Dog On Lead Minehunting underwater cable simulation

Ing. Claudio Pedrazzi
Riva Calzoni S.p.A.
c/o Technical Computing Center
Via Emilia Ponente 72, 40133 Bologna, Italy
Tel: 051-4130677 Fax: 051-4130655

Introduction

Today's minehunting operations are by far more complex and risky than ever before: this is basically caused by the invention of new types of mines (mobile mines, multi-influenced "smart" mines,...), and a variety of new operational aspects, such as very shallow or deep water capabilities (down to a depth of 300 m). A Mine CounterMeasure (MCM) system based on the coupling of the traditional hull-mounted sonar with a remotely operated Self Propelled Variable Depth Sonar Vehicle (SPSV or SVDS) for detection, classification and disposal of mines ahead of the MCM ship, has proven to be very efficient.

With today's technology, a fully autonomous vehicle without any tether to the mother ship poses formidable problems of power source, data transmission, and reliability: this will be the medium future solution; the current systems of today are cable guided sonar (this configuration is usually called "Dog On Lead minehunting"). There are two ways of operating a minehunting Remotely Operated Vehicle (ROV) carrying sonar: by providing power down the wire or by carrying the power source on-board. Systematic assessment of alternatives, and decisions regarding the powering requirements and operational profiles (depth, reach, speed, max current) are made much easier by the development of a numerical simulation of the whole system.

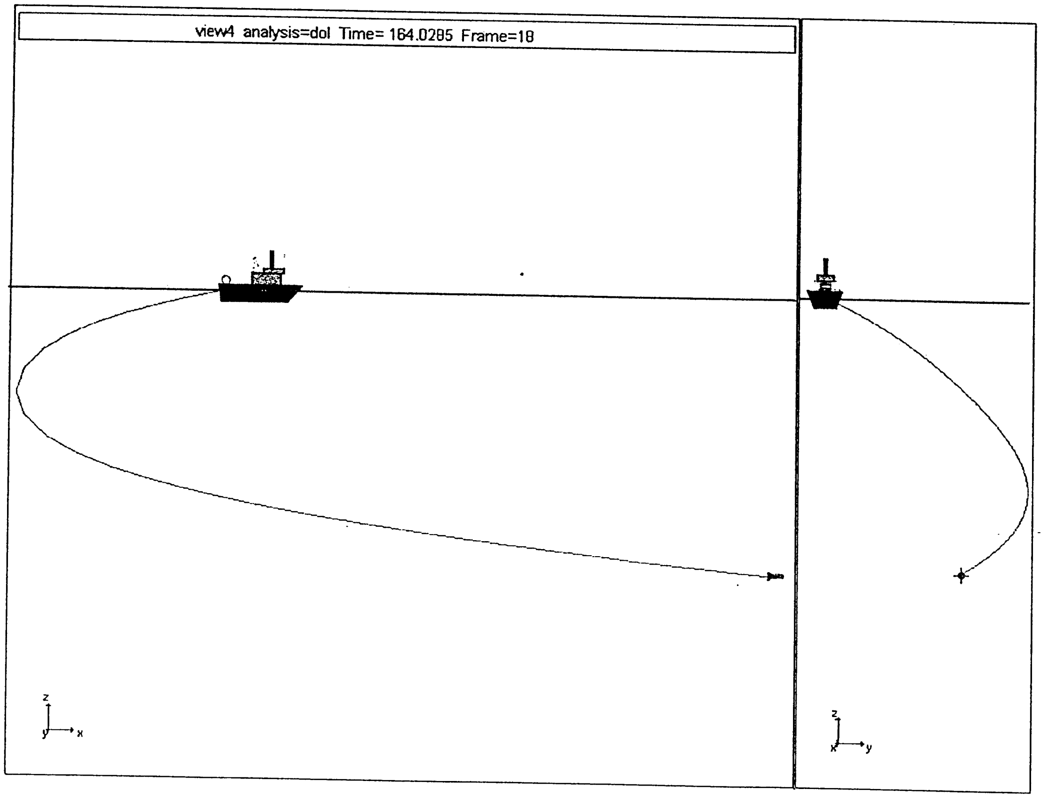

Fig. 1: A typical Dog On Lead minehunting configuration, in presence of a lateral current; ship and ROV are out of scale with respect to real operating depth and reach

Simple elementary fluid-dynamic considerations show the importance of the drag of the umbilical cable in determining the total system performance: for example a small length of cable (diameter 25 mm, length 1 m, dragged sideways) produces the same amount of drag as a "fish" of 0.5 m diameter and a length of 3m. The drag of the cable influences the power requirements for the ROV propulsion, that in turn influences the cable diameter, that finally augments the drag, in a typical non-linear design loop.

The SIREN (System for Immersed ROV Evaluation in Navigation) ADAMS based software package is in the first stages of development, but is already proving its effectiveness in evaluating certain design alternatives (such as the cable spooling strategy, or the deployment manoeuvre). It has been preliminary validated by comparison with the results obtained via a traditional Finite Element Program. The main technical highlights of the model are:

- very extensive use of ADAMS/View "cmd" language, with iterative and looping constructs, used for non-linear equation solution, and for assembly of many identical pieces of cable, each one with its own equations and parts constraints.
- completely parametric model in order to benefit of the ADAMS/View Design Study, DOE and Optimisation Features (for example for a given depth, reach and misalignment there is an optimal length of cable in order to minimise the total drag).
- two different formulations of the hydrodynamic drag force on a cable: Morison law (a simple uncoupled equation) and Nordell and Meggit formula for coupled tangential - normal effect. A logical flag can switch from one to another
- modelization of buoyancy forces on cable and on ROV.
- in one of the execution modes, the ship and the ROV can move on arbitrary 3-D trajectories, while the underwater current is taken into account as a constant speed and orientation uniform motion: in this mode the main unknown are the forces on ship and ROV and the cable tension and shape as a function of time.
- alternatively, the ROV can be left free to move subjected to the forces of its propulsors, given a functions of time, and to the total drag vector generated from the cable: in this mode of execution the main unknown is of course the ROV trajectory (the next step will be the control system for trajectory stabilisation)
- automatic reporting feature (i.e. the generation, at the end of the run, of a complete printable numerical report summarising the main input and output data)

The physical problem

Riva Calzoni S.p.A., a high-technology European concern in the Defence Sector for Navy, Army and Aerospace applications, for more than 25 years has been the design authority for a number of automatic handling systems, like: missiles and other weapons handling systems, helicopter handling systems, and of course Remotely Operated Vehicles (ROV) for undersea mine countermeasure operations. Its past production includes a self-propelled cable guided ROV (called MIN), carrying a camera and a sonar, used by the Italian Navy, capable of operating up to 300 m of depth.

As already pointed out in the introduction, the trend in this field is toward self-propelled cable-powered vehicles capable of operating continuously ahead of the mother ship, and at various depths (for example it can reveal objects under a thermocline layering of the water that cannot be observed by the main hull-mounted sonar). In this context, the Riva Calzoni company is participating to a pilot project, financed by the Italian Navy, for the conceptual design and development of a real prototype of a new SPSV.

In order to understand the physical problems posed by the situation, here are some typical order of magnitude for the dimensioning parameters of such a system [4], [5], [7]:

- reach: 300 m ahead of the ship (slant range)
- depth: 300 m
- cross current: 3 knots
- ship and ROV speed: 10 knots
- cable diameter: 10 mm
- ROV length: 2m

As a matter of fact, the length and diameter of cable mean that the **hydrodynamic drag on the cable is the prevailing effect** at normal operating depths, and so there is no need (at least initially) to have an exact estimation of the drag forces on the ROV (which would require Computational Fluid Dynamics software). We can see the typical pattern of a geometrically non linear structural problem: force depends on deformed shape, that depends on the force itself ... (think for example to the determination of the shape of a sail under the pressure of the wind); the final resultant force determines the power requirements for the vehicle propulsion system, which in turn impacts on the cable cross-section, re-iterating the loop again because this changes the drag on the cable.

The drag force on a simple body such as an immersed cylinder, when the flow is turbulent (Reynolds number is about 10000 for typical operating conditions) is available from different sources [8], [9], and is generally expressed in function of the free-stream relative velocity, acceleration, and of the angle formed between the cable axis and the relative velocity vector; formulas differ slightly, and will be discussed in the following chapters.

The underwater **current** must of course be taken into account: it can increase considerably the total net force required to drive the ROV on a predetermined path; in general the current can change direction and speed with depth, reducing to zero on the sea bottom. Another phenomenon that can influence the overall system behaviour is the **buoyancy** effect: even if the cable (and the ROV) are designed to be neutral, changes in water temperature and salinity in different seas will change the water density, creating a net positive or negative buoyancy force, that on hundreds of meters of cable has to be considered.

Last but not least, during operations, a software controlled "intelligent winch" can **spool** in and out the cable, for example for deployment, for minimisation of the power required to keep the ROV going, or finally to compensate for wave motion, avoiding strong accelerations on one cable end.

There should be no need to explain the great advantages of a numerical simulation of the problem: design alternatives can be tested very early, power requirements can be assessed, different types of manoeuvring strategies can be evaluated, and so on; the literature on the subject shows many different approaches, ranging from the theoretical to the real-time simulator [1], [2], [3], [6]. Due to the limited time and resources initially allocated to the project, we decided to build our model based on some kind of commercially available simulation software.

The model had to address the following main problem areas:

- forces on both ends of cable, and shape of it, in any stationary situation of uniform forward motion (both for a towed and for a self-propelled body) with current
- forces as a function of time during certain typical manoeuvres (e.g. various turns, change of depth,...) assuming that the ROV and the ship follow a given predetermined 3-D trajectory (the "ship" could also be a submarine: the model should make no unnecessary assumptions)
- trajectory of a free ROV with different hypothesis of propulsion and direction control means (one or more thrusters, fins, ...) subjected to the drag forces on its body and of the trailing cable
- development and test of an automatic stabilisation and control system, acting on the chosen propulsion and direction devices in order to follow a programmed exploration path with the maximum insensibility to external random disturbances (current fluctuations, bottom effect, wave motions,...); it is very important to understand that the problem posed by this development can be very different from the standard aerodynamic/fluid-dynamic vehicle stabilisation, due to the dominant effect of the cable.

The choice: FEM or MSS ?

At first sight, the problem, although highly **non-linear** (geometrically), seems to be a good application for Finite Element Method (FEM) software codes: some of these contain appropriate elements, whose formulation already contain the equations of fluid-dynamic drag, freeing the user from the need to code them in the model. The only real work to be done is a description of the geometry of the problem, and a setup of the correct solution parameters (non linear convergence features, integrator parameters when studying the dynamic cases).

As a matter of fact, the ANSYS code, routinely used in Riva Calzoni, features an "**immersed pipe**" element [9] that suited our needs, and during an intermediate phase of the project we did develop a parallel homologue FEM model of the system, that is still kept for validation purposes (when applicable, results are in very good agreement: maximum displacement error is 5%, and in general much less than this).

Nevertheless, even if for static simulations the computer times were comparable, for dynamic analyses the CPU time requirements for good convergence of the integrator were, at best, 13 (thirteen!) times greater than those of a comparable ADAMS model. This could probably be optimised and anyway overcome with a "brute force" approach, but:

- the element formulation, even if very complete and well documented, is hard-coded in the FEM software, and does not lend itself to user modification and experimentation: while in the structural domain there is no need of this, the drag on a cable is somewhat dependent on the source, being mainly the result of heuristic approximations and interpolation of experiments.
- post-processing capabilities are limited to "deformed shape" plots, and a complex representation of the "virtual reality" type is unthinkable
- we could not see any way of introducing the concept of spooling (variable length of cable) in a FEM model
- it seemed very difficult to open the simulation toward the domain of automatic control systems, that is one of our initial requirements

So we decided to stick to ADAMS, accepting the challenge to re-create with a Multibody Simulation System (MSS) some kind of Finite Element architecture: in fact, one could not avoid the task of **discretizing** the cable in a relevant number of "segments", each one with the same identical formulation of parts, joints, forces and constraints. What came out could be appropriately called an hybrid Multibody - Finite Element model: tests have been carried out to ensure that the maximum discretization affordable in ADAMS (of the order of 50 segments) is well inside the asymptotic convergence region.

The ADAMS model

The ADAMS model, completely written at the level of macro and command files for ADAMS/View, with no use of user-written FORTRAN routines, has a certain number of unusual features, that we are going to point out in the following paragraphs. It also can be of some interest the fact that, in order to accommodate the large number of parts and force functions, even with the more streamlined formulation, we were forced to increase some of the ADAMS/Solver memory allocation parameters available in the "uconfg.f" routine, re-linking then a new Solver executable.

Because of the many choices that had to be made during the development of the simulation, generally going towards an increased level of accuracy and physical realism, at the expense of some more computer time, the build phase of the model is controlled by several "**logical flags**", that are simply string variables that serve as "**switches**" turning on or off an optional feature, or choosing between some mutually exclusive formulations: for example inclusion of buoyancy effects, choice of drag formulas to use, or variable length of cable, and so on.

Geometry

The cable connecting an immersed body (towed or self propelled) is usually spooled from the aft part of the ship: if we refer to the typical operating situation **(fig. 2)**, the cable will leave the surface of the sea in a direction opposite to the motion of the ship (approximately pointing in the direction opposite to the relative velocity ship/current), will form an ample curve toward the immersed body, eventually passing under the ship hull if the body is self-propelled ahead of it.

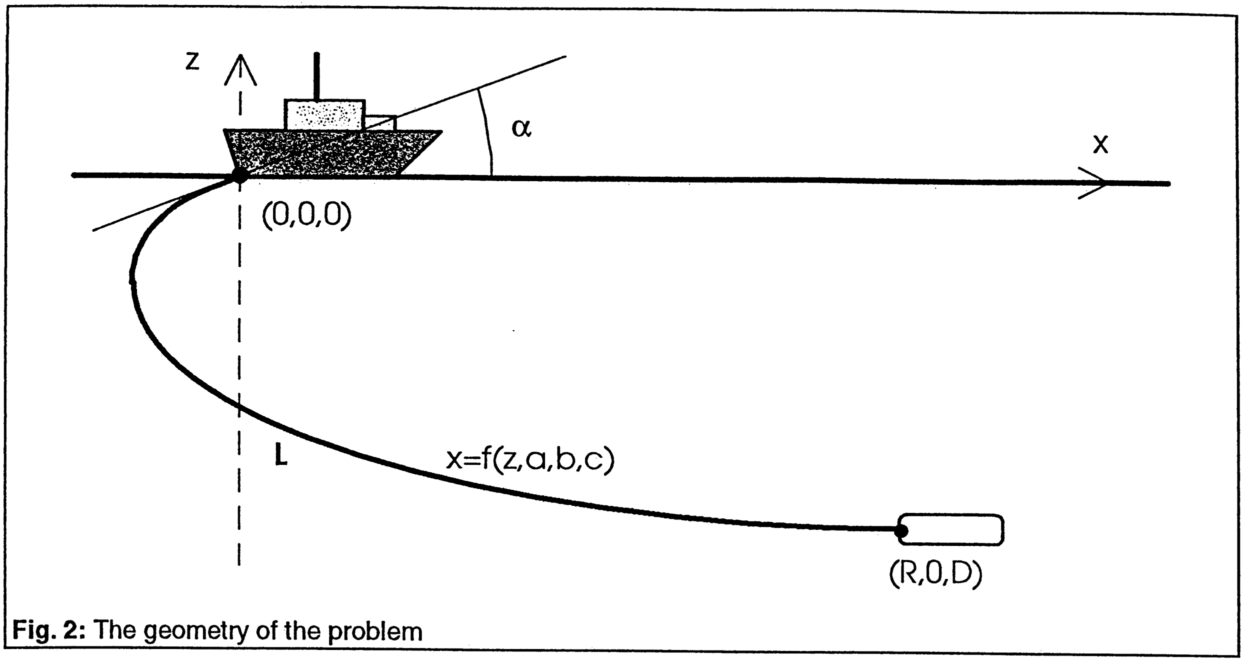

Fig. 2: The geometry of the problem

The exact shape of the cable is, of course, part of the problem: but to build a model we must supply some kind of initial position (and velocity) to all parts, preferably as near as possible to a stationary solution, in order to avoid long computational times only to reach a physically acceptable "initial condition".

The operator of the system (and / or the "intelligent winch" software) only knows, for a given instant in time, the position (reach, misalignment, depth) of the ROV with respect to the cable origin (i.e. the ship) and the **length** of the cable spooled until that time: it is from these data that the initial condition of our model should be built. This will only be possible if an analytic equation describing the cable shape is chosen, leaving then to the ADAMS solution routines the task of "shaping" the cable to the real final solution.

To impose the length of an appropriate parametric curve and two points of passage, asking to resolve analytically for the curve parameters, has proven to be a difficult task: for a plane problem, with the origin on the ship, one can write

$$x=f(z,a,b,c)$$

$$f(0,a,b,c)=0 \qquad f(D,a,b,c)=R$$

$$\int_0^D \sqrt{\left(\frac{d}{dz}f(z,a,b,c)\right)^2+1}\,dz=L$$

where L is the known length of cable, D is the signed depth, and R is the reach; this is a system of three simultaneous non linear equations in the three unknowns a, b, c, representing the function parameters.

Of the three suitable functions we thought of (sine/cosine, catenary, parabola) **none** could be resolved in closed form; so the choice we made, for the **parabola**, was driven only by relative simplicity **(fig. 3).** The formulas shown have been implemented in the .cmd language, but they do not permit to compute the parabola coefficients *a, b* when the length *L* is known.

$$x=a\cdot z^2+b\cdot z \qquad y=\frac{M}{D}\cdot z \qquad a=\frac{R-\dfrac{D}{\tan(\alpha)}}{D^2} \qquad b=\frac{1}{\tan(\alpha)}$$

$$a1=\sqrt{4\cdot D^3\cdot a\cdot b+D^2\cdot b^2+4\cdot D^4\cdot a^2+D^2+M^2}$$

$$b1=\sqrt{D^2+D^2\cdot b^2+M^2}$$

$$a2=\ln\left(a1+2\cdot D^2\cdot a+D\cdot b\right)$$

$$b2=\ln(b1+D\cdot b)$$

$$L1=\frac{1}{4}\cdot\frac{D^2\cdot(2\cdot a\cdot a1+a2-b2)+M^2\cdot(a2-b2)+b\cdot D\cdot(a1-b1)}{\left(D^2\cdot a\right)}$$

Fig. 3: The formulas for computing the arc of parabola length L in the general 3-D case (R= reach, M= misalignment, D= depth)

A partial solution to the requirement of using *L* as an input data, has been found by programming the well known **"secant" iterative non-linear equation solution method** in the .cmd language, using the conditional looping construct "do while": the problem is transformed in one equation in one unknown (namely the initial angle *alfa* between cable and the horizontal water surface, see fig. 2), iterated until the cable length has reached, with a certain precision, the input *L*. *Then* the normal creation of parts, markers, etc. can begin, using the final value of the parabola coefficients found before: it is a partial solution, because the model is not parametric on *L* (in the ADAMS/View sense of the word: if one *modifies* the real value of the variable *L*, the model geometry will not modify itself! ...but it will if *alfa* is modified, and will re-compute a new *L*).

Another little geometrical problem that had to be solved somehow, is that of generating constant length "elements": this is desirable for numerical stability, for optimal precision, for clear graphical post processing, and also for aesthetical reasons. Of course this cannot be obtained by equal spacing on the depth coordinate (z): in this case too, a closed analytic solution could not be found: an approximate algorithm, similar to a numerical integration, has been introduced, with satisfying results.

Model Architecture

Due to the above mentioned intrinsic "Finite Element" characteristic of the system, a very intensive use of the iterative, looping, and conditional constructs available within the .cmd language has been necessary: most parts, markers, joints, primitive joints , single forces, and other data base objects have both names and, when applicable, function expressions, dependent on an **integer counter** *"i"* that represents the "node" number. Ample use of the little known array capability offered by the *design variable index* option has been required, for example for storing and retrieving the depths of the *i*-th "node", or the length of the *i-(i-1)* "element".

Sometime cumbersome and syntactically obscure **(fig. 4)**, it must however be acknowledged that this kind of use of the .cmd Aview language has really interesting potentialities: a repetitive structure, similar to a chain, with complex relationships between each segment has been assembled with a compact and completely parametric source.

```
force create direct force_vector &
 force_vector_name = (EVAL("cable_drag_"//i)) &
 i_marker = (EVAL("cable_nod_"//i//".cm")) &
 j_part = .dol.sea &
 ref_marker = (EVAL("cable_nod_"//i//".cm")) &
 x_force_function = &
  "-DM(",(EVAL(".dol.cable_nod_"//i//".cm")),",",(EVAL(".dol.cable_nod_"//(i-1)//".cm")),")", & ! L
  "*0.5*Cd*ro*de", &                          ! Normal Drag:    1/2 * Cd * ro * de
  "*VARVAL(",(EVAL("u_norm_"//i)),")", &       ! |Un|*Unx
  "*VX(",(EVAL(".dol.cable_nod_"//i//".cm")),",.dol.current.cm,",(EVAL(".dol.cable_nod_"//i//".cm")),")" &
 y_force_function = &
  "-DM(",(EVAL(".dol.cable_nod_"//i//".cm")),",",(EVAL(".dol.cable_nod_"//(i-1)//".cm")),")", & ! L
  "*0.5*Cd*ro*de", &                          ! Normal Drag:    1/2 * Cd * ro * de
  "*VARVAL(",(EVAL("u_norm_"//i)),")", &       ! |Un|*Uny
  "*VY(",(EVAL(".dol.cable_nod_"//i//".cm")),",.dol.current.cm,",(EVAL(".dol.cable_nod_"//i//".cm")),")" &
 z_force_function =  &
  "-DM(",(EVAL(".dol.cable_nod_"//i//".cm")),",",(EVAL(".dol.cable_nod_"//(i-1)//".cm")),")", & ! L
  "*0.5*Cdt*ro*de", &                         ! Tangential Drag: 1/2 * Cdt * ro * de
  "*ABS(VZ(",(EVAL(".dol.cable_nod_"//i//".cm")),",.dol.current.cm,",(EVAL(".dol.cable_nod_"//i//".cm")),"))", &
  "*VZ(",(EVAL(".dol.cable_nod_"//i//".cm")),",.dol.current.cm,",(EVAL(".dol.cable_nod_"//i//".cm")),")"  ! |Ut|*Ut
```

Fig. 4: an excerpt of the "iterative" part of model generation ... not exactly crystal clear! This part of .cmd code is defining the Morison's equation for drag on a cable segment

The first version of the cable was built using the "point mass" element, because from the beginning we knew there was no need to include bending flexion effects, and consequently rotational Degrees Of Freedom (DoF). But in the course of development this was upgraded to normal "parts", whose rotational DoF were eliminated by primitive joints, because the presence of an associated moving reference system on each node simplified enormously the drag force formulation.

At present the cable model architecture **(fig. 5)** includes:

- a series of N+1 **parts** that one could call "nodes" (cable_node_*i*, for *i*=0, N) , equally spaced along the cable, with mass properties parametrically computed taking half the mass of each segment connected to it. Part 0 is the beginning of the cable, on the ship, and part N is the end, on the ROV. A typical "production" value of N is about 40.
- a series of primitive **inline joints** that remove two rotational DoF and force the Z axis of each *i*-th node to point toward the (*i-1*)-th node, so approximating, for each node, the "tangential" direction during the solution
- a series of primitive **perpendicular joints** that remove the last rotational DoF of each "segment" around its Z axis (think to a mechanical tachometer cable), connecting the *i*-th node rotation to the *(i-1)*-th, up to the ship.
- a series of N **spring-dampers** (or single forces, depending on input logical flags) that one could call "elements", acting between each node and the previous, simulating the cable stiffness and damping; for our application this is typically very high, but the model makes no assumption on this
- and finally a **stretchable outline** connecting each node to the previous one, for graphical representation purposes

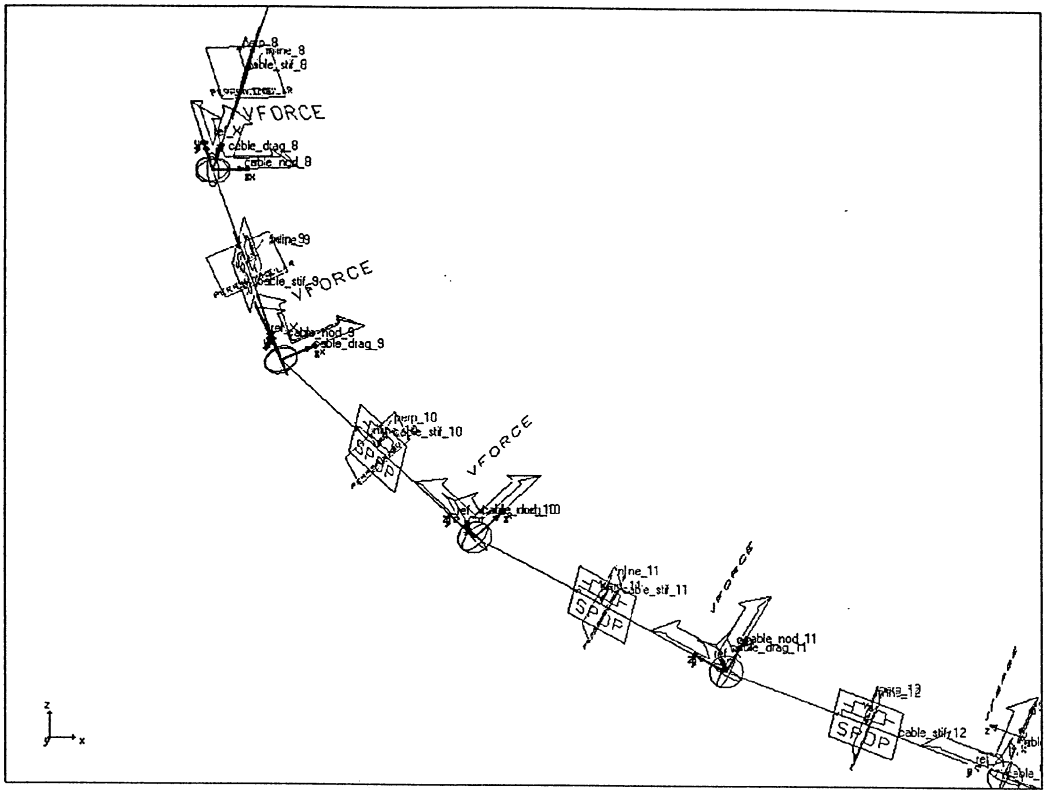

Fig. 5: an ADAMS/View's conception of the model architecture

the rest of the "iterated" part of the cable construction will be explained speaking of external forces. The ship and ROV are normal parts of no particular interest.

Current

Physically, the current can change dramatically with depth, both in magnitude and direction: but this is restricted to particular environmental conditions. At the moment our model can only simulate a current of constant speed and direction, and we have not still devised an ADAMS way of doing better; this remains one of the few areas where the FEM ANSYS model can produce exclusive results: but with a drag force formulation that has not proved to be the best one.

The current has been introduced in the model as a **"rigid body"** with zero DoF, translating at an uniform speed in a given horizontal direction (via a prametrically oriented prismatic joint and a parametric motion). A stretchable outline going from the origin to the actual point reached by the "current" provides visual feedback during animation.

This solution, that closely resembles the one already adopted for the wind in a previous work [10], is probably strange, but satisfying, because it permits, everywhere needed, to express the relative velocity of any point of the cable with respect to the current, simply using the standard *VX, VY, VZ, VM* ADAMS/Solver built-in functions.

Drag Forces

As it can be supposed, the "core" of the model is the representation of the drag forces on each segment of cable: this is by far the prevailing external force that drive the solution, and during the development phase we could observe that apparently small changes to the formulas could produce relevant disagreements in the results.

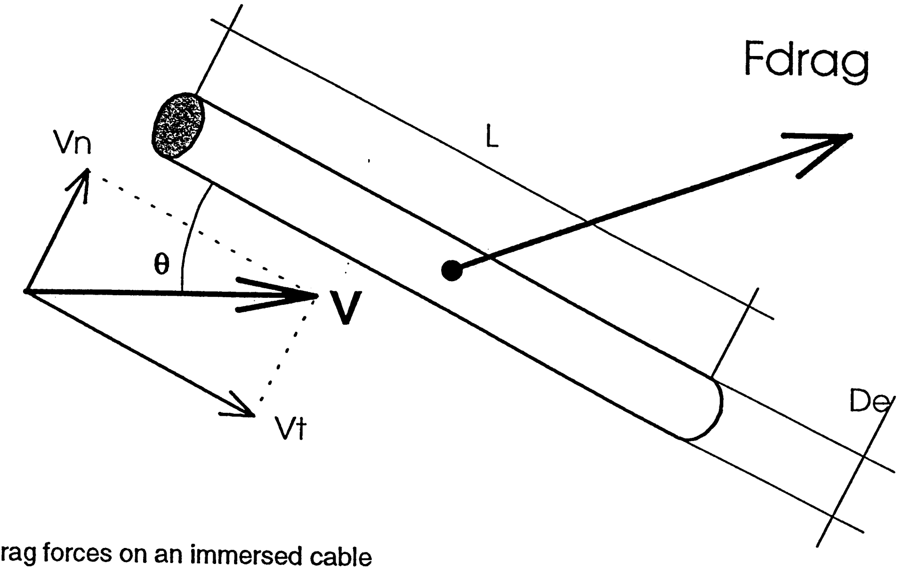

Fig. 6: The drag forces on an immersed cable

At the moment the model can be assembled, depending on a logical switch, using two different approaches to the evaluation of the stationary drag forces on an immersed slender cylinder **(fig. 6)** of length L, diameter D_e, surrounded by a moving fluid of density ρ having relative free-stream velocity V:

1. generalised Morison's equation [9]:

$$F_{drag}=L\cdot\left(\frac{1}{2}\cdot\rho\cdot C_t\cdot D_e\cdot\left|V_t\right|\cdot V_t+\frac{1}{2}\cdot\rho\cdot C_n\cdot D_e\cdot\left|V_n\right|\cdot V_n\right)$$

   this is the typical FEM programs formula, that implies the existence of two **independent** effects: the tangential drag, only function of the tangential relative velocity V_t, and of a constant adimensional coefficient C_t (for a given Reynolds number); and the normal drag, only function of the normal relative velocity V_n, and of a constant coefficient C_n. It is of course using this formula that the ADAMS model has been validated versus the FEM results.

2. Nordell-Meggit formulas [8], [11] that can be written, using the same symbols, as:

$$F_{drag}=L\cdot\left[\frac{1}{2}\cdot\rho\cdot\pi\cdot D_e\cdot\left(a\cdot|V|+b\cdot\left|V_t\right|\right)\cdot V_t+\frac{1}{2}\cdot\rho\cdot C_n\cdot D_e\cdot\left|V_n\right|\cdot V_n\right]$$

   this is a more complete approach, derived from experience (a, b are known numeric adimensional coefficients): as it can be seen, it differs form the previous formula only for the tangential drag component, that can be interpreted as a generalisation of the other; the physical meaning is that there is some **cross-coupling between the normal velocity drag and the tangential one**: infact, for a constant tangential velocity, the tangential drag component (i.e. C_t) depends *also* from the total free-stream relative velocity V. This is of course more plausible, and has been shown to produce considerably different results for our typical configurations.

The implementation of these formulas in the ADAMS model architecture has been attained by means of a series of N **vector forces** (one for each segment *i* of cable), and their expression has been greatly simplified by the introduction of parts reference systems that stay oriented along the cable during all the simulation: in this way, for example, V_t is just the VZ component, and V_n is simply the square root of the sum of the squares of the VX and VY components. With the same results, this simple "restructuring" of the model gave 15% CPU time gains.

Beside the drag due to a constant velocity of the fluid, the total force contains an **acceleration** dependent drag term, that we could define "inertial drag", physically representing the force required to "accelerate" the fluid mass displaced by the motion of the cable in the normal direction:

$$F_{inert\_drag}=L\cdot\left[C_m\cdot\rho\cdot\frac{\pi}{4}\cdot\left(D_e\right)^2\cdot A_n\right]$$

Where A_n is the acceleration in the normal direction, and C_m is the usual adimensional coefficient: due to the fact that at the moment no data is available on this coefficient, and that accelerations (and diameters) are comparatively small for our problem, we added this term under the conditional control of a logical switch.

Finally, it must be underlined that, differently from what happens using a standard FEM code, here the user has a complete control over the formulation for the drag load: we could add any kind of mathematical formulas we will find in the future, always keeping the same base model.

Other Forces

The other external forces that have been included, until now, in the model architecture, have a lesser importance, and no particular ADAMS implementation interest: they are briefly described for the sake of completeness.

The **buoyancy**, both of the cable and of the ROV, has been optionally taken into account: although in nominal operating condition the cable should be almost exactly neutral (this minimises the likelihood of cable fouling or of touching the sea bed), this will not always be possible, due to changes in salinity and temperature of water. So, under the control of the usual logical switch, gravity can be introduced in the simulation, and in this case a series of N+1 vertically oriented single forces (one for each part *i* of the cable discretization) are also created, based on the Archimedes law. Concerning the ROV, another vertical single force is applied in its center of buoyancy (while of course gravity will act on its center of mass).

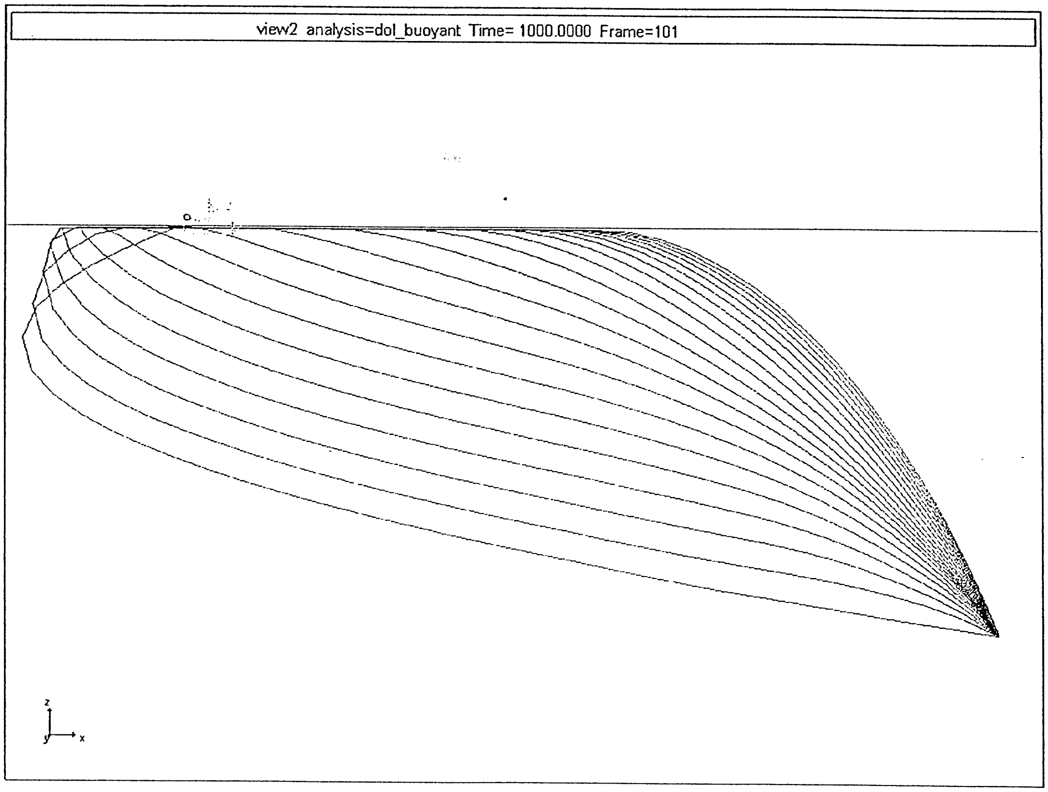

Fig. 7: A buoyant cable test case, with stationary ship and ROV

The **hydrodynamic drag** force **on the ROV** body, although small compared to those of the cable, has been expressed as a general force (3 forces + 3 torques) referred to ROV hydrodynamic center: their exact expressions will have to wait a better definition of the body itself; at the moment the translational components are computed using an estimated projected area for each Cartesian direction, while the torque components, normally zero, can be used to simulate the effect of directional fins, rudders, or other control surfaces, if the final ROV design will adopt this feature to help navigation control.

The same pre-design stage indetermination applies to the **propulsion** force: many options are still evaluated, that pose no problem of simulation; the simplest one has been implemented, namely an swivelling main propeller, represented by a vector force. Other solutions include two propellers for directional control, and/or various thrusters for better positioning.

Constrained or free motions

As already anticipated in the introduction, the model has to answer to at least two different kinds of question, possibly during different stages of project development:

1. find forces, given a trajectory
2. find trajectory given propulsion and control forces.

The first model implementation used two straight line motions for the ROV and for the ship, with simple uniformly accelerated motion laws for both of them (this covers a wide range of typical operations, from steady forward motion to deployment or recovery), leaving no DoF for the two vehicles: we have defined this mode of execution **"constrained motion"**, and its answers partially to question "1".

More general manoeuvre capability is required (turning, change of depth both for ship/submarine and for ROV, waves on the ship), so at present a **"3-D motion capability"** has been introduced for one or both the vehicles: a generic parametric equation as a function of time can be specified for position and orientation of the vehicle, i.e.

```
X= X(t)          AX= AX(t)
Y= Y(t)          AY= AY(t)
Z= Z(t)          AZ= AZ(t)
```

This is a very general formulation, and can describe almost any conceivable interesting motion, if one remembers the IF capability and the CUBSPL / AKISPL interpolation functions of the Solver; we considered it much more general than the "point-on-curve" method, to which anyway no motion could be imposed. Waiting for the announced 6 DoF motion available in ADAMS release 9, a macro command file had to be developed, and the model assembly procedure will call it one or two times, for ship wrt sea and/or for ROV wrt sea (that is logically equivalent to ground, in the model !).

The "**six_dof_motion**" macro has a general applicability every time 6 DoF must be removed between one part and another, at the meantime specifying analytic functions of time for the removed DoF in any direction (the same problem, for the wave induced motion of a ship, had been resolved as a particular case in a preceding work [10]): the macro takes as input parameters the names of the two parts to connect, and a marker on the first one to use as the "hinge" position; the result is the removal of exactly 6 DoF from the model, and the creation of 6 appropriately named motions (3 translational and 3 rotational) that will displace the second part with respect to the first. This is achieved by the creation of five dummy intermediate massless parts, connected by appropriate prismatic and revolute joints, on which are imposed the motions.

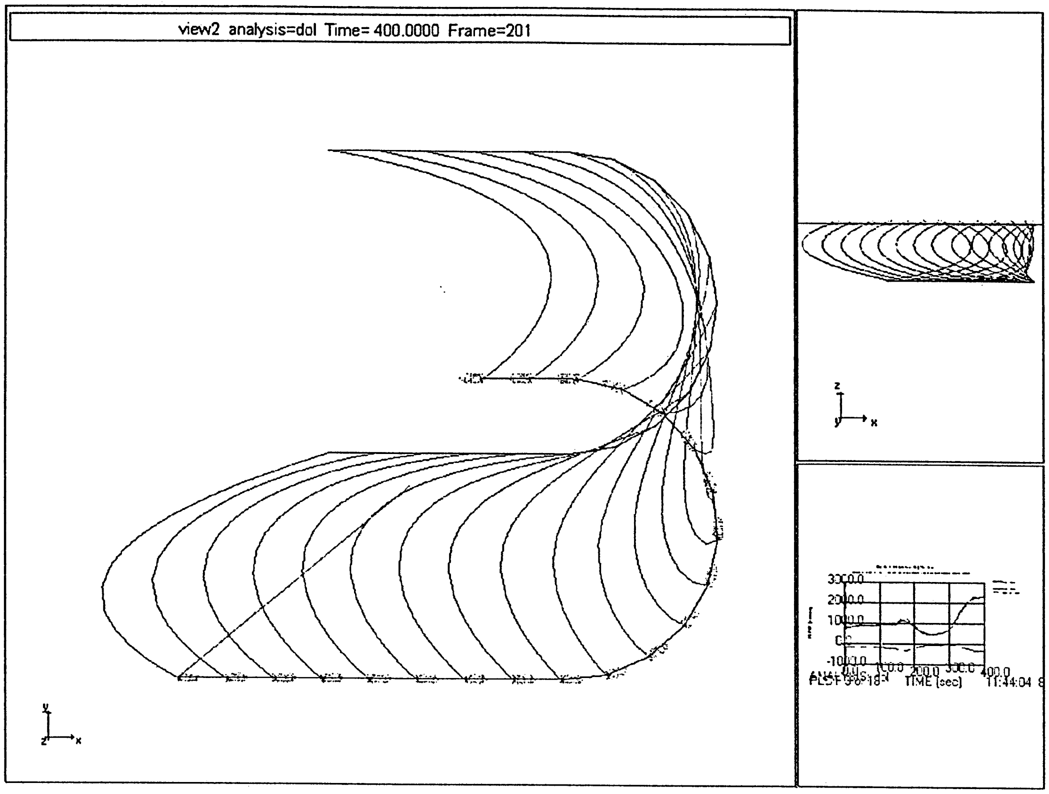

Fig. 8: An U-turn manoeuvre, with tail current, seen from above and from the side, with some cable intermediate positions. The small chart shows the evolution of forces during the turn (the final magnitude being higher than the initial due to the current).

One last interesting problem to be resolved for the "constrained motion" runs (question "1") was to avoid excessively long simulations, all the same being sure that a **stationary situation** of the cable (when this is possible) has been reached. For example, starting from the initial parabolic shape, in case of uniform equal velocity straight motion ahead of ship and ROV, a stable shape of the cable (and hence constant values of the reactions) can be reached in any time comprised between 30 sec and even 900 sec of simulation, depending on cable length and stiffness, current angle and speed, depth, slanting, reach, and many other factors: especially with a completely parametric model, one could not guess how long to run it in order to reach "equilibrium". The solution has been to (optionally) introduce a **special sensor** evaluating a combination of the relative velocity and the angular velocity of the middle point of the cable wrt the ship, *after* the initial transient: when this falls under a predetermined small fraction of the relative speed ship - current, the run will automatically be stopped.

The answer to question "2" is the **"unconstrained ROV"** mode of execution: no constraint is imposed on the ROV, but it is left free to evolve under the applied external forces (that in fact are activated *only* in these types of runs): cable drag and buoyancy, its own drag and buoyancy, and propulsion (as a function of time). It is very straightforward to realise that, left to itself, even with an appropriately oriented propulsion thrust, the underwater vehicle rapidly derives from the initial depth and position, under the effect of the cable "pulling its tail".

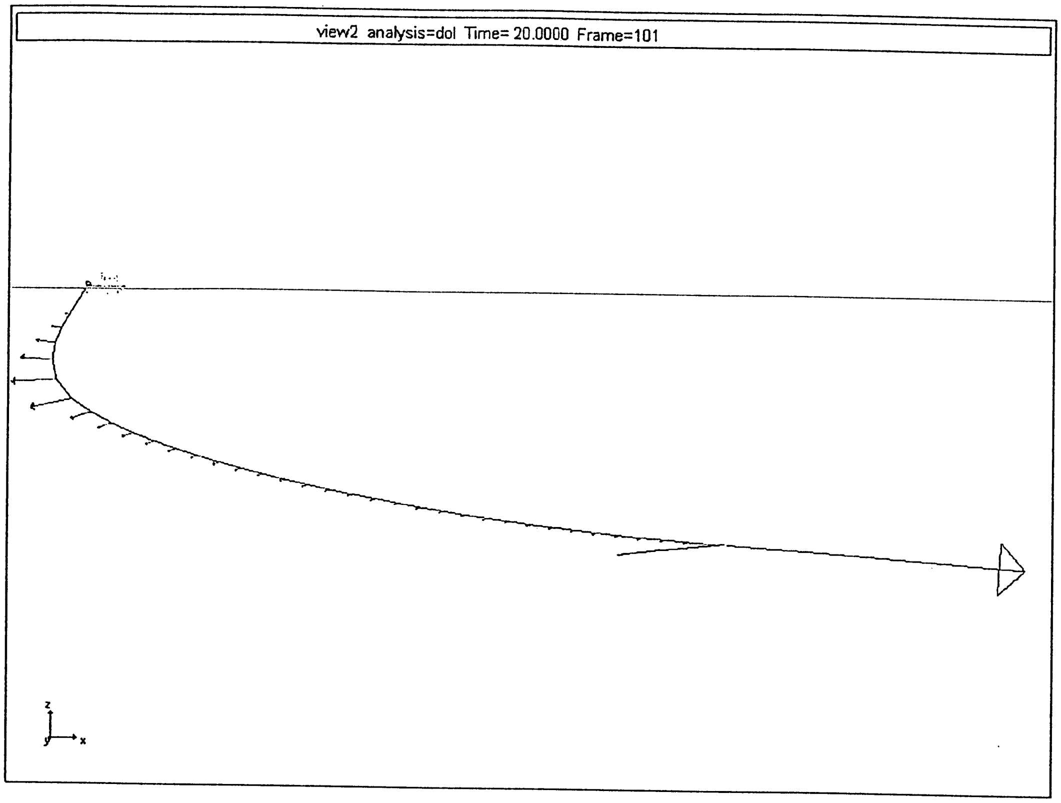

Fig. 9: A sample of free ROV motion, under the thrust of its propeller (in this case kept fixed in the axial direction)

This is more a **"feasibility" demonstration** at this stage, due to the indetermination about the final ROV shape and architecture: and it is in the same spirit that we even conducted some elementary **control system** development tests, with satisfying results: a simple PD control was applied simulating a diving rudder with the target of keeping a constant depth of the ROV. For a production development, use of the Linear module, and probably of the GSE instruction will be required: the literature on the subject even contains cases of real time simulations for control system development and testing [6].

Optimisation

From the beginning, it was required to build a completely parametric model, because of the anticipated interest of the design optimisation features built into ADAMS/View. For example, it is easy enough to understand that, for a given operational configuration (reach, depth, speed ...) and cable parameters (density, diameter,...), there should be and **optimal length of cable** connecting ship to ROV, in order to minimise the cable tension and/or the reaction on the ROV (and hence the power required from the motor): in fact **(fig. 10)** a short length of cable will increase the normal drag, while an excessively long cable will increase the tangential drag.

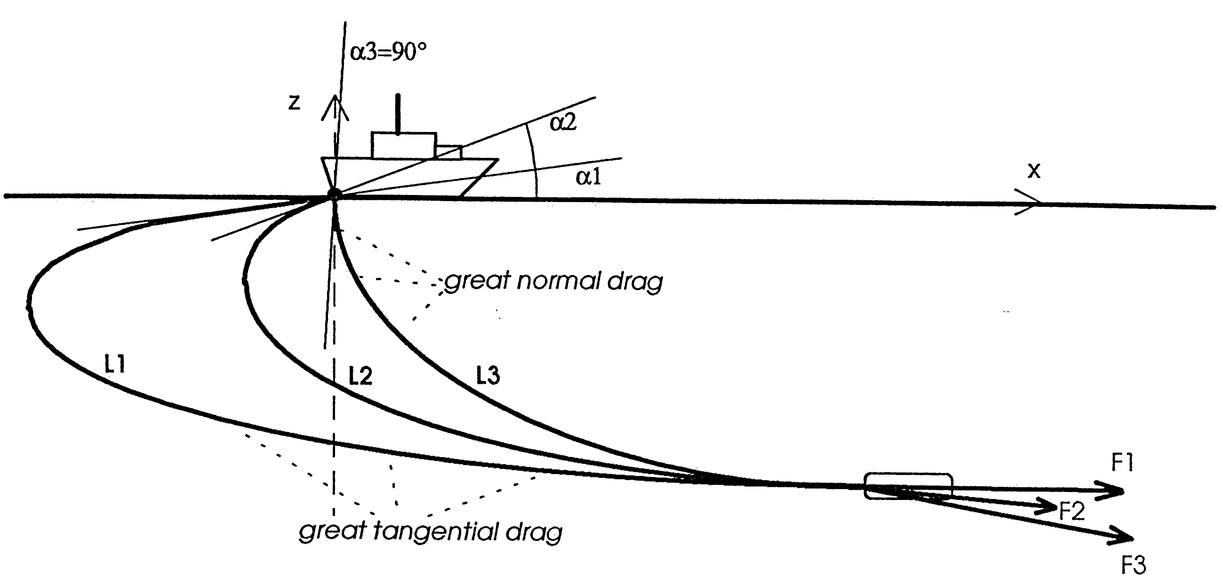

Fig. 10: There is an optimal length for any given configuration

All three levels of ADAMS/View optimisation tools (Design Study, Design Of Experiments, Simple Optimisation) can be used on the model, with the only limitations, described in the "Geometry" section, that there is no way to parametrize on *L*, the length of cable, due to the iterative nature of the solution: but the same result can be obtained modifying the angle "*alfa*", and then computing the derived Lengths (as a matter of fact, post-processing cmd files have been written in order to list and plot interesting object functions versus L).

Of particular interest for our problem is the **DOE**, with its **"fit response surface"** feature, that can produce a simple polynomial approximation to any desired result: this function (or simply the matrix of results) could be programmed into the software controller of the "intelligent winch", so giving it the ability, together with the measured data concerning the instantaneous position of the ROV, the current speed, etc, of evaluating the optimal length of spooled cable *in real time*, with no need of complex calculations.

Spooling

The "last frontier" of the actual model is the spooling simulation: of course it is interesting for operations like **deployment** and **recovery**, but also a simple change of depth or of reach would inevitably involve some spooling in or out of the cable from the mother ship. Being able to simulate these kind of operations would extend even more the capability of the model, for example for studying different strategies for such manoeuvres, in order to reach the best compromise between time and power required.

With a conventional FEM software, even using some kind of "element birth and death" features, we could not devise any way of doing this at a reasonable computational cost. On the contrary, after some experimentation, a good solution has been found in ADAMS, simply substituting the series of N spring-dampers with point-to-point **single forces**, reproducing the classical spring-damper equation, but where the "free length" L_0 variable is substituted by functions of time, each one of which is driven by a single "variable" L(t) to be defined, once, as a function of time in the model, and having the meaning of "spooled length" at time t:

$$F_{\mathrm{spring}(i)} = \frac{E \cdot A}{L_{0i} \cdot \frac{L(t)}{L_{0tot}}} \cdot \left(L_i - L_{0i} \cdot \frac{L(t)}{L_{0tot}} \right)$$

This means to introduce **"variable free length" springs**: if one remembers also to include the correction to the stiffness of the springs themselves, the formulation is sound and simple: in a validation study, a 66% increase of length of the cable produced only 1% maximum differences (in shape and reactions) with the same calculation made with normal "built-in" spring-dampers. Of course this feature, due to the greater computational requirements, is controlled by the already familiar logical switch.

Sample results

As an example of the results that can be obtained, we use a spooling - deployment test case. The current speed is 2 knots at 60 degrees, while the ship travels forward at an uniform speed of 3 knots, half of that of the ROV, travelling at 6 knots in the same direction at the constant depth of -150 m, slanted of 200 m; initially the ROV is exactly under the mother ship, and the cable length is 600 m.

After ten seconds from the start of the simulation, the cable is spooled out at exactly (6-3) knots, in order to maintain an approximately constant shape, while the ROV is constantly increasing its reach (a very simple and unrealistic deployment strategy, used only as an example of the model capabilities).

The cable is a 12 mm very stiff cable, with no buoyancy effect; the drag on it is computed using the Nordell-Meggit formula, and without the inertial drag contribution.

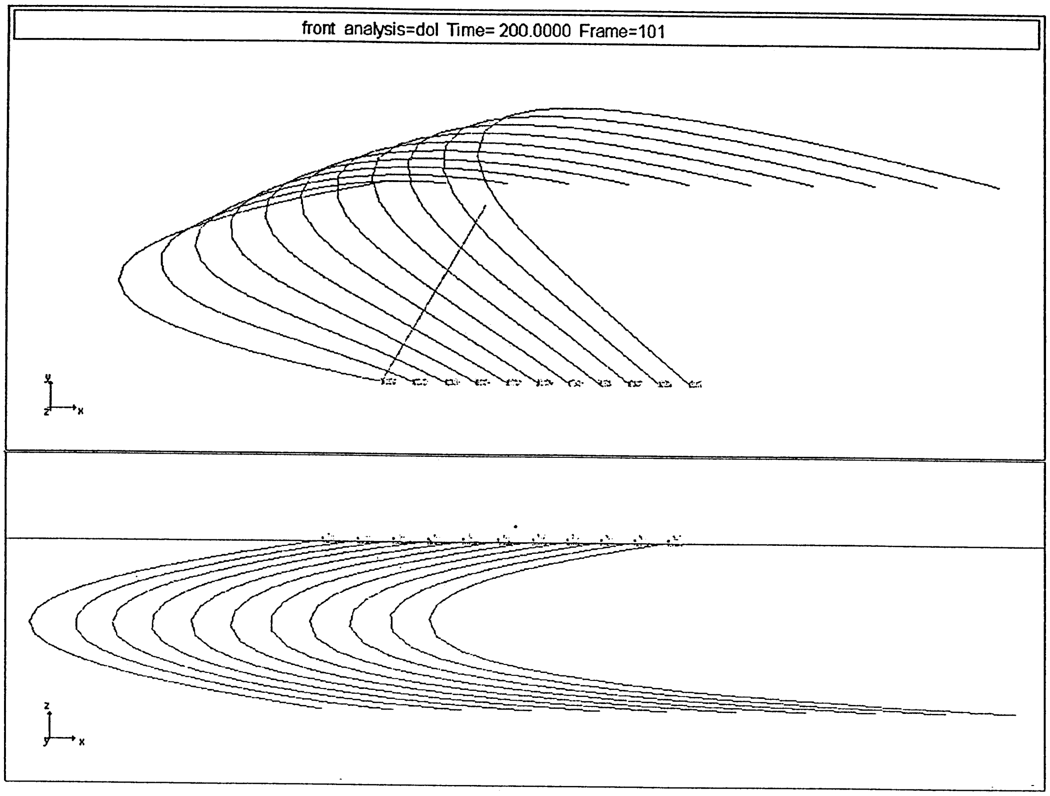

Fig. 11: Evolution in time of the shape of the spooled cable during the sample deployment test case: the "backward" curve of the parabola remains approximately undeformed...

The complete printout of the little report automatically produced at the end of a SIREN run, summarising input and output data, is contained in the **Appendix**. Here we supply two images showing some aspects of the simulation: **fig. 11** illustrates the shape of the cable, while **fig 12** concerns reaction forces.

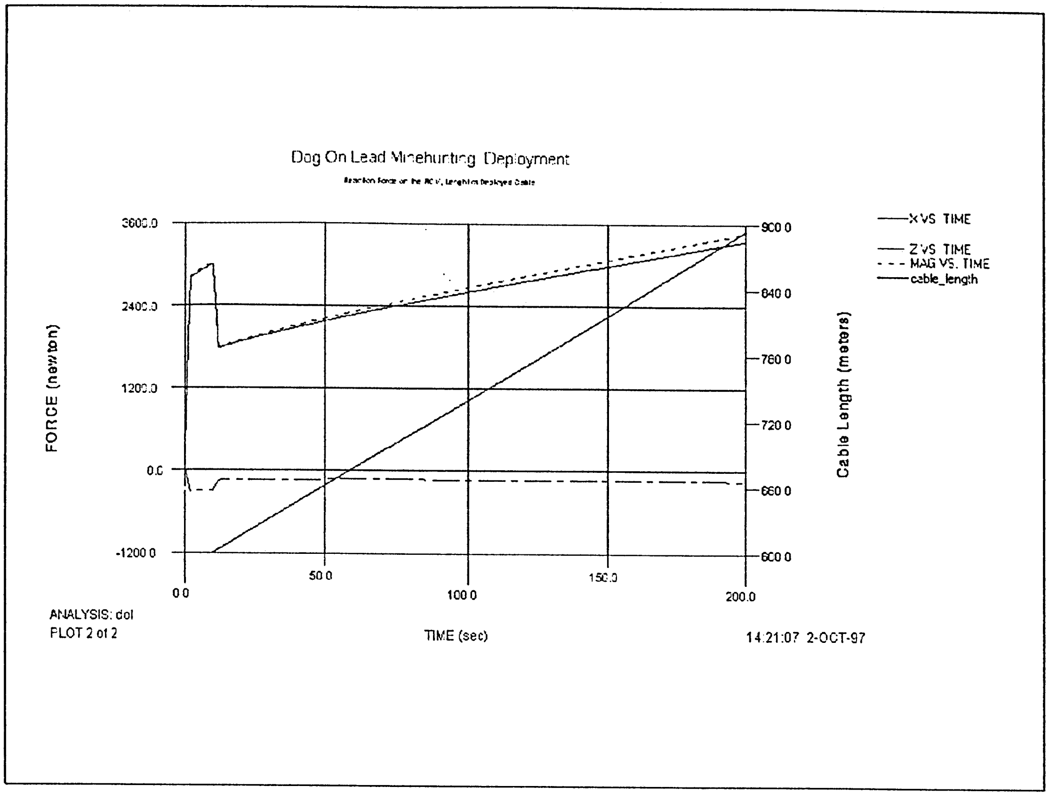

Fig. 12: Evolution in time of the reaction force on the ROV (i.e. of the power required to drive it at constant speed), superimposed to the spooled length of cable: it can be remarked the sudden decrease in the force when the cable begin to "loosen", followed by a steady linear increase due to the tangential drag of the horizontal part of the cable, becoming longer and longer.

Conclusions

The ADAMS based part of the project, provisionally named SIREN (System for Immersed ROV Evaluation in Navigation) was started only in August '97, and it seems obvious that it will undergo changes and evolution, being just in the first stage of development. We believe that it can be an invaluable tool of "virtual prototyping", enabling the company to explore many different design solutions and operating strategies during a very early stage of the product conception.

From the ADAMS user point of view, we would like to see an even better implementation of the .cmd language, especially concerning array and variable name composition syntax (e.g. some kind of feature to generate "series" of similar entities, like single forces or parts...), and also a better integration of the ADAMS/View language (.cmd) with the ADAMS/Solver language (.adm) (e.g. the accessibility of View array variables from the Solver language, the same built-in functions available at both levels...).

Future evolution of SIREN is difficult to predict, but will involve a steady growth in complexity, approximating better and better the real physical system that will emerge, with time, from the designers tables: regions of interest will be

- the propulsion scheme chosen for the ROV and, in close connection with this,
- the development of an automatic navigation control system (there should be no need to point out the extreme importance for the "fish" of following a predetermined path)
- the introduction of wave motions effect
- some form of real time driving simulator.

Acknowledgements

The author would like to thank all the colleagues and other persons that, in a way on another, supported, inspired, advised and endured him during the development of this work: they include Ing. Marco Baraldini, Ing. Carlo Crescentini, Ing. Sonia Benetti, Ing. Stefano Ghignone, Mr. Carlo Magagnini, Mr. Marco Tomada. To them go my thanks and my gratitude. Last but not least, thanks to Ing. Diego Minen and Ing. Daniele Catelani (Mechanical Dynamics Italy) for their precious and continuous assistance in the use of the ADAMS software.

References

[1] Lacey - Computer simulation aids cable handling - Undersea Defence Technology (UDT) Report, issue 1.

[2] Perles, Ranz-Guerra - Calculation of the shape and position of a towed variable depth fish

[3] Barrie, Watters, Stanton - A systematic approach to the evaluation of the dog on lead minehunting configuration

[4] Klevebrant, Akesson - A precision manoeuvrable MCM ROV system to encounter tomorrow's mine threat - UDT conference, London 1996

[5] Hornfeld, Kermorgant, Leicht - Unmanned underwater vehicles for mine detection, classification and identification - UDT conference, London 1996

[6] Kalwa, Hoffmann, Kermorgant - Application of virtual prototyping in the design of a SVDS minehunting system - UDT conference, London 1996

[7] Kalwa, Hoffmann - Relationship of underwater vehicle parameters to SVDS performance - UDT conference, Hamburg 1997

[8] Blevins - Applied Fluid Dynamics Handbook - Van Nostrand Reinhold 1984

[9] SAS IP, Inc - ANSYS Elements Reference / ANSYS Theory Reference

[10] Pedrazzi, Barbieri, Spanghero - ARCHIMEDES: The Riva Calzoni Helicopter Ship Dynamic Interface modelling software - WARSHIP '97 Royal Institution of Naval Architects (RINA) conference, London 1997

[11] Nordell, Meggit - Undersea Suspended Cable Structures - American Society of Civil Engineers, Journal Struc. Div. n. 107, 1981

Appendix: report file listing

```
SIREN - System for Immersed ROV Evaluation in Navigation
Riva Calzoni - Scientific Computing Department - Ing. C.Pedrazzi


Title: Dog On Lead Minehunting: Deployment

Model: dol   Analysis: dol   Units: MKS    Angles: degrees
Time & date: 14:21:07  2-OCT-97    Result steps: 101    Request steps: 101
Reported  Step Number:  101   Time: 200

Discretization  n: 30    nc: 30

=== I N P U T ===

Logical Flags
=============
Gravity & Buoyancy   : OFF
Drag Forces Model    : Blevins
Inertial Drag (Cm)   : OFF
Variable Cable Length:  ON
Constrained R.O.V.   :  ON
R.O.V. motion 3-D    : OFF
Ship motion 3-D      : OFF

Operational Parameters
======================
Ship Vel. & Accel. X: Vs=     +1.5433   As=          +0
ROV Vel. & Accel.  X: Vf=     +3.0866   Af=          +0
ROV Initial Position: Rx=          +0   My=        +200   Dz=       -150
Cable Angle & Length: al=     +8.1892   Lc=      599.22   L =        600
Parabola Coeff.& Dat: a =   +0.046325   b =     +6.9488   Lt=     599.95

Cable Length    L(t): STEP(time,10,.dol.Length,10.1,.dol.Length+(.dol.Vf-.dol.Vs)*(time-10))

Cable Design Parameters
======================
Diameters           : Dc=       0.012   De=       0.012
Drag Coefficients   : Cd=         1.2  Cta=       0.083  Ctb=     -0.035  Cm= 1
Material properties : ro=        1110   E =     2.1e+11   tc=        0.1

Environmental Parameters
========================
Current Speed & Dir.: Vc=     +1.0289   Hc=          60
Water properties    : ro=        1025   nu=    1.05e-06


=== O U T P U T ===

Reaction Forces (in Ship & ROV ref. sys.)
===============
Ship                : FX=      +672.7   FY=      -573.1   FZ=     +106.4
ROV                 : FX=     +3352.4   FY=     -844.52   FZ=     -144.4

Applied Forces (in Sea ref. sys.)
=============
Velocity Drag       : FX=       -4025   FY=     +1417.6   FZ=    +38.018

Geometrical Data
================
Total Force Depth   :                                     Zf=    -107.44

Cable Nodes Coordin.:    === X(i) ===      === Y(i) ===      === Z(i) ===
                 0 :          +308.66                +0               +0
                 1 :          +286.56           +18.828          -3.4956
....
                28 :           +554.7           +215.28          -147.08
                29 :          +583.59           +208.32          -148.46
                30 :          +617.33              +200             -150
```