# Flexible 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, 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.

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



Fig. 1: Example of a complex underwater environment, *O. Blintsov, "Development of the mathematical modeling method for dynamics of the flexible tether as an element of the underwater complex", 2017.* [1]

## I. INTRODUCTION

In the world of marine and underwater robotics, we can identify two categories of elements: mobile marine objects (MMO) and flexible tethers (FT) [1]. Mobile marine objects include surface vessels, submarines, and remotely operated vehicles [2]. Flexible tethers can represent umbilical cables, traction cables, and anchor chains in the marine environment [2]. They constitute all the necessary elements to achieve a mission in this environment. FIGURE 1 shows an example of a complex underwater environment as we could find in real conditions.

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... [2], [3]

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 [1]. 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.

Most of the existing solutions in terms of physical simulation are position-based simulations [4]. This is a field that is well found in the various references that propose flexible tether simulation methods. The idea is to make a simulation based on a discreti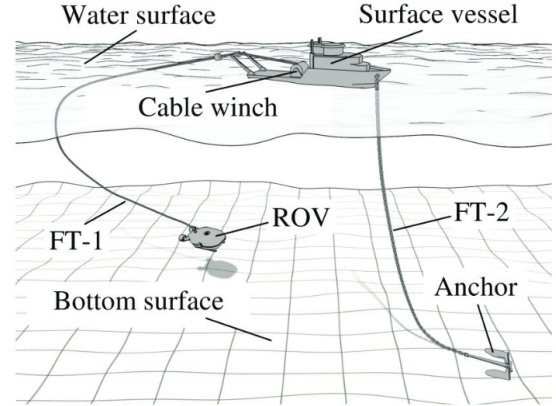zation of the tether by finite elements, whose successive positions over time are calculated by a force-based approach [1], [5]–[9].

The problem induced by this method is that it is essential to know the force that each node applies to its neighbor, which is not necessarily known. Some use the equations of continuum mechanics to determine this force [8], [9], and others propose a method based on a proportional corrector on the erroneous distance between adjacent nodes [1], [5], [7].

The solution proposed in this paper is to propose a behavioral model of this force and to see to what extent the results given by this modeling are physically acceptable. This solution has not been as advanced in the other papers proposing a simulation based on position by discretization [1], [7]. We will therefore be able to use this proposed force model to make a finite element simulation.

## 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 $\mathbf{w}$, the buoyancy, noted $\mathbf{b}$, the force exerted by the previous element on the considered element, noted $\mathbf{f_p}$, as well as that of the next element, noted $\mathbf{f_n}$, and the drag force, noted $\mathbf{d}$

- **Weight** $\mathbf{w}$ : Considering that each element has a mass $m$, and by noting $g$ the standard gravity, we have :

$$\mathbf{w} = \begin{bmatrix} 0 \\ 0 \\ -m.g \end{bmatrix}$$

- **Buoyancy** $\mathbf{b}$ : If we note the volume of each element $V$ and $\rho$ the density of the fluid in which the tether is immersed, we have :

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \rho.V.g \end{bmatrix}$$

- **Tether force** $\mathbf{f_p}$ **and** $\mathbf{f_n}$ : 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:

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

  In these expressions, it is assumed that $\mathbf{u}$ is the unitary vector oriented from the current node to the neighboring node, $e$ 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 :

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

  These two forces $\mathbf{f_p}$ and $\mathbf{f_n}$ can therefore be expressed using the expression of $\mathbf{f}$, taking care to take the correct current and previous element for each case.

- **Drag** $\mathbf{d}$ : By noting $f$ the coefficient of viscous friction, $\mathbf{v}$ the velocity of each node and considering that the frictions are quadratic, we have :

$$\mathbf{d} = -f \cdot \|\mathbf{v}\| \cdot \mathbf{v}$$

Thus we have expressed the forces necessary for the simulation of the tether. FIGURE 2 shows us the modelization of the tether. We see the different TetherElements represented in blue. For clarity reasons, the different forces are represented
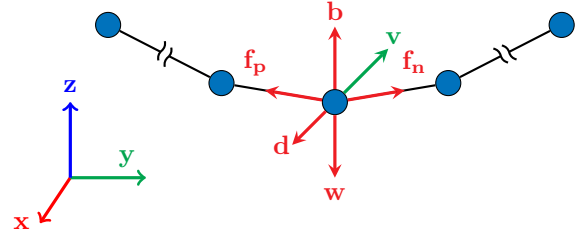


Fig. 2: Modelization of the problem

only on one node but must be computed for each node. The speed of the node is noted $\mathbf{v}$. Finally, in order not to fully draw the tether, the representation has been deliberately cut after the first node and before the last node to represent only three central nodes.

## III. IMPLEMENTATION

The Python 3 implementation of this simulation is available in a GitHub repository[1]. This code is based on Numpy [10], Matplotlib [11] and Scipy [12] packages. 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 $\overrightarrow{F_{t,p}}$ and $\overrightarrow{F_{t,n}}$.

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 have the mass and the volume of each node, but also the length of each link between nodes in order to correctly instantiate each *TetherElement*.

A diagram of these two classes is visible on the FIGURE 3. It respects the UML format and allows to see the different class variables and methods associated to each class.
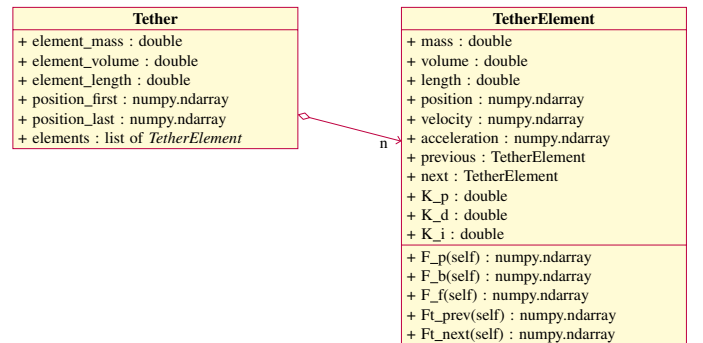


Fig. 3: UML diagram of the *Tether* and *TetherElement* classes

---

[1] Avalable at : https://github.com/Teusner/Tether_Modeling

## 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 [13] [14]. 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 [13] [14] 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 [14], 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 [14]:

$$L = c_1 \cdot sinh\left(\frac{x_n + c_2}{c_1}\right) - c_1 \cdot sinh\left(\frac{x_1 + c_2}{c_1}\right)$$

$$z_1 = c_1 \cdot cosh\left(\frac{x_1 + c_2}{c_1}\right) + c_3$$

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

It is possible to solve the system of equations numerically using the function *fsolve* of the package *scipy.optimize* [12]. 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, as it is done in [5].



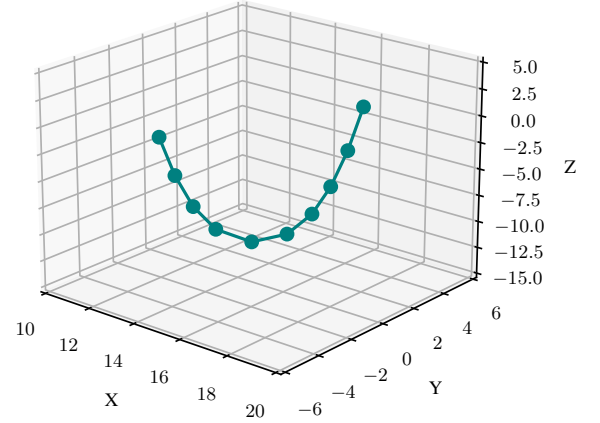Fig. 4: Tether simulation

### 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 behavioral model is correct.

The FIGURE 5 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.
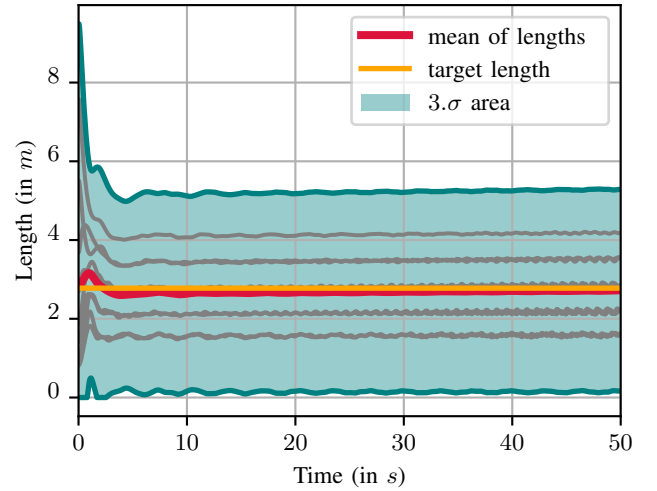


Fig. 5: Length of the links, average length and confidence interval

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 due to the wheight perturbations.

### B. Relative error

The relative error is also very interesting because it makes it possible to be truly aware of the error but also let us easily tune behavioral model's coefficients.

The FIGURE 6 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.
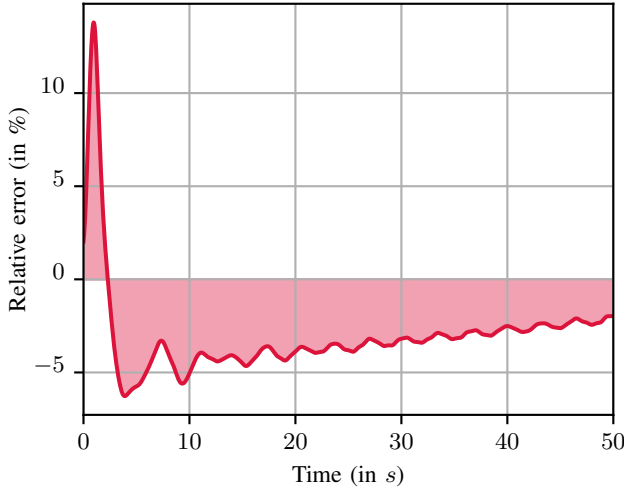


Fig. 6: Relative error between the target length and the mean length of links

### C. Energetical approach

Finally, the energy approach is the most important to validate the simulation, because it determines whether the simulation makes physical sense. 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.

The mechanical energy of the system decomposes into the sum of two energies: kinetic energy and potential energy [?].

- **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.
The FIGURE 7 presents the evolution of the kinetic energy of the system, with in gray the kinetic energy of each node, in crimson the average of the kinetic energies and the blue area represents the 95% confidence interval. We can see that the kinetic energy decreases and cancels rapidly. This is explained by the fact that the Tether after
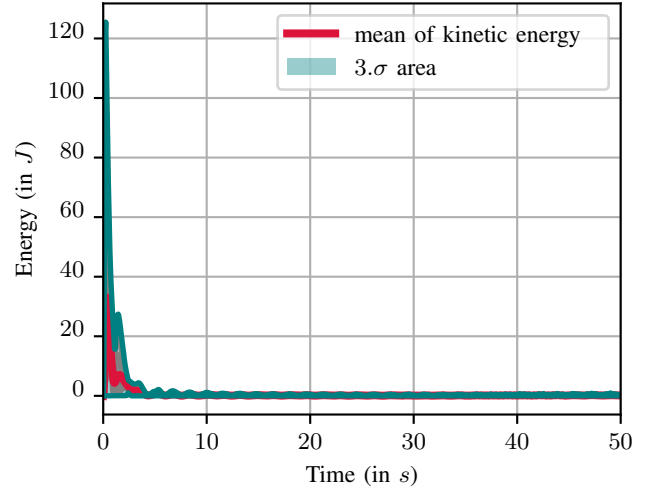


Fig. 7: Kinetic energy of the system

a few seconds is correctly initialized and the nodes come to a standstill.

- **Potential Energy** : To calculate the potential energy related to the application of a force on a solid, it is necessary to use the definition of potential energy [?]. Indeed we know that by noting $dt$ the time step of the simulation :

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

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

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, which is set to 0 in our case.

The FIGURE 8 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.

- **Mechanical Energy** : Finally, by summing the kinetic and potential energies previously calculated, we are able to get 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 9 shows us the evolution of the mechanical energy over time. In gray is plotted the mechanical energy
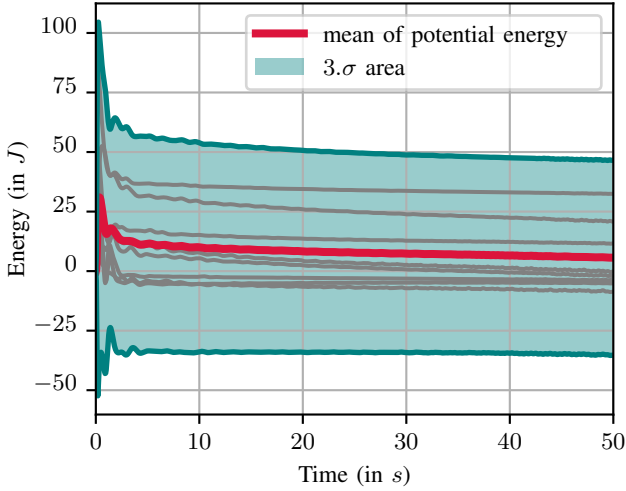
Fig. 8: Potential energy of the system over the time

of each node, in crimson is the average mechanical energy and the blue area represents the 95 % confidence interval.
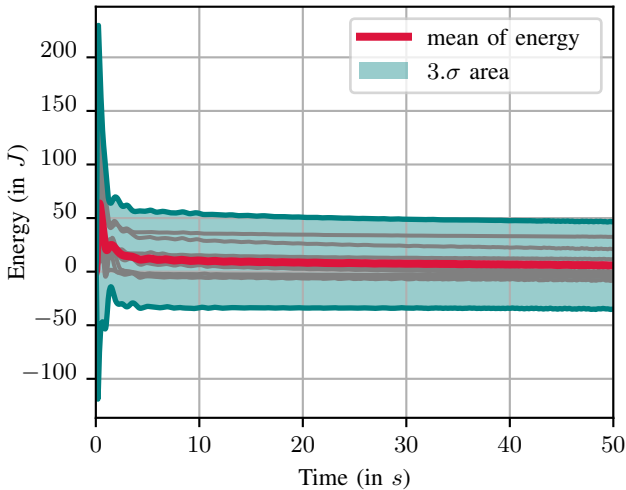


Fig. 9: Energy of the system

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

## VI. CONCLUSION

In conclusion, the tether modeling as presented in this 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. A solution seems to be presented in [**?**]. Finally, there is no force that simulates the stiffness of the Tether, but it is not infinitely flexible.

## REFERENCES

[1] O. Blintsov, "Development of the mathematical modeling method for dynamics of the flexible tether as an element of the underwater complex," 2017.
[2] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011. [Online]. Available: https://books.google.fr/books?id=oR3sBgAAQBAJ
[3] L. Jaulin, *Mobile Robotics*. Wiley, 2019. [Online]. Available: https://books.google.fr/books?id=OMWxDwAAQBAJ
[4] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, and M. Macklin, "A survey on position-based simulation methods in computer graphics," *Comput. Graph. Forum*, vol. 33, no. 6, p. 228–251, Sep. 2014. [Online]. Available: https://doi.org/10.1111/cgf.12346
[5] J. R. Ellis, "Modeling, Dynamics, and Control of Tethered Satellite Systems," Ph.D. dissertation, Virginia Polytechnic Institute and State University, The address of the publisher, 03 2010. [Online]. Available: http://scholar.lib.vt.edu/theses/available/etd-03182010-130812/
[6] R. Marshall, R. Jensen, and G. Wood, "A general newtonian simulation of an n-segment open chain model," *Journal of Biomechanics*, vol. 18, no. 5, pp. 359–367, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002192908590291X
[7] O. Ganoni, R. Mukundan, and R. Green, "Visually realistic graphical simulation of underwater cable," jan 2018.
[8] J. Koenemann, P. Williams, S. Sieberling, and M. Diehl, "Modeling of an airborne wind energy system with a flexible tether model for the optimization of landing trajectories **Support by the EU via ERC-HIGHWIND (259 166), ITN-TEMPO (607 957), and ITN-AWESCO (642 682) and by DFG in context of the Research Unit FOR 2401." *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 944–11 950, Jul. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317315227
[9] S. Prabhakar and B. Buckham, "Dynamics modeling and control of a variable length remotely operated vehicle tether," in *Proceedings of OCEANS 2005 MTS/IEEE*, 2005, pp. 1255–1262 Vol. 2.
[10] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2
[11] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
[12] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
[13] A. S. of Mechanical Engineers, *Applied Mechanics Reviews*. American Society of Mechanical Engineers, 1968, no. vol. 21. [Online]. Available: https://books.google.fr/books?id=00muEpyMcn0C
[14] W. Ren, M. Huang, and W. Hu, "A parabolic cable element for static analysis of cable structures," *Engineering Computations*, vol. 25, no. 4, pp. 366–384, May 2008. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/02644400810874967/full/html