Rigid Tether Modeling for Underwater Environment Simulation

Quentin Brateau, ENSTA Bretagne, Brest, France, quentin.brateau@ensta-bretagne.org

Abstract—Simulation in the field of robotics is a powerful tool. Indeed, it allows to easily and quickly test the robot in different conditions and to have a reproducibility of the results. Then it let us be able to create situations that would be difficult to find in reality, in order to make sure of the robot's behavior. It should be noticed that the simulation of robots does not replace tests in real conditions, but it remains practical during the development phase. However, the simulation of robots in the maritime environment is a field that still has shortcomings, especially when we want to simulate the tethers of submarine robots. Indeed, it is not easy to find an analytical way to simulate a tether that does not require large resources, especially when simulation environments become complex. This is why we will try to suggest a finite difference simulation of the tether by proposing a behavioral model of force between each tether element. The results seem satisfactory and with a correct initialization of the position of the tether elements, the behavior of the tether seems quite right. However, the tether stiffness phenomena are not taken into account in this modeling, the tether is of fixed length, and both ends have been fixed so the system has not been tested dynamically.

Index Terms—Modeling, Tether, Remotely Operated Underwater Vehicle, Simulation.

I. Introduction

In the world of marine and underwater robotics we can indentify two categories of elements: mobile marine objects (MMO) and flexible tethers (FT) [?]. Mobile marine objects include surface vessels, submarines and remotely operated vehicles [?]. Flexible tethers can represent umbilical cables, traction cables and anchor chains in the marine environment. They constitute all the necessary elements to achieve a mission in this environment which is sometimes quite difficult to explore.

The simulation of moving marine objects is something well known [?]. We are now able to know from state equations the behavior of robots in their environment, and these equations are known for ships, submarines, sailboats, etc... [?]

On the other hand, determining the behavior of a flexible tether becomes more complicated. Indeed, the equation of motion of these objects involves non-linear partial differential equations and the motion between the different objects in the environment are dynamically dependent [?]. It is well illustrated that if the boat moves, it will induce a motion in the flexible tether that will modify the trajectory of the remotely operated vehicle. This is why it is difficult to find a way to model flexible tethers in underwater environments.

The idea proposed in the scientific paper is to solve this problem using finite element simulation. This implies that we need to discretize the tether in order to simulate its global behavior.

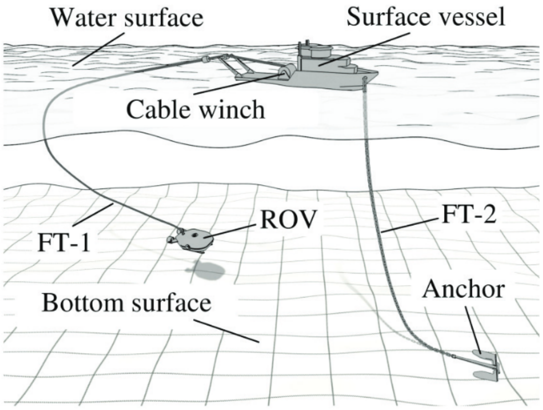

Fig. 1. Example of a complex underwater environment

II. Formalism

Suppose we want to simulate a tether of length L. We will then divide it into a finite number n of nodes connected by links. These links should be of length $l = \frac{L}{n-1}$ as the two nodes at the ends of the tether will not be connected to any other links.

We will focus here on the case where the first node and the last node are immobile, because otherwise we would have to simulate a mobile marine object that would be attached at the end, which is not the goal of our study.

Next, it is necessary to make a balance of the forces that apply to each tether element. For this simulation, we will take into account the weight, noted $\overrightarrow{F_g}$, the buoyancy, noted $\overrightarrow{F_b}$, and the force exerted by the previous element on the considered element, noted $\overrightarrow{F_{t,p}}$, as well as that of the next element, noted $\overrightarrow{F_{t,n}}$. The last force added to this model is the drag force noted $\overrightarrow{F_f}$

- Weight $\overrightarrow{F_g}$: Considering that each element has a mass m, we have :

$$\overrightarrow{F_g} = \begin{bmatrix} 0 \\ 0 \\ -m.g \end{bmatrix}$$

- Buoyancy $\overrightarrow{F_b}$: If we note the volume of each element V and ρ the density of the fluid in which the tether is immersed, we have :

$$\overrightarrow{F_b} = \begin{bmatrix} 0 \\ 0 \\ \rho.V.g \end{bmatrix}$$

- Tether force $F_{t,p}$ and $F_{t,n}$: It is difficult to find an analytical form to describe these two forces. Therefore, we must find a way to describe them. This is why we will use a behavioral model here. We know that each node will have to be at a distance l from each of its neighbors. We can assume that the system behaves here as a three-dimensional damped mass-spring system and we will then consider that these forces are like elastic spring forces and viscous frictionnal forces.

  By noting then p_p the position of the previous node and p_c the position of the current node, by introducing three coefficients K_p, K_d and K_i allowing to express the stiffness with which a node will correct its position with respect to its neighbors, we are able to express the behavioral model of these two forces:

  $$\overrightarrow{F_t} = -\left(K_p \cdot e + K_d \cdot \dot{e} + K_i \cdot \int_0^t e(\tau) \cdot d\tau\right) \cdot \overrightarrow{u}$$

  In these expressions, it is assumed that $\overrightarrow{u}$ is the unitary vector oriented from the current node to the neighboring node, $e(t)$ is the error of position between two nodes, $\dot{e}$ is the derivative of this error and $\int_0^t e(\tau) \cdot d\tau$ is the integral of this error. Both derivative and integral part will be estimated numerically respectively using the Euler's method and the rectangles method. We have therefore :

  $$\overrightarrow{u} = \frac{\overrightarrow{p_c - p_p}}{\|p_c - p_p\|} \qquad e(t) = \frac{\|p_c - p_p\| - l}{\|p_c - p_p\|}$$

  These two forces $F_{t,p}$ and $F_{t,n}$ can therefore be expressed using the expression of F_t, taking care to take the correct current and previous element for each case.
- Drag $\overrightarrow{F_f}$: By noting f the coefficient of viscous friction, $\overrightarrow{v}$ the velocity of each node and considering that the frictions are quadratic, we have :

  $$\overrightarrow{F_f} = -f \cdot \|\overrightarrow{v}\| \cdot \overrightarrow{v}$$

Thus we have expressed the forces necessary for the simulation of the tether.

III. Implementation

The Python 3 implementation of this simulation is available in a GitHub repository [?]. It was developped using Numpy [?], Matplotlib [?] and Scipy [?]. The goal of this simulator is to study the viability of such a system, in particular to validate the performance of the tether with this behavioral model.

So we will create a class TetherElement which will represent a node. It will have to contain its mass, volume and distance information from its neighbors, but also its position, velocity and acceleration, as well as a pointer to each of its two neighbors. Finally it is necessary to have each coefficient K_p, K_d and K_i to compute $F_{t,p}$ and $F_{t,n}$.

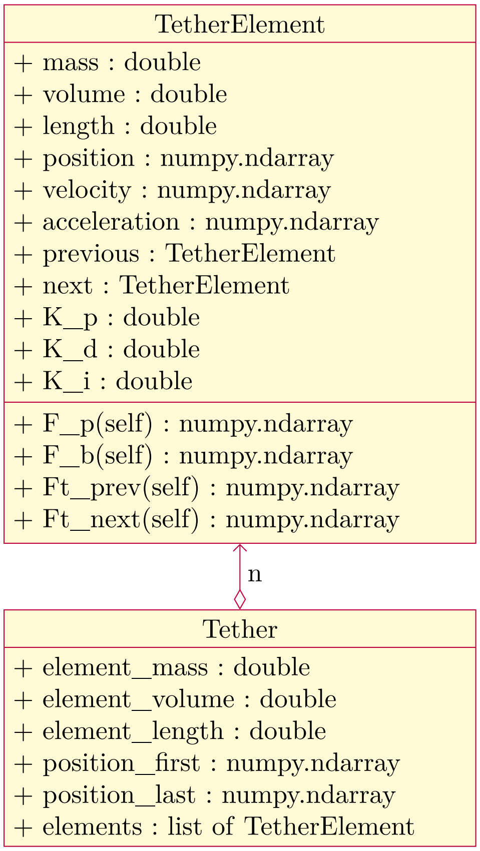
TetherElement
+ mass : double
+ volume : double
+ length : double
+ position : numpy.ndarray
+ velocity : numpy.ndarray
+ acceleration : numpy.ndarray
+ previous : TetherElement
+ next : TetherElement
+ K_p : double
+ K_d : double
+ K_i : double
+ F_p(self) : numpy.ndarray
+ F_b(self) : numpy.ndarray
+ Ft_prev(self) : numpy.ndarray
+ Ft_next(self) : numpy.ndarray
n
Tether
+ element_mass : double
+ element_volume : double
+ element_length : double
+ position_first : numpy.ndarray
+ position_last : numpy.ndarray
+ elements : list of TetherElement

Fig. 2. UML diagram of the implementation

This will allow us later on to implement a Tether class to be able to simulate a tether. This object must have a length, a number of elements and a list containing the different TetherElement that compose it. It must also know the mass, the volume of each node in order to correctly instantiate the TetherElement, but also the length of each link between nodes.

A diagram of these two classes is visible on the Figure ??. It respects the UML format and allows to see the different class variables and methods associated to each class.

IV. Initialization

Initialization is an important step because if the initial position of each TetherElement is random, the Tether will take a long time to converge and the system will be inconsistent. This is mainly due to the fact that the coefficients of the behavioral model are set to keep the nodes at a good distance from each other when a small perturbation is brought to the system.

To initialize the different nodes, we use the catenary equation. The idea is to use the shape taken by a rope attached at the ends to two fixed coordinate points. This rope will want to minimize its energy and so it takes this shape. This chain should check the following second order differential equation.

$$\ddot{z} = \frac{1}{k} \cdot \sqrt{1 + \dot{z}}$$

The solutions are known and of the form :

$$z(x) = k \cdot cosh\left(\frac{x}{k}\right)$$

However, this solution shows a rope centered around the ordinate axis. This is not necessarily a situation that we will find in our simulation. Here we would like to set the two fixed extremities, noted (x_1, y_1, z_1) and (x_n, y_n, z_n). By introducing c_1, c_2 and c_3 three coefficients allowing to correctly place the rope, we will then want to find here an equation of the form :

$$z(x) = c_1 \cdot cosh\left(\frac{x + c_2}{c_1}\right) + c_3$$

To find these coefficients, we have at our disposal three conditions: the two conditions related to the end points and the length of the string which must be equal to L. These conditions are expressed by the following equations:

$$\begin{aligned} L =& c_1 \cdot sinh\left(\frac{x_n + c_2}{c_1}\right) - c_1 \cdot sinh\left(\frac{x_1 + c_2}{c_1}\right) \\ z_1 =& c_1 \cdot cosh\left(\frac{x_1 + c_2}{c_1}\right) + c_3 \\ z_n =& c_1 \cdot cosh\left(\frac{x_n + c_2}{c_1}\right) + c_3 \end{aligned}$$

It is possible to solve the system of equations numerically using the function fsolve of the package scipy.optimize. Finally, from the calculated coefficients, and by knowing the length between the first node and the i^{th} node, it is possible to initialize the nodes by reusing the previous constraints but the unknowns become the position x_i and z_i. Note that the constraint on the position of the first node brings nothing to the system of equations, which leads to a system of two equations with two unknowns. The y_i of each TetherElement are linearly spaced between the two extremities.

V. Results

This section will present the results of the simulation. To set the different coefficients of the simulation, we will need to build some tools to evaluate the performance of such a model.

A. Length of different links

It is useful to trace the length of the links over time in order to verify that the Tether's behavior is correct.

The Figure ?? shows in gray the plot of the length of each link for a simulation containing 15 nodes. In crimson is plotted the average of these lengths and in yellow the target length. Finally, the blue area shows the 95 % confidence interval.

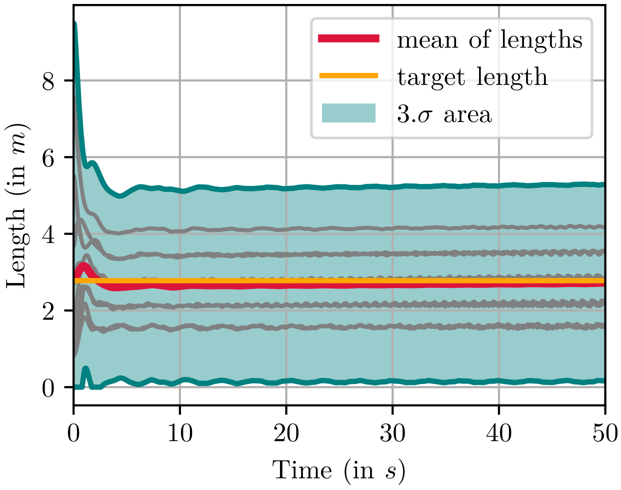

Fig. 3. Length of the links, average length and confidence interval

As we can see, after setting the coefficients we are able to have a coherent behavior a few seconds after initialization. On average the link lengths converge towards the set length and the standard deviation is not too high. However, there are still some oscillations and not all links are exactly at the target length.

B. Relative error

The relative error is also very interesting because it makes it possible to be truly aware of the error but also to see more clearly when setting the coefficients of the behavioral model.

The Figure ?? shows us the average relative length error of the different links compared to the target length.

The adjustment of the coefficients of the behavioral model is done as follows. First the K_p is set to obtain an oscillating system with a bounded average link length. Then we add a derivative effect to remove oscillations on the system by increasing the coefficient K_d. Finally we add an integrator effect to remove a static error in order to make the length of the links reach the target length despite the presence of disturbances such as weight and buoyancy. This coefficient must be as large as possible in order to quickly converge but not to the point of destabilizing the system.

C. Energetical approach

Finally, the energy approach is the most important to validate the simulation, because it carries the physical meaning of the system. The system must not have a divergent energy, which would be a physical counter-sense, but here, as there is no energy source, the overall energy must decrease over time, as soon as there is some fluid friction due to the drag force.

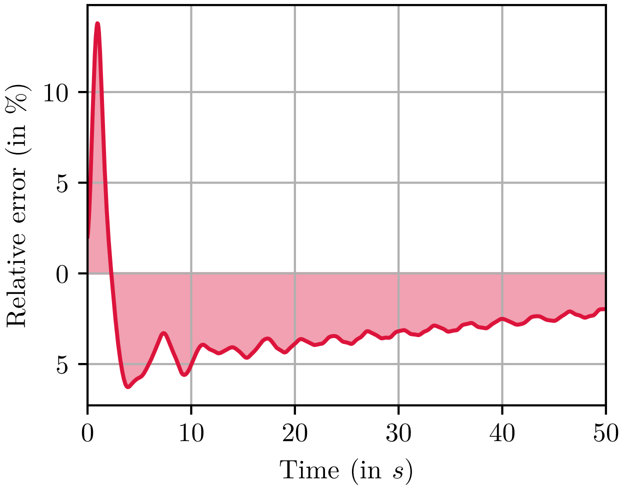

Fig. 4. Relative error between the target length and the mean length of links

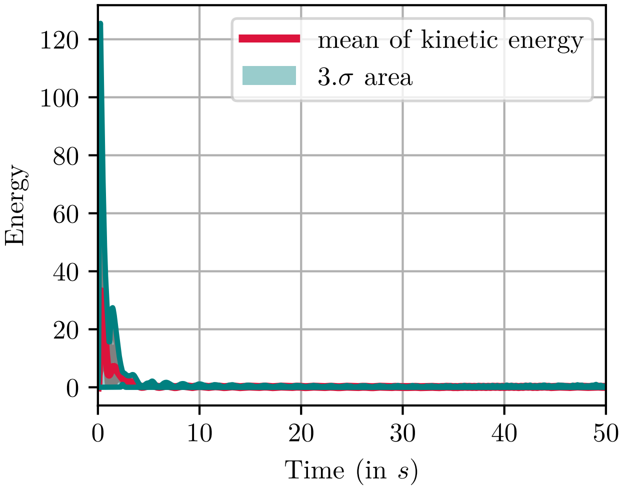

Fig. 5. Kinetic energy of the system

1) Kinetic Energy: The global kinetic energy of the Tether is calculated simply by summing the kinetic energies of the different TetherElements.

$$E_k = \sum_{i=0}^{N} \left(\frac{1}{2} \cdot m \cdot v_i^2 \right)$$

Where v_i is the velocity of the i^{th} element.

2) Potential Energy: To calculate the potential energy related to the application of a force on a solid, it is necessary to return to the definition of potential energy. Indeed we know that by noting dt the time step of the simulation :

$$\delta W(\overrightarrow{F}) = \overrightarrow{F} \cdot \overrightarrow{dOM} = \overrightarrow{F} \cdot \overrightarrow{v} \cdot dt$$

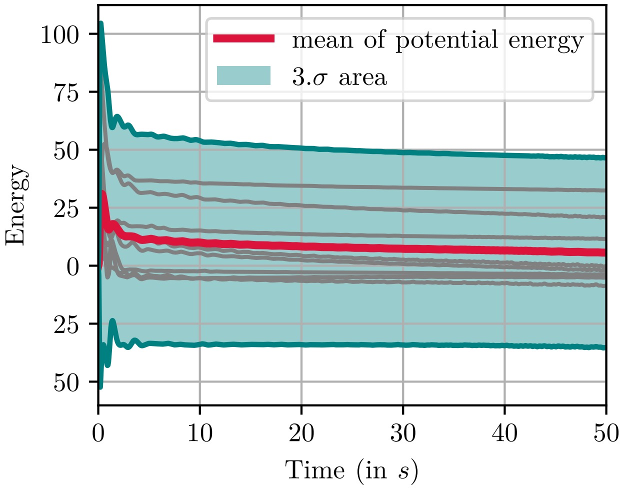

Fig. 6. Potential energy of the system over the time

$$E_p = \int_0^t \sum_{F_{ext}} \delta W(\overrightarrow{F_{ext}}) + cste$$

Thus we are able to calculate the potential energy by calculating the sum of the elementary work of each force on the system, i.e. the forces that apply to each node, and then by integrating this quantity over time. We then have an expression of the potential energy of the system over time for the Tether which is defined to within a constant. This constant is initially set to 0 in our case.

The Figure ?? shows us the evolution of potential energy over time. In gray is plotted the potential energy of each node, in crimson is the average potential energy and the blue area represents the 95 % confidence interval. We see that the potential energy of the system does not diverge, and that the system tends to position itself in a minimum of potential energy. This is something that has a strong physical meaning because all real systems tend to put themselves in positions that minimize their potential energy. This seems to validate the behavior of our modeling on this point.

3) Mechanical Energy: Finally, by summing the kinetic and potential energies previously calculated, we are able to define the mechanical energy of the system. This will give us information about the non-conservative forces that are included in this system. Indeed, the drag force will cause the system to lose energy since this energy will not be transformed into another form that can be used by the system, but will be dissipated as heat.

The Figure ?? shows us the evolution of the mechanical energy over time. In gray is plotted the mechanical energy of each node, in crimson is the average mechanical energy and the blue area represents the 95 % confidence interval.

We can see that the energy of the system does not diverge, which seems to support the idea that the modeling is correct. Then, we notice that the system tends to minimize

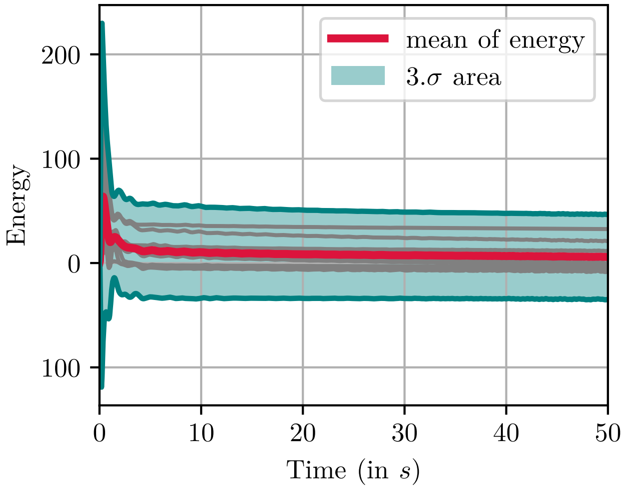

Fig. 7. Energy of the system

its energy, which is exactly the behavior expected for any system.

VI. Conclusion

In conclusion, the tether modeling as presented in this scientific paper seems to give correct results. The idea of reasoning by finite differences allows us to simulate the tether as a succession of tether elements linked by unstretchable links. The problem induced by this method is to provide a model to describe the force of a node on its neighbors. The proposed behavioral model provides good results here, and which leads to a simulator with a physical meaning.

In addition, there are still some issues that have not been addressed. First of all the Tether was not tested in an environment where the extremities were subjected to movement. Second, the Tether is simulated with a fixed length, which is not necessarily the case during a submarine mission. Indeed it is common to have to unroll and rewind the Tether during the mission to prevent it from becoming tangled. Finally, there is no force that simulates the stiffness of the Tether. It is unthinkable to be able to bend a real Tether, it is not infinitely flexible.