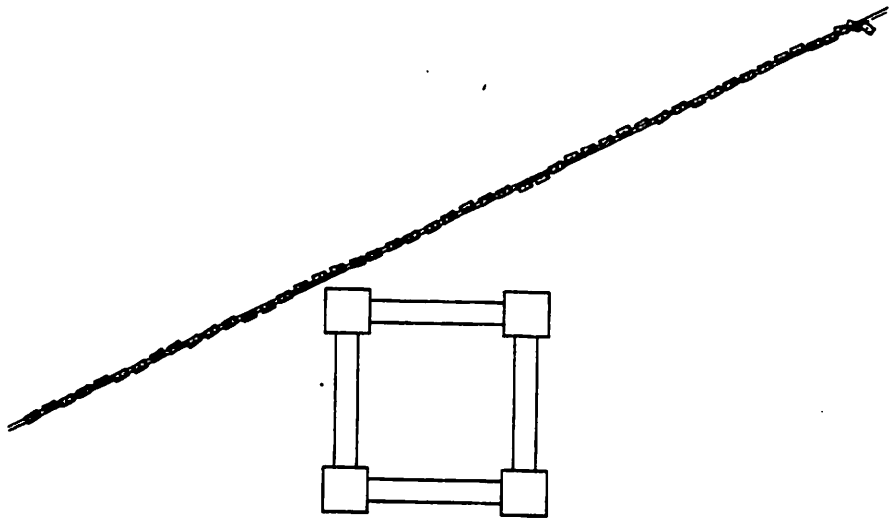


fig. 7.4.a:

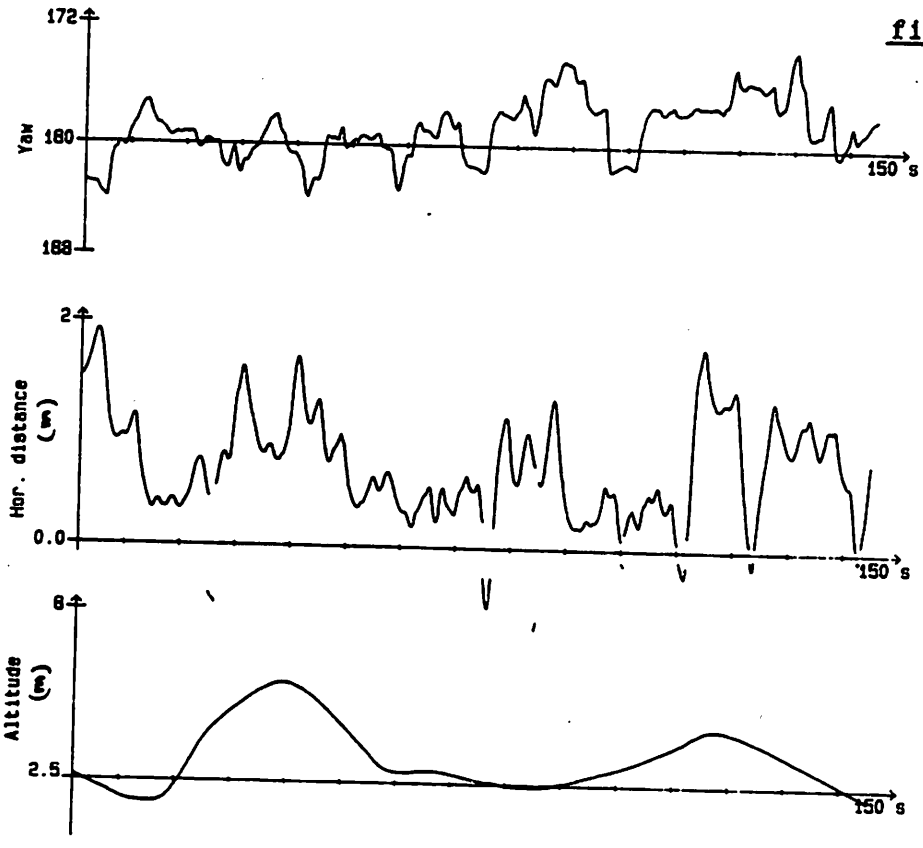


figure 7.4.b: No current, top-view, no control.

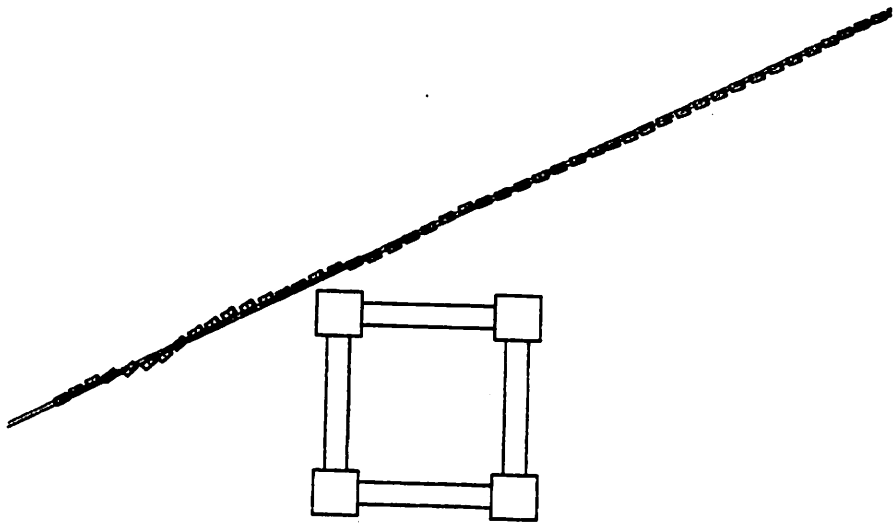


fig. 7.5.a:

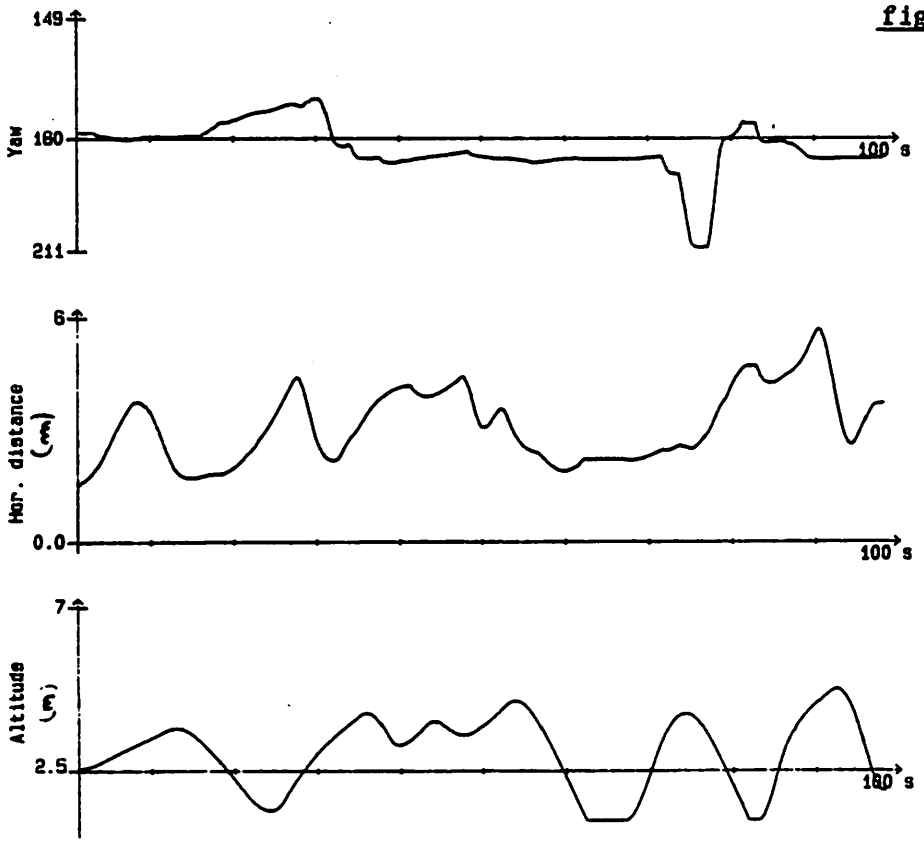


figure 7.5.b: No current, no top-view (only camera), no control.

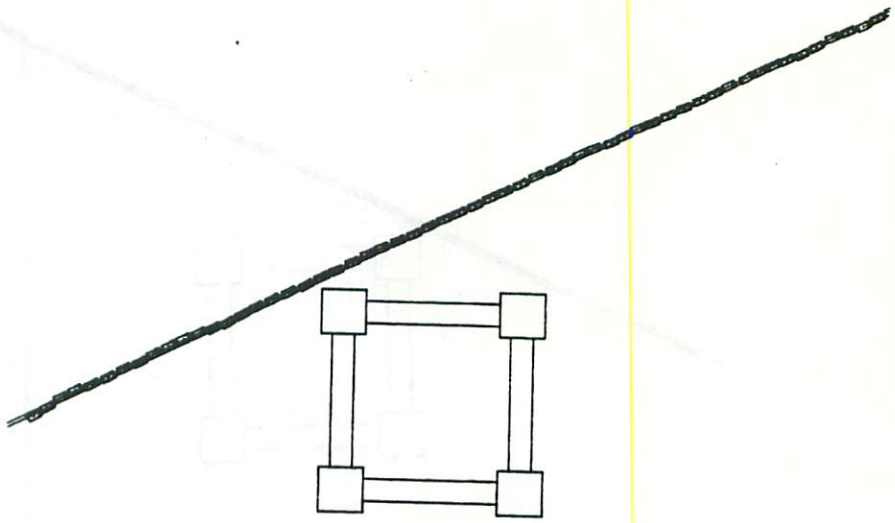


fig. 7.6.a:

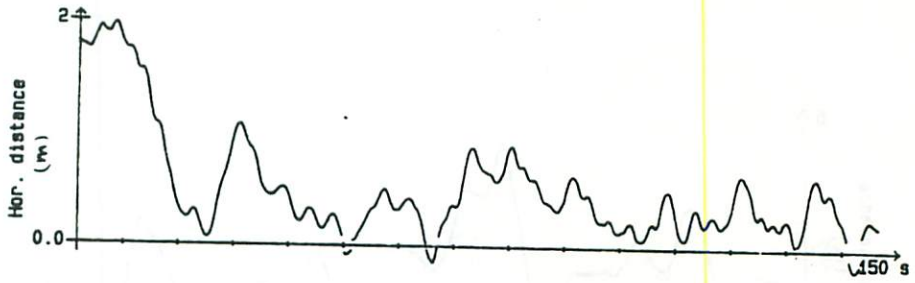


figure 7.6.b: No current, top-view, depth control

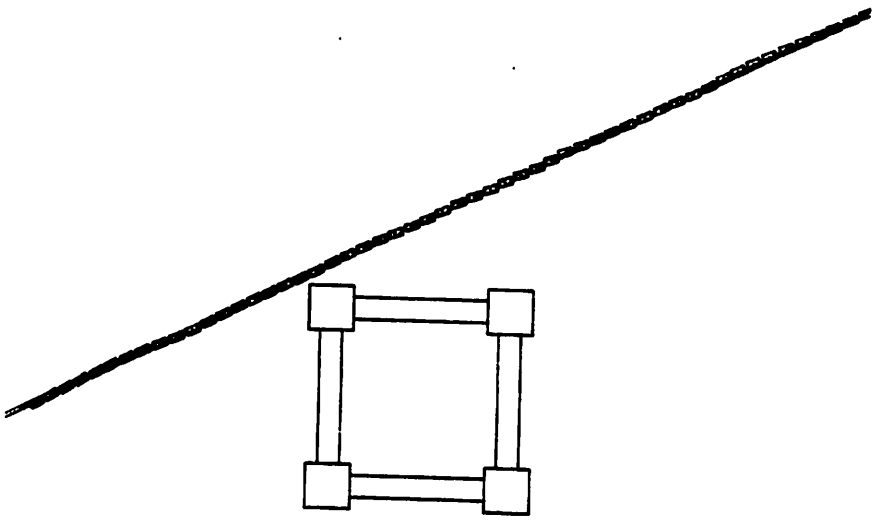


fig. 7.7.a:

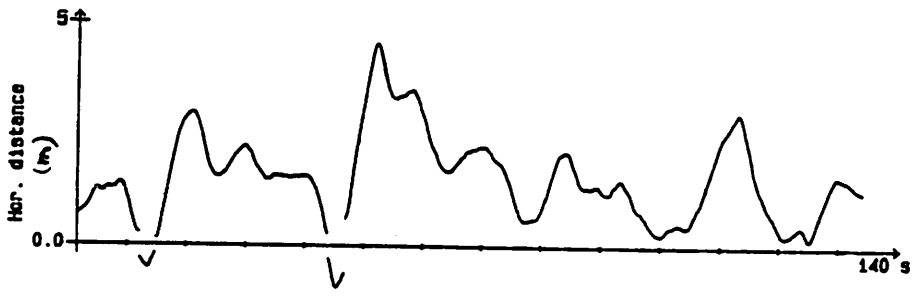


figure 7.7.b: No current, no top-view, depth control.

of depth. The variation of the distance is much larger without control than with, though the control does not have any direct effect on it.

Hence, the experiments thereafter were conducted with the auto-depth control. This control simulation, as explained in Chapter 6 does not control by the means of the thrusters but simply assigns numerically a value to the depth (i.e. perfect depth control). As a consequence, a real control will perform less well, even if this mode of control is one of the easiest for a ROV.

7.3 Influence of the camera.

Figures 4, 6, 8 represent piloting with the top-view, whereas figures 5, 7, 9 are obtained with only the camera display. The environment for that experiment was reduced to only the pipe and the cube, allowing an increase in the display rate of the link module frequency (4 hz). Flying the vehicle is obviously easier with the top-view than the camera only. Nevertheless, the camera was somewhat helpful in limiting the depth variation (without auto-depth). In order to help the pilot to stabilize his vehicle to a given depth without auto-depth, some other form of depth indicator should be designed.

7.4 Influence of only the current.

Figures 8 are the result of a current, and should be compared to figures 6. The average distances do not change drastically. Nevertheless, various situations with transverse current might change the effects on hovering. A more difficult situation would occur with an unsteady current.

This page is intentionally blank.

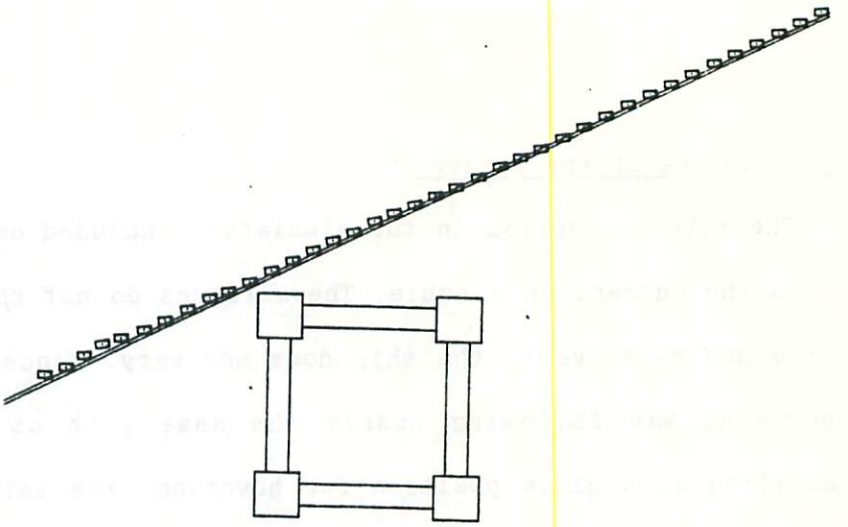


fig. 7.9.a:

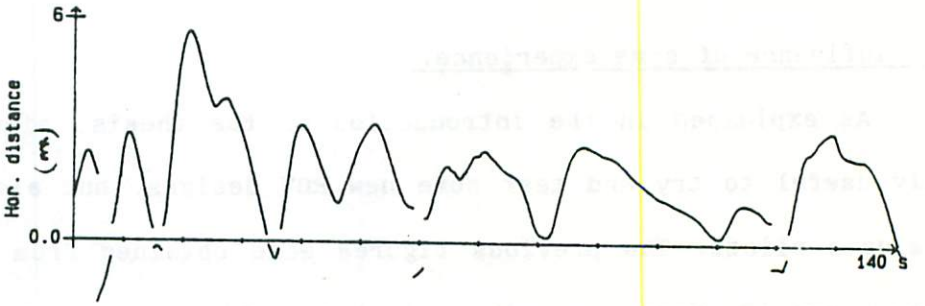


figure 7.9.b: Current, no top-view (only camera), depth control.

7.5 Influence of the tether.

The tether modelled in the simulator, included only the static forces due to the current on a cable. These forces do not change if the position of the ROV relative to the ship does not vary. Since the ship during the experiment was following nearly the same path as the vehicle or was stabilized at a given position for hovering, the tether forces were only steady additional forces applied to a point different for the rotation center of the vehicle. Thus the pilot had to compensate for a yaw moment and an additional translation force, which stayed roughly parallel to the current.

For real operation, the tether interactions with the ROV are more complicated, and will involve, for instance, dynamic turbulent effects.

Figures 10 describe some results obtained for the simulator tests with a tether in a current. Error magnitudes are larger than in the previous situations.

7.6 Influence of some experience.

As explained in the introduction of the thesis, simulators are not only useful to try and test some new ROV designs, but also to train and evaluate pilots. The previous figures were obtained from a subject with some hours of practice on that simulator. Figures 11, 12 are the results from a subject with only a few minutes of practice.

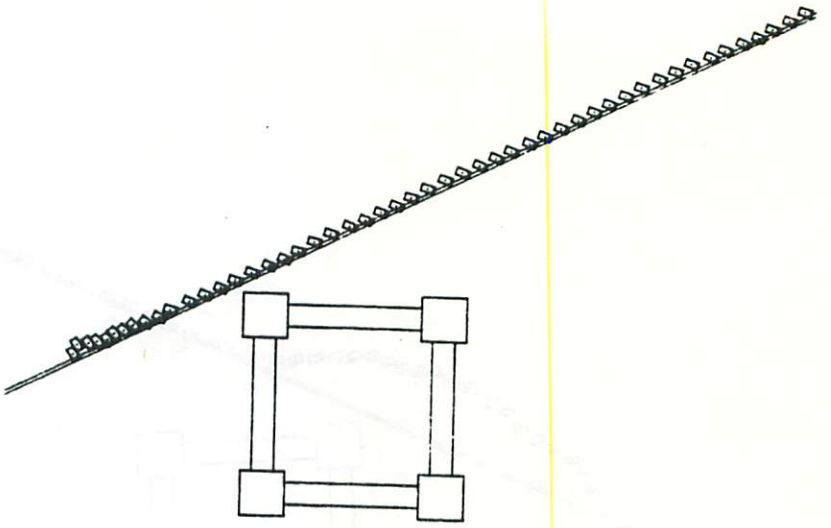


fig. 7.10.a:

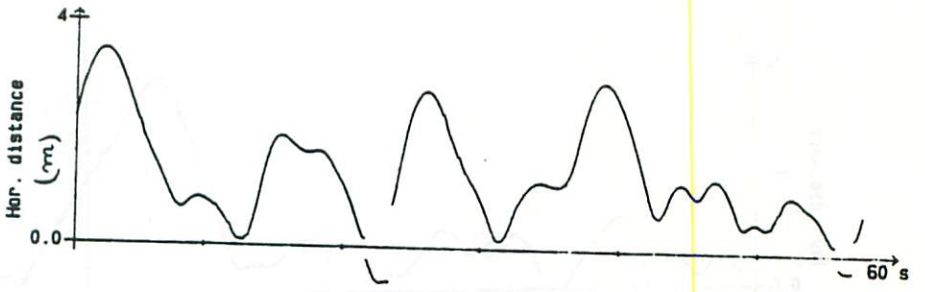


figure 7.10.b: Current and cable, top-view, depth control

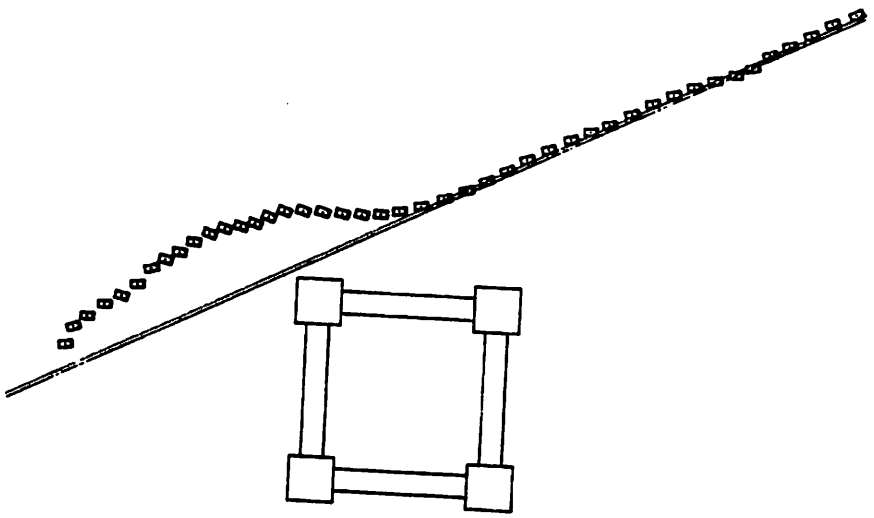


figure 7.11.a:

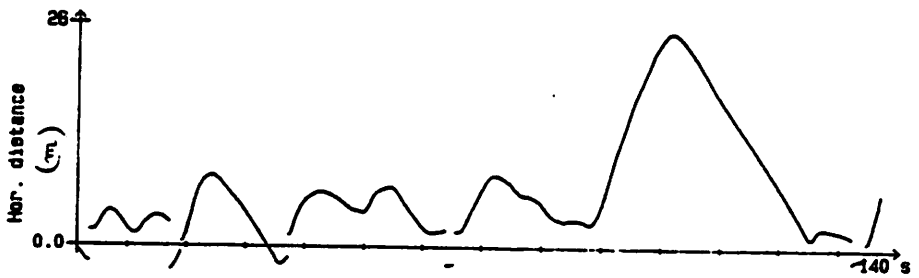


figure 7.11.b: Current, camera, depth control, "non experienced" subject

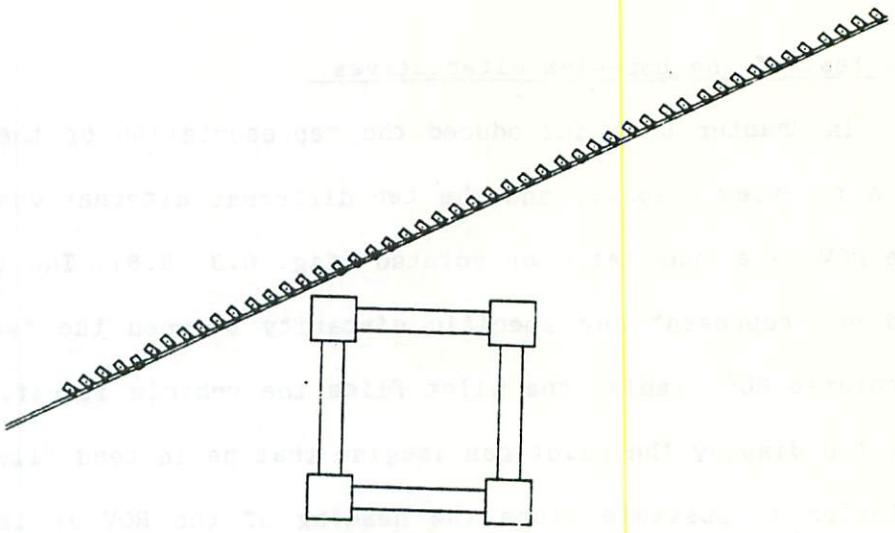


figure 7.12.a:

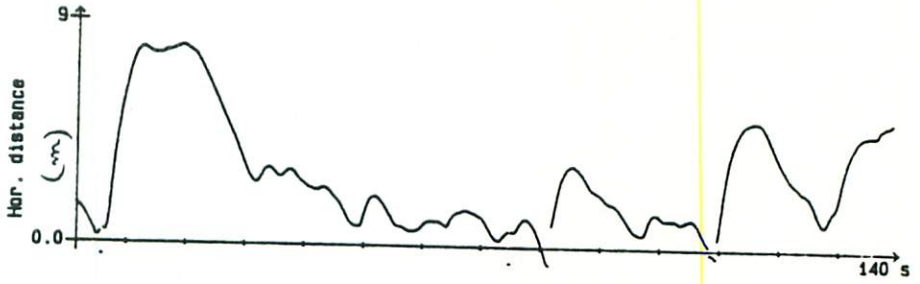


figure 7.12.b: Tether, top-view, depth control, "non experienced" subject

7.7 Test of the top-view alternatives.

In Chapter 6, I introduced the representation of the ROV environment as a top-view display, and the two different alternatives in representing the ROV, i.e. unrotated or rotated (fig. 6.3, 6.6). The test for hovering did not represent any specific disparity between the two forms. For the unrotated ROV display the pilot flies the vehicle itself, whereas for the rotated display the pilot can imagine that he instead flies the cube. This solution is possible since the heading of the ROV is 180° and thus the image is simply reversed.

The results for the pipe-line tracking were more striking. Figure 13 is the result of a trajectory from the structure to the NE corner of the environment, whereas figure 14 is the return trip, where the direction of the forces from the thrusters are at 135° from their corresponding joystick commands. This figures show that the correlation of thruster to joystick is important. (Note that with the only camera, no difference should be noticed.) Figures 15 and 16 should be compared to figure 6.a.

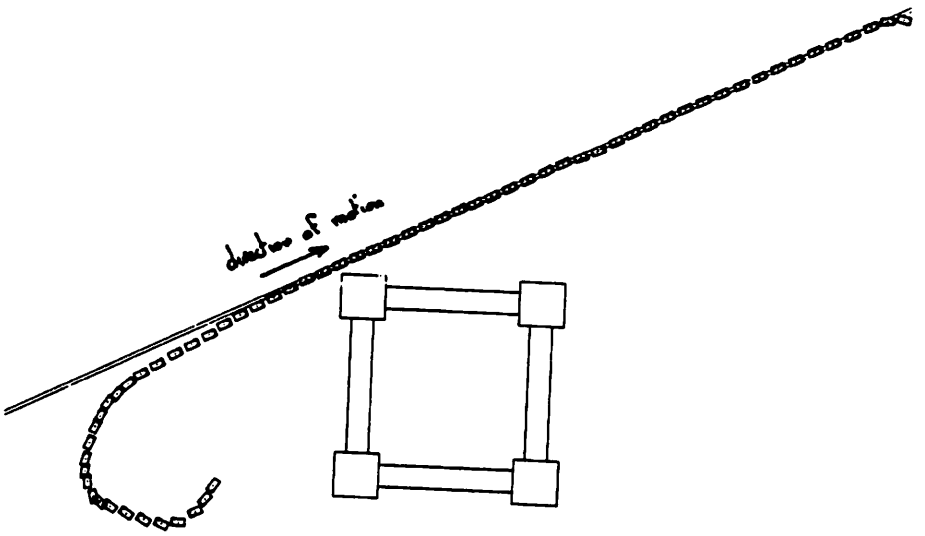


figure 7.13:

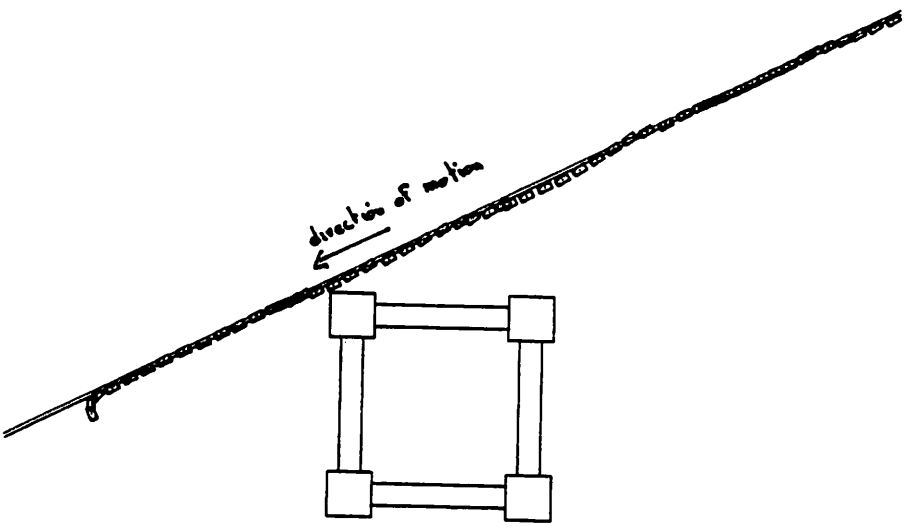


figure 7.14:

CHAPTER 8

SUGGESTION FOR A FUTURE CONFIGURATION USING A MULTI-MICROPROCESSOR SYSTEM

A modular program is, in general, very interesting in regard to simpler modification, or test, or complement, of one of its part. This is the main reason for the modular concept of the simulator I developed.

Each module has its own data and variables and communicates with the others through a common area in the memory and event flags.

Obviously, such programs can run on small or large computers with a multi-tasking operating system. The solution however is not optimal since we do not use all the facilities provided by this solution, and the systems lose time to switch between the different modules. Besides, "improvement" of the program by adding modules degrades also its real-time capabilities.

For a much lower ratio of cost/efficiency, a modular simulator can be implemented on a multi-microprocessor systems. Each module is associated with a board equipped with its own microprocessor and arithmetic coprocessor when needed, and local memory. Each board is then linked via a bus system to the others and a common memory. One improves the simulator by simply adding new boards corresponding to new modules, or by improving a specific board by changing its components.

Each board can be completely independent of the others, and have different characteristics or components, whenever the link to the bus is respected.

Such a simulator could be created, for instance, with a Multibus 2 (32 bits path bus from Intel), using the new 32 bit processor and coprocessor 80286 and 80287 from Intel, or the 680xx (68000, 68010, 68020) and 68881 from Motorola. Complete boards with these components are beginning to be available on the market. These units exist with different characteristics, mainly their operating frequency, which can be chosen according to the need of the specific modules. These processors (the 80286 (10 MHz) or the 68000 (16 MHz)) have higher performance than the DEC VAX 11/780 operating with the resident operating system VMS. The lower versions (80286 (8 MHz), 68000 (8MHz)) still perform reasonably well compared to the 11/780.

CHAPTER 9

CONCLUSIONS

The objectives of this study have been attained, namely, the program is able to simulate fairly accurately the dynamics of an ROV, can provide a 3D display of the scenery, and can compute an approximation of the actions of the umbilical tether.

It is used, just now, for the testing of the depth and heading control, which soon will be implemented on the MMSL vehicle. Some illustrative experiments show that some research can be done on that simulator to determine how to improve piloting of the vehicle by modifying indicators or commands. Training is also possible.

This study can be improved in different ways. The first task should be to implement the simulator on a multi-microprocessor computer system. Then the display can be modified to give more detail. The environment emulator should detect the collisions and take them into account in the dynamics. The current should also vary. Finally, some dynamic effects of the tether might be included.

REFERENCES

[1] ABKOWITZ M.A., "Stability and motion control of ocean vehicle." MIT press, Canbridge, 1969

[2] BLAGOVESHCHENSKY S.N., "Theory of ship motions.", Dover publication, New-York, 1962

[3] BLIEK A., "Dynamic Analysis of single-span Cables.", Ph.D. Thesis, MIT Ocean Eng. Dpt., 1984

[4] EVENSEN G., "ROV training is imperative", Conference Proceedings, ROV 83, San Diego 1983

[5] EVERY M.J. & DAVIES M.E., "Prediction of the drag and performance of umbilical cables", Conference proceedings, ROV 84, San Diego, 1984

[6] GERTLES M. & HAGEN G.R., "Standard equations of motion for submarine simulation.", NSDRC report n° 2510, Washington DC, 1967

[7] HAMMOND J.T., "Development of a six degree of freedom motion simulation model for use in submarine design analysis", M.S. Thesis, MIT Ocean Eng. Dpt., 1971

[8] JOHANSSON P., "A finite elememt model for dynamic analysis of mooring cables", Ph.D. thesis, MIT Ocean Eng. Dpt., 1976

[9] ODAHARA T. "Experiments in Supervisory Control of a computerized vehicle for inspecting curved surfaces.", M.S. Thesis, MIT Mech.Eng. Dpt., 1980.

[10] RUSSEL G.T. & DAVID D.M. & WELLS S.C., "Autonomous submersible systems", Proceedings on Remotely Operated Vehicles, U.S. department of commerce, NOAA, 1983

[11] SHERIDAN T.B. & FERRELL W.R., "Supervisory Control for remote manipulation.", IEEE Spectrum, 4-10, October, 81-88.

[12] TACKABERY R. & LAGUE T., "Ocean currents and ROV's", conference proceedings, ROV 83, San Diego 1983

[13] TRIANTAFYLLOU M.S. "The dynamics of taut inclined cables.", QJ Mech. Appl. Math., Vol 37, Pt 3, 1984

[14] ROV training center , Offshore, April 85.

[15] "MIT Underwater Vehicle Research", Sea Grant College Program,

Opportunity Brief #36.

[16] SCHEMPF H.,"Hydrodynamic coefficients for a ROV modelled as a cylinder.", Report written for Dr Dana YOERGER of Woods-Hole Institute, 1984

BIBLIOGRAPHY

HARRINGTON S. "Computer Graphics, a programming approach.", McGraw-Hill, 1983.

KAZEROONI H. "Computer simulation and control of underwater vehicle", Sea Grant College Program, Report n° 82-19

THOMASSON P.G. "Simulators for use as design aids and for operator training.", Proceedings on Remotely Operated Vehicles, U.S. department of commerce, NOAA, 1983

LEWIS D.J. & LIPSCOMBE J.M. & THOMASSON P.G., "The simulation of remotely operated underwater vehicles.",Conference proceedings, ROV 84, San Diego, 1984

APPENDIX A

Complements on the dynamics.

Part1.

Using the notation from chapter 2.

$$\Sigma(\text{external forces on body}) = m\frac{d}{dt}(\hat{U} + \hat{\Omega}\times\hat{R}_G)$$

The system is supposed Gallilean and so the coriollis effect is neglected.

$$\frac{d}{dt}(\hat{U} + \hat{\Omega}\times\hat{R}_G) = \frac{d}{dt}\hat{U} + \frac{d}{dt}(\hat{\Omega}\times\hat{R}_G)$$

$$\frac{d}{dt}\hat{U} = \frac{d}{dt}u\hat{i} + \frac{d}{dt}v\hat{j} + \frac{d}{dt}w\hat{k} + u\frac{d}{dt}\hat{i} + v\frac{d}{dt}\hat{j} + w\frac{d}{dt}\hat{k}$$

$$\frac{d}{dt}\hat{i} = r\hat{j} - q\hat{k} \quad ; \quad \frac{d}{dt}\hat{j} = -r\hat{i} + p\hat{k} \quad ; \quad \frac{d}{dt}\hat{k} = q\hat{i} - p\hat{j}$$

$$\frac{d}{dt}\hat{U} = (\dot{u} - vr + wq)\hat{i} + (\dot{v} + ur - wp)\hat{j} + (\dot{w} - uq + vp)\hat{k}$$

$$\frac{d}{dt}(\hat{\Omega}\times\hat{R}_G) = \frac{d}{dt}\hat{\Omega}\times\hat{R}_G + \hat{\Omega}\times\frac{d}{dt}\hat{R}_G$$

$$\frac{d}{dt}\hat{\Omega} = \dot{p}\hat{i} + p\frac{d}{dt}\hat{i} + \dot{q}\hat{j} + q\frac{d}{dt}\hat{j} + \dot{r}\hat{k} + r\frac{d}{dt}\hat{k}$$
$$= (\dot{p} - qr + qr)\hat{i} + (\dot{q} + pr - rp)\hat{j} + (\dot{r} - pq + qp)\hat{k}$$
$$= \dot{p}\hat{i} + \dot{q}\hat{j} + \dot{r}\hat{k}$$

$$\frac{d}{dt}\hat{R}_G = x_G\cdot\frac{d}{dt}\hat{i} + y_G\cdot\frac{d}{dt}\hat{j} + z_G\cdot\frac{d}{dt}\hat{k}$$
$$= (-y_Gr + z_Gq)\hat{i} + (x_Gr - z_Gp)\hat{j} + (-x_Gq + y_Gp)\hat{k}$$

$$\frac{d}{dt}(\hat{\Omega}\times R_G) = \begin{vmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{vmatrix} \times \begin{vmatrix}x_G\\ y_G\\ z_G\end{vmatrix} + \begin{vmatrix}p\\ q\\ r\end{vmatrix} \times \begin{vmatrix}-y_G r + z_G q\\ x_G r - z_G p\\ x_G q + y_G p\end{vmatrix}$$

$$= [-x_G(q^2+r^2) + y_G(pq-\dot{r}) + z_G(pr+\dot{q})]\hat{i}$$

$$+ [\,x_G(pq+\dot{r}) - y_G(p^2+r^2) + z_G(qr-\dot{p})]\hat{j}$$

$$+ [\,x_G(pr-\dot{q}) + y_G(qr+\dot{p}) - z_G(p^2+q^2)]\hat{k}$$

and finally

$$= m\{[(\dot{u} - vr + wq) - x_G(q^2+r^2) + y_G(pq-\dot{r}) + z_G(pr+\dot{q})]\hat{i}$$

$$+ \quad [(\dot{v} + ur - wp) + x_G(pq+\dot{r}) - y_G(p^2+r^2) + z_G(qr-\dot{p})]\hat{j}$$

$$+ \quad [(\dot{w} - uq + vp) + x_G(pr-\dot{q}) + y_G(qr+\dot{p}) - z_G(p^2+q^2)]\hat{k}\}$$

$$\Sigma(\text{moments in G}) = \frac{d}{dt}(I'_x p\hat{i} + I'_y q\hat{j} + I'_z r\hat{k}) =$$

$$\text{with } I'_x = I_x - m(y_G^2 + z_G^2)$$

$$I'_y = I_y - m(x_G^2 + z_G^2)$$

$$I'_z = I_z - m(x_G^2 + y_G^2)$$

and $\Sigma(\text{moments in P}) = \Sigma(\text{moments in G}) + m\hat{R}_G \times \frac{d}{dt}(\hat{U}+\hat{\Omega}\times\hat{R}_G)$

$$\frac{d}{dt}(I'_x p\hat{i} + I'_y q\hat{j} + I'_z r\hat{k}) = [I'_x\dot{p} - I'_y qr + I'_z rq)]\hat{i}$$

$$+ [I'_y\dot{q} + I'_x pr - I'_z pr)]\hat{j}$$

$$+ [I'_z\dot{r} - I'_x pq + I'_y pq)]\hat{k}$$

$$m\hat{R}_G \times \frac{d}{dt}(\hat{U}+\hat{\Omega}\times\hat{R}_G) = m.\begin{vmatrix} x_G & \times & [(\dot{u}-vr+wq) - x_G(q^2+r^2) + \\ & & \quad y_G(pq-\dot{r})+z_G(pr+\dot{q})] \\ x_G & & [(\dot{w}-uq+vp) + x_G(pr-\dot{q}) + \\ & & \quad y_G(qr+\dot{p}) - z_G(p^2+q^2)] \\ x_G & & [(\dot{v}+ur-wp) + x_G(pq+\dot{r}) - \\ & & \quad y_G(p^2+r^2) + z_G(qr-\dot{p})] \end{vmatrix}$$

Σ(moments in P) =

$$\{I_x\dot{p} + (I_z-I_y)qr + m[y_G(\dot{w}-uq+vp) - z_G(\dot{v}+ur-wp) + x_Gy_G(pr-\dot{q}) - x_Gz_G(pq+\dot{v}) + y_Gz_G(r^2-q^2)]\}.\hat{i}$$

$$+ \{I_y\dot{q} + (I_x-I_z)rp + m[z_G(\dot{u}+qw-rv) - x_G(\dot{w}+pv-qv) + y_Gz_G(qp-\dot{r}) - x_Gy_G(qr+\dot{p}) + x_Gz_G(p^2-r^2)]\}.\hat{j}$$

$$+ \{I_z\dot{r} + (I_y-I_x)pq + m[x_G(\dot{v}+ru-pw) - y_G(\dot{u}+qw-rv) + x_Gz_G(rq-\dot{p}) - y_Gz_G(rp+\dot{q}) + x_Gy_G(q^2-p^2)]\}.\hat{k}$$

Part 2.

The transformation matrices are the followings :

$(\hat{I},\hat{J},\hat{K})$ to $(\hat{I}_1,\hat{j}_1,\hat{k}_1)$:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$(\hat{I}_1,\hat{j}_1,\hat{k}_1)$ to $(\hat{i}_2,\hat{j}_2,\hat{k}_2)$:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} c\phi & 0 & -s\phi \\ 0 & 1 & 0 \\ s\phi & 0 & c\phi \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

$(\hat{i}_2, \hat{j}_2, \hat{k}_2)$ to $(\hat{i}, \hat{j}, \hat{k})$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & s\theta \\ 0 & -s\theta & c\theta \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

and combined $(\hat{I}, \hat{J}, \hat{K})$ to $(\hat{i}, \hat{j}, \hat{k})$:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & s\theta \\ 0 & -s\theta & c\theta \end{bmatrix} \begin{bmatrix} c\phi & 0 & -s\phi \\ 0 & 1 & 0 \\ s\phi & 0 & c\phi \end{bmatrix} \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

or

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c\psi c\phi & s\psi c\phi & -s\phi \\ c\psi s\phi s\theta - s\psi c\theta & s\psi s\phi s\theta + c\psi c\theta & c\phi s\theta \\ c\psi s\phi c\theta + s\psi s\theta & s\psi c\theta s\phi - c\psi s\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\hat{\Omega} = \dot{\psi}\hat{K} + \dot{\phi}\hat{j}_1 + \dot{\theta}\hat{i}_2.$$

$$\hat{K} = -s\psi.\hat{i}_2 + c\psi.\hat{k}_2$$

$$\hat{j}_1 = \hat{j}_2$$

$$\hat{i}_2 = \hat{i}$$

$$\hat{j}_2 = c\theta.\hat{j} - s\theta.\hat{k}$$

$$\hat{k}_2 = s\theta.\hat{j} + c\theta.\hat{k}$$

Hence:
$$\begin{cases} p = -\dot{\psi}s\phi + \dot{\theta} \\ q = \dot{\psi}c\phi s\theta + \dot{\phi}c\theta \\ r = \dot{\psi}c\phi c\theta - \dot{\phi}s\theta \end{cases}$$

or :
$$\begin{cases} \dot{\psi} = (q.s\theta = r.c\theta)/c\phi \\ \dot{\phi} = q.c\theta - r.s\theta \\ \dot{\theta} = p + \tan(\psi).(r.c\theta + q.s\theta) \end{cases}$$

APPENDIX B.

Miscellaneous

Color limitation

The Lexidata has usually 256 colors shared in 8 planes. It does not have however, the possibility to erase a specific area of the screen, if this area does not include the top left point of the screen. The only alternative is thus to erase color planes.

The principle of the display is to show on one part of the screen a finished image. Meanwhile the next image is built on another part of the screen. Each image is thus associated with 4 color planes, which gives only 16 colors for each image. By erasing the corresponding planes, I can erase the image being updated.

The listing of the different modules can be obtained from

Professor Thomas B. SHERIDAN

or from the author.

(≈ 3000 lines).