Recursive inverse dynamics of a swimming snake-like robot with a tree-like mechanical structure

Xiao Xie, Johann Herault, Vincent Lebastard, Frédéric Boyer

Recursive inverse dynamics of a swimming snake-like robot with a tree-like mechanical structure

Xiao Xie[1], Johann Herault[1,2], Vincent Lebastard[1], Frédéric Boyer[1]

***Abstract*—In this paper, we report a recursive inverse dynamical model for a new snake-like robot called NATRIX. This robot has been designed to maintain its gaze on the water surface and monitor sensible ecosystems. Inspired by real snakes, the robot features rotating outer shells allowing to change the level of immersion of each module and re-stabilize quickly the robot. This new degree of freedom leads to an original tree-like geometric structure. We present here the theoretical model and the numerical solutions that allow us to simulate in real time the dynamics of the robot on the water surface. After reporting the benchmark of the simulator, we present surprising preliminary results suggesting the possibility of capsizing for a given frequency range.**

I. Introduction

More than 70% of the world's population lives near coastal regions which induces great anthropogenic pressure on ecosystem. Nutriment introduction and chemical pollution over the past century has led to ecological collapse and destabilizing biospheres. To face this challenge, marine surface robotics is one of the most promising solutions [1, 2]. However, unmanned surface vehicles (USV) with propellers can deeply perturb sensitive ecosystems causing strong turbulence and mud flocs resuspension [2, 3].

This point has motivated the development of bio-inspired robots taking inspiration from the fishes or reptiles to limits their impact on ecosystems, such as swimming snake-like robots [4, 5, 6, 7]. Unfortunately, after twenty years of intense development, snake-like robots have not been as successful as expected due to severe impediments.

The stability of robots on the water surface is one of the current limitations to their deployment. Because of their slender morphology, these robots have a low axial inertia and are consequently very sensitive to rolling perturbation on surface [5, 8]. This rotational motion could significantly perturbs the swimming performance as well as the vision guidance. In the worst case, the robot can capsize. How can these robots be actively stabilised? Yet, we don't know.

To face this challenge, we proposed a new bio-inspired snake robot called *Natrix* (see Fig. 1) that features new technologies enabling active gaze control [9]. *Natrix* stands for New Advanced Technoly for Robot Involved in the eXploration and takes inspiration from cottonmouthes [8], semi-aquatic snakes exhibiting an exceptional sense of balance at surface.

To control the stability of the robot, those semi-aquatic snakes modify in real time the immersion level of each body section [8]. These three-dimensional deformations are composed of lateral and dorso-ventral bendings as well as some torsion. However, we are still far from managing such complexity of deformation. Moreover, we prefer to keep a planar structure of the robot in order to preserve the efficiency of the undulatory propulsion.

Therefore, we took as a starting point a serial structure, which drives the lateral undulation, on which is superimposed external shells. The rotation of the shells with triangular-like cross-section can modify the level of immersion as well as the rotational inertia without changing the planarity of the robot [9]. Besides, our robot can lift its head out of the water thanks to three degrees of freedom located in its neck (see Fig. 2). The conception of the robot is still under development, and preliminary tested has been performed (cf Fig. 1). Thanks to these new improvements, we aim at producing new strategies of control to stabilize the orientation and elevation of the head. Before, we need to develop the dynamical model and the corresponding simulator paving the way to active control of the gaze.

In previous studies [9, 10], we developed quasi-static feed-forward controllers enabling to deform the robot geometry while keeping its gaze constant. Furthermore, we proved that the static stability can be improved by exploiting the external rotating shells. However, as soon as the robot starts moving in real conditions, it naturally exhibits 3D oscillations around an equilibrium due to its inertia. Such effect was not captured by our previous simulator. Hence, the present paper aims at extending our previous work by considering the complex 3D inertial dynamics of the multi-body system subject to hydrostatic forces.

Fig. 1. First swimming test of *Natrix* on the water surface.

Here, we choose to develop our own simulator based on a Newton-Euler recursive algorithm with an exact 3D geometric representation of each module. Unlike previous serial swimming robots [5, 11, 12], *Natrix* exhibits a tree-like structure due to the rotating external shells, which requires to extend the previous models [5, 11].

The most significant contributions of this paper are (I) a theoretical extension of the Newton-Euler recursive algorithm with a tree-like architecture coupling rotational and flexural

[1] IMT Atlantique, LS2N, UMR CNRS 6004, F-44307 Nantes, France.
[2] johann.herault@imt-atlantique.fr

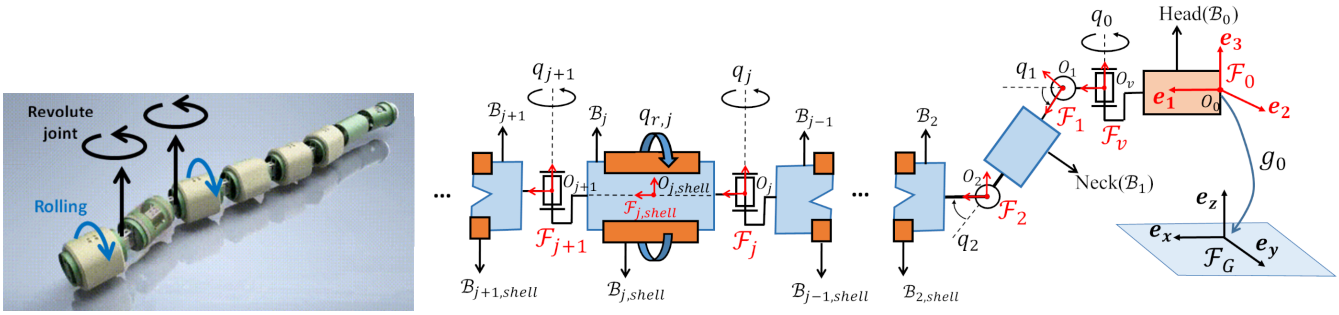

Fig. 2. Left: schematic diagram of the robot's movements. Right: Illustration of the geometric parametrization of each module (see text for details)

motion; (II) the development of the simulator and its benchmark; and (III) a first investigation of stability for undulatory swimming revealing a surprising capsizing phenomenon.

II. Parametrization and Geometrical Model of the Robot NATRIX

The robot is composed of 14 rigid bodies all actuated by one servomotor (see Fig.2), including a head, a neck and six body modules each with an autonomous rotating outer shell. The real robot features a passive tail (cf Fig.1) that is yet not included in our model.

Hence, nine serial rigid bodies noted $\mathcal{B}_j$ form the robot backbone (cf. Fig.2), with $j = 0$ the head, $j = 1$ the neck, and $j \in [2, 3, ..., 8]$ for the serial body modules. The outer shells $\mathcal{B}_{j,shell}$ with $j \in [2, 3, ..., 8]$ lead to a tree-like structure mounted on the serial structure (see Fig. 3). The configuration of each $\mathcal{B}_j$ is given by a local frame $\mathcal{F}_j$ of origin O_j (cf Fig. 2). For the body module, this origin coincides with the center of rotation in the servomotor whose rotation leads to the change of orientation of the reference frame $\mathcal{F}_j$. The motion of the floating frame $\mathcal{F}_0$ located in the head characterises the solid body motion of the robot with respect to the Galilean frame $\mathcal{F}_G$.

The relative configuration of each module is determined by the joint coordinates $\mathbf{q} \in \mathcal{S} = (S^1)^{14}$. The head and the neck is connected by a two-axes servomotor with respective angles amplitudes q_0 (yaw) and q_1 (pitch). To model it, we introduce a mass-less body $\mathcal{B}_v$ associated to $\mathcal{F}_v$ that provides the mechanical linkage. Then the joint with angle q_2 connects the neck to the rest of the body as illustrated on Fig.2. It is worth noting that the pitching joints q_1 and q_2 allow for lifting the head while keeping its orientation constant as a pan tilt for camera. The planar undulation is provided by five yaw joints (q_j) with $j \in [3, 4, ..., 7]$, while the local rolling motion of the j-th shell is denoted $(q_r)_j$ $(j \in [2, 3, ..., 7])$.

The configuration space of the robot is fully determined by the couple $(g_0, \mathbf{q}) \in C$ with $C = \text{SE}(3) \times \mathcal{S}$ [9]. The element $g_0 \in \text{SE}(3)$ characterizes the configuration of the head frame $\mathcal{F}_0$ (i.e. the floating base) with respect to the surface frame $\mathcal{F}_G$. Its homogeneous representation reads

$$g_0 = \begin{pmatrix} R_0 & \mathbf{p}_0 \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \in \text{SE(3)} \tag{1}$$

where $R_0(t)$ and $\mathbf{p}_0(t)$ are the rotation matrix and the position vector. Now, to locate each module along the backbone in the global frame $\mathcal{F}_G$, we introduce its configuration g_j given by

$$g_j = g_0 \left({}^0g_v\right) \left({}^vg_1\right) ... \left({}^{j-1}g_j\right) \tag{2}$$

with $j = [1, 2, ..., 8]$. Reciprocally, the shell configuration $(g_{shell})_j$ mounted on the j-th body can be computed thank to the relative configuration h_j where $(g_{shell})_j = g_j h_j$.

TABLE I
Denavit–Hartenberg parameters of NATRIX

Frame	$\alpha_j [deg]$	$\phi_j [deg]$	$a_j [mm]$
$\mathcal{F}_0$ to $\mathcal{F}_v$	0	q_0	100
$\mathcal{F}_v$ to $\mathcal{F}_1$	-90	q_1	0
$\mathcal{F}_1$ to $\mathcal{F}_2$	90	q_2	105
$\mathcal{F}_{j-1}$ to $\mathcal{F}_j$, $j \in [3, 4, ..., 7]$	0	q_j	170
$\mathcal{F}_j$ to $\mathcal{F}_{j,shell}$, $j \in [2, 3, ..., 7]$	$q_{r,j}$	0	85

Each relative configurations ${}^{j-1}g_j$ and h_j depends solely on a corresponding joint component that is parameterized by the modified Denavit–Hartenberg convention [13], given by

$${}^{j-1}g_j \ , \ h_j = \begin{pmatrix} \cos\phi_j & -\sin\phi_j & 0 & a_j \\ \sin\phi_j \cos\alpha_j & \cos\phi_j \cos\alpha_j & -\sin\alpha_j & -b_j \sin\alpha_j \\ \sin\phi_j \sin\alpha_j & \cos\phi_j \sin\alpha_j & \cos\alpha_j & b_j \cos\alpha_j \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with the Denavit–Hartenberg's parameters α_j, ϕ_j, a_j reported in Tab.I ($b_j = 0$).

III. Dynamical Model and Computational Aspects

In most of the previous works (see for instance [5, 12]), the motion of the robots has been assumed planar. Here, we extend these models to 3D motion to include the heave, roll and pitch. The 3D dynamical model of snake-like robots can be written into the very general Lagrangian form [5, 11]

$$\begin{pmatrix} M_0 & M_q \\ M_q^T & \mathcal{M} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\eta}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\beta}_b \end{pmatrix} = \begin{pmatrix} \mathbf{W}_{stat} \\ \mathbf{w}_{stat} \end{pmatrix} + \begin{pmatrix} \mathbf{W}_{dyn} \\ \mathbf{w}_{dyn} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\Gamma} \end{pmatrix} \tag{3}$$

The left-hand side terms results from the momentum conservation while the right-hand side terms correspond to the generalized external forces. In this paper, we will focus on the first row that stands for the head dynamics. Its velocity $\boldsymbol{\eta}_0^T = (\mathbf{V}_0^T, \boldsymbol{\omega}_0^T)^T \in \mathbb{R}^6$ is expressed in the head frame $\mathcal{F}_0$,

and is given by the twist components of the skew-symmetric matrix $\hat{\boldsymbol{\eta}}_0 \in \mathrm{se}(3)$ where [14]

$$\hat{\boldsymbol{\eta}}_0 = g_0^{-1}\dot{g}_0. \tag{4}$$

The matrices $\mathcal{M}(\mathbf{q}) \in \mathbb{R}^{6\times 6}$ and $M_q(\mathbf{q}) \in \mathbb{R}^{6\times n}$ in equation 3 correspond respectively to the generalized inertia of the head (for a frozen body shape) and to the coupled inertia between the head and the joints. The vector $\boldsymbol{\beta}(\boldsymbol{\eta}_0, \mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^6$ contains all the quadratic terms corresponding to the Coriolis and centrifugal forces. The wrench $\mathbf{W}_{stat}(g_0, \mathbf{q}) \in \mathbb{R}^6$ is the resultant of the gravity and buoyancy forces that we call hydrostatic wrench [15].

For the sake of generality, we have also included hydrodynamic forces $\mathbf{W}_{dyn}$ that will not be considered in this study. Unlike underwater locomotion, surface swimming requires consideration of the influence of confined flow dynamics and waves. A major effect is the dependence of the added mass with respect to the body configuration g_j and the beat frequency [15]. Reciprocally, waves can lead to a drag. So far, we do not know how to integrate such effects into concise and real time models. Indeed, to our knowledge, there is no theoretical model for 3D propulsion at the water surface that takes into account surface dynamics. In a first approach, we omit these complex effects that requiring experimental and numerical investigations that are beyond the scope of this study.

The robot motion is achieved by a position control of the joint $\mathbf{q}(t)$ and we assume that the servomotors can provide the expected joint trajectories $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. Consequently, the robot's dynamics consists in solving the first line of equation 3, i.e. to compute $\dot{\boldsymbol{\eta}}_0$ as a function of $(g_0, \boldsymbol{\eta}_0)$ and $\mathbf{q}$ and its derivative. It is worth noting that we can also compute the required torque $\boldsymbol{\Gamma}$ to drive the motion, all these computations realise the inverse dynamic model.

Going into further details, the resolution of the inverse dynamics consists in updating the new head state $(g_0, \boldsymbol{\eta}_0)_{(t+dt)}$ from the current acceleration $(\dot{\boldsymbol{\eta}}_0)_t$ depending on the state $(g_0, \boldsymbol{\eta}_0)_t$ and the current joint trajectories $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})_t$. First, $(\dot{\boldsymbol{\eta}}_0)_t$ is computed via a recursive Newton-Euler (N-E) algorithm detailed in section IV. Then, the acceleration $(\dot{\boldsymbol{\eta}}_0)_t$ is integrated to obtain $(\boldsymbol{\eta}_0)_{t+dt}$. Finally, the head configuration $(g_0)_{t+dt}$ is determined from $(\boldsymbol{\eta}_0)_{t+dt}$ by a geometric integrator in which the rotation matrix $\mathrm{R}_0(t)$ matrix (see equation 1) is computed with unit quaternions [11].

The simulator has been developed in Python. The time-integration is based on a fourth-order Runge-Kutta integrator with a time step of typically $1ms$. Gauss quadrature has been used to spatially integrate the immersed volume and compute the buoyancy wrench along the segments of robot. For a simulation corresponding to 1s physical time, the computational time is $0.12s$ when the buoyancy is neglected, but increases up to $1.23s$ when it is considered. Our calculation are performed with a laptop with intel core i5-8365U CPU, 1.60 GHz and 16.0 Gb of xRAM.

IV. Newton-Euler algorithm and head motion

The recursive Newton-Euler (N-E) algorithm was initially applied to rigid manipulators [16], generalized to floating-base systems [17] and extensively applied to bio-inspired locomotion [18]. Unlike manipulators, the acceleration of the first module of the robot, i.e. the floating base of the locomotor, is not known and must be computed. We will see that this point requires further theoretical developments. Besides, *Natrix* differs from previous swimming robots [5] by its additional external shells that lead to extra terms in the recursions.

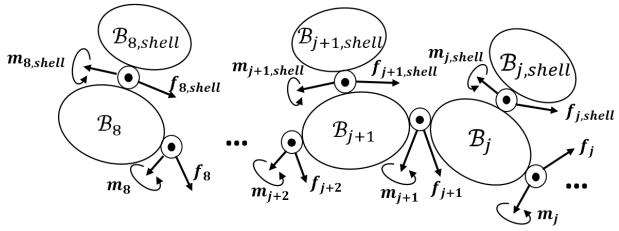

Fig. 3. Tree-like structure with the wrenches exerted on the linkages.

For standard N-E algorithms, the recursive inverse dynamics is composed of three sequential recursive loops: two forward loops and one backward loop. To compute the head acceleration $\dot{\boldsymbol{\eta}}_0$, only the first forward and backward loops are required. The third one aims at computing the required torques provided by the servomotors to produce $\mathbf{q}(t)$ and will not be reported in the present paper.

In this process, we will need three recursive equations. First, the velocity $\boldsymbol{\eta}_j$ and acceleration $\dot{\boldsymbol{\eta}}_j$ of j-th body $\mathcal{B}_j$ can be recursively computed. Starting from $j = 0$ (the head) to the last body, the velocity of each module is given by the recursive equation

$$\boldsymbol{\eta}_j = \mathrm{Ad}_{j\,g_{j-1}}\boldsymbol{\eta}_{j-1} + \mathbf{A}_j\dot{q}_j, \tag{5}$$

with $\mathbf{A}_j = (\mathbf{0}_3^T, \mathbf{e}_j^T)^T$, where $\mathbf{e}_j$ is the unit vector along which the j-th joint rotates. Similarly, the acceleration is transferred from one module to another by the following recursion

$$\dot{\boldsymbol{\eta}}_j = \mathrm{Ad}_{j\,g_{j-1}}\dot{\boldsymbol{\eta}}_{j-1} + \boldsymbol{\zeta}_j + \mathbf{A}_j\ddot{q}_j. \tag{6}$$

The term $\boldsymbol{\zeta}_j$ comes from the time variation of the axis of rotation and reads $\boldsymbol{\zeta}_j = \mathrm{ad}_{\boldsymbol{\eta}_j}\mathbf{A}_j\dot{q}_j \in \mathbb{R}^6$. In both equation, we have used the adjoint representation of the Lie group $\mathrm{Ad}_g \in \mathrm{SE(3)}$ and the Lie algebra $\mathrm{ad}_{\boldsymbol{\eta}_j} \in \mathbb{R}^{6\times 6}$ [14], that are defined by

$$\mathrm{Ad}_g = \begin{pmatrix} R & \hat{\mathbf{p}}R \\ 0_{3\times 3} & R \end{pmatrix}, \quad \mathrm{ad}_{\boldsymbol{\eta}_j} = \begin{pmatrix} \hat{\boldsymbol{\Omega}}_j & \hat{\mathbf{V}}_j \\ 0_{3\times 3} & \hat{\boldsymbol{\Omega}}_j \end{pmatrix}. \tag{7}$$

From Newton's law and Euler's theorem, one can state a recursive backward equation (from the last module) such that

$$\begin{aligned} &\mathbf{F}_j - \mathrm{Ad}^T_{j+1\,g_j}\mathbf{F}_{j+1} - \mathrm{Ad}^T_{h_j}\mathbf{F}_{j,shell} + \mathbf{F}_{ext,j} \\ &= \mathcal{M}_j\dot{\boldsymbol{\eta}}_j + \boldsymbol{\beta}_j(\boldsymbol{\eta}_j, \dot{q}_j) \end{aligned} \tag{8}$$

where the wrench $\mathbf{F}_j^T = (\mathbf{f}_j^T, \mathbf{m}_j^T)$ is composed of the linear force $\mathbf{f}_j$ and torque $\mathbf{m}_j$ that are exerted by the body $\mathcal{B}_{j-1}$ onto $\mathcal{B}_j$ (see Fig.3). This equation holds only for the module along the serial structure, not for the shell module. Reciprocally, $\mathbf{F}_{j,shell}$ is the wrench exerted by the j-th local shell onto $\mathcal{B}_j$, and $\mathbf{F}_{ext,j}$ the wrench of external force (such as gravity).

In equation 8, the right-hand side terms correspond to the inertial terms composed of the Coriolis and centrifugal forces $\boldsymbol{\beta}_j = -ad^T_{\boldsymbol{\eta}_j}\boldsymbol{\eta}_j$ and $\mathcal{M}_j\dot{\boldsymbol{\eta}}_j$. Here, the local matrix of inertia $\mathcal{M}_j$ expressed in $\mathcal{F}_j$ reads

$$\mathcal{M}_j = \begin{pmatrix} m_j I_3 & -m_j\hat{\mathbf{S}}_j \\ m_j\hat{\mathbf{S}}_j & J_{Gj} - m_j\hat{\mathbf{S}}_j\hat{\mathbf{S}}_j \end{pmatrix} \tag{9}$$

with $\mathbf{S}_j \in \mathbb{R}^3$ the coordinate of center of mass in $\mathcal{F}_j$, and J_{Gj} the matrix of moment of inertia at the center of mass. For the rest of the paper, we will lighten the notations by integrating the external forces $\mathbf{F}_{ext,j}$ in the Coriolis and centrifugal forces by making the change of variable $\boldsymbol{\beta}_j - \mathbf{F}_{ext,j} \mapsto \boldsymbol{\beta}_j$ as in [11].

Before going into the details of the body index loops, we can already identify in equations 5-8 the difficulty of our problem. Indeed, our ultimate goal is to establish Newton's law and Euler theorem for the head module:

$$\dot{\boldsymbol{\eta}}_0 = \mathcal{M}_0^{-1}\left(-\boldsymbol{\beta}_\mathbf{0} - \mathrm{Ad}^T_{0_{g_1}}\mathbf{F}_1\right) \tag{10}$$

Ideally, the wrench $\mathbf{F}_1$ is deduced from the backward recursive equation 8. However, at each recursion both terms $\mathbf{F}_j$ and $\dot{\boldsymbol{\eta}}_j$ are unknown in equation 8, and $\mathbf{F}_j$ can't be computed. Unlike manipulators, the acceleration $\dot{\boldsymbol{\eta}}_j$ can't be updated in the forward recursion since we seek to compute $\dot{\boldsymbol{\eta}}_0$. To overcome this issue, we will reformulate the equations 8 and 10 in the recursive backward loop.

The Forward loop. In this first loop, we compute all the quantities that are independent of $\dot{\boldsymbol{\eta}}_0$. Hence, we start by the geometrical model of the robot presented in section II. Then, the velocities $\boldsymbol{\eta}_j$ can be updated thanks to equation 5. At this step, some elements of the equation 6 and 8 can be computed. Hence, the contribution of the servomotor to the body acceleration given by $\boldsymbol{\zeta}_j + \mathbf{A}_j\ddot{q}_j$ is estimated. Besides, the Coriolis and centrifugal forces and the external forces stacked in the vector $\boldsymbol{\beta}_j$ depend solely on the robot velocities and the configuration of each $\mathcal{B}_j$ with respect to $\mathcal{F}_G$ (cf buoyancy and gravity forces).

The backward loop. Since equation 8 can't be directly used under its current form, we propose to rewrite an analogue equation relating $\boldsymbol{F}_j$ to the force and acceleration $\dot{\boldsymbol{\eta}}_j$. To do so, we introduce a modified inertia $\mathcal{M}_j^*$ and a generalized force $\boldsymbol{\beta}_j^*$ to obtain the new momentum balance deduced from equation 8

$$\boldsymbol{F}_j = \mathcal{M}_j^*\dot{\boldsymbol{\eta}}_j + \boldsymbol{\beta}_j^* \tag{11}$$

Here, equation 6 has been written backward in order to cascade sequentially the inertia and the Coriolis and centrifugal forces from the last to the first body along the backbone. Hence, at each new recursion the term $\mathbf{F}_{j+1}$ will only depend on the generalized inertia $\mathcal{M}_{j+1}^*$ and force $\boldsymbol{\beta}_{j+1}^*$. This process leads to an algorithm (cf section 4.2 of [5]) where the terms $\mathcal{M}_j^*$ and $\boldsymbol{\beta}_j^*$ are computed recursively along the body modules, such that

$$\begin{aligned} \mathcal{M}_j^* &= \mathcal{M}_j + \mathrm{Ad}^T_{j+1_{g_j}}\mathcal{M}_{j+1}^*\mathrm{Ad}_{j+1_{g_j}} + \mathrm{Ad}^T_{h_j}\mathcal{M}^*_{j,shell}\mathrm{Ad}_{h_j}. \\ \beta_\mathbf{j}^* &= \beta_\mathbf{j} + \mathrm{Ad}^\mathbf{T}_{\mathbf{j+1_{g_j}}}\left[\beta^*_\mathbf{j+1} + \mathcal{M}^*_\mathbf{j+1}\left(\boldsymbol{\zeta}_\mathbf{j+1} + \mathbf{A_{j+1}}\ddot{\mathbf{q}}_\mathbf{j+1}\right)\right] \\ &\quad + \mathrm{Ad}^T_{h_j}\left[\beta_\mathbf{j} + \mathcal{M}_\mathbf{j}\left(\boldsymbol{\zeta}_\mathbf{j} + \mathbf{A_j}\ddot{\mathbf{q}}_\mathbf{j}\right)\right]_{shell}, \end{aligned} \tag{12}$$

For $j \in [2, 3, ..., 8]$, the bracket $[-]_{shell}$ contains all the terms associated with the j-th shell. It is worth recalling that the terms $\beta_\mathbf{j}$ take into account the wrench exerted by external forces. For the last modules without an external shell (neck+head), we omit the components related to the shell.

Finally, one obtains the head acceleration given by

$$\dot{\boldsymbol{\eta}}_0 = -(\mathcal{M}_0^*)^{-1}\left(\boldsymbol{\beta}_\mathbf{0}^*\right), \tag{13}$$

with $\mathcal{M}_0^*$ positive definite and thus invertible. This equation is the recursive version of the first row of the dynamic model 3 with $\mathcal{M}_0^* = \mathcal{M}$. Note that the third (forward) loop can be then performed to obtain the torques $\boldsymbol{\Gamma}$, a step not reported here.

If one is only interested in the head motion, a gain of computation time can be obtained by computing directly the inertia matrix and the net force exerted on the robot head without performing the full backward loop. Hence, the inertia matrix can be estimated by expanding the embedded sum in equation 12

$$\mathcal{M} = \mathcal{M}_0 + \sum_j \mathrm{Ad}^T_{j_{g_0}}\mathcal{M}_j\mathrm{Ad}_{j_{g_0}} + \sum_j \mathrm{Ad}^T_{j_{h_0}}(\mathcal{M}_{shell})_j\mathrm{Ad}_{j_{h_0}} \tag{14}$$

with ${}^jg_0$ and ${}^jh_0$ the relative configurations of the head frame with respect to the body module and its shell. The second inertia matrix takes the form

$$M_q = \left(\mathrm{Ad}^T_{v_{g_0}}\mathcal{M}_v^*\mathbf{A}_v, ..., \mathrm{Ad}^T_{j_{g_0}}\mathcal{M}_j^*\mathbf{A}_j, ..., [\mathrm{Ad}^T_{8_{h_0}}\mathcal{M}_8^*\mathbf{A}_8]_{shell}\right) \tag{15}$$

Reciprocally, the pseudo-forces acting on the head in equation 3 are given by

$$\boldsymbol{\beta} = \boldsymbol{\beta}_0 + \sum_{j=1}^{8}\mathrm{Ad}^T_{j_{g_0}}\left(\boldsymbol{\beta}_j + \mathcal{M}_j^*\boldsymbol{\zeta}_j\right) + \sum_{j=2}^{8}\mathrm{Ad}^T_{j_{h_0}}\left[\boldsymbol{\beta}_j + \mathcal{M}_j^*\boldsymbol{\zeta}_j\right]_{shell} \tag{16}$$

An other advantage for computing the terms is to identify the Lagrangian model of the robot and to perform linear analysis around equilibrium configurations (see [9]).

V. Benchmarks and preliminary results

A. Benchmarks

Before presenting our first results, we want to make sure that physical quantities such as energy and momentum are preserved. First, we will test whether the solid motion of the robot (i.e. without deformation) preserves the total energy, a quantity directly related to the accuracy of the temporal and spatial integration schemes and the mechanical consistency of the model. In the second case, we check the conservation of momentum during the robot deformation.

Hence, we start with a set-up where all joint positions are fixed and the head motion is simulated. Since the gravity and

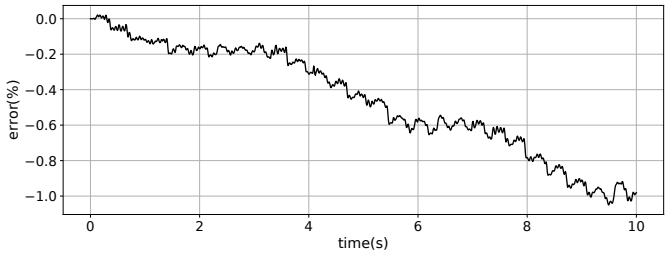

Fig. 4. Time-Evolution of the relative energy loss during the robot oscillations around an equilibrium in a frozen body shape.

buoyancy forces are the only source of external forces in this case, the system is conservative. The relative variation of total energy (potential+kinetic) of the system is then calculated to quantify the accuracy of the simulator. For the initial conditions, we choose to initiate a small rolling angle from various local equilibrium.

We have reported in Fig.4 a simulation where the robot is frozen on a cosine-like shape (see for instance [9]). One observes two time-scales of energy variation. The fastest time scale is due to the physical oscillations of the robot around the equilibrium configuration with typical value around 0.4s. Note that the energy frequency is twice the motion frequency. The slowest one is the energy decrease caused by the intrinsic scheme inaccuracy such that the system looses 1% of its initial energy after 10s, i.e typically 15 oscillations around an equilibrium. This is a good trade-off between the simulation speed and the energy conservation.

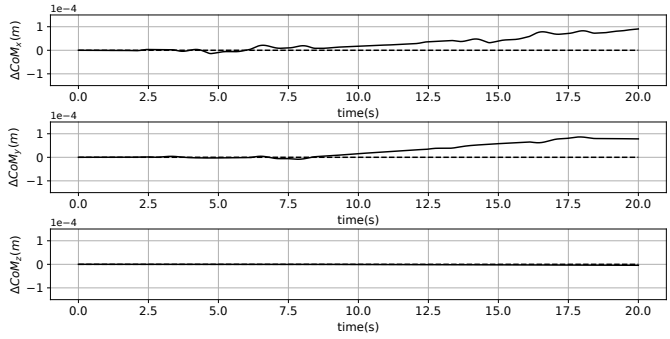

Fig. 5. Time-Evolution of the displacement of the center of mass $\Delta CoM = (\Delta CoM_x, \Delta CoM_y, \Delta CoM_z)^T$ during 20s in $\mathcal{F}_G$ with $t_f = 10s$ (see equation 17)

For the second benchmark, no external force is applied on the robot, and only the inertial terms are considered. In this case, it is expected that the linear momentum of the robot is conserved [19] during its body deformation. To verify this property, we propose the following joint trajectory for a uniform lateral bending $q_j(t)$ with time-evolution for an half-period $t \in [0, t_f]$

$$f(t) = \frac{1}{160}\left(0.25\ t^4 - 0.5\ t_f t^3 + 0.25\ {t_f}^2 t^2\right) \qquad (17)$$

and then $f(t) \mapsto -f(t - t_f)$ for the second half period $t \in (t_f, 2t_f]$. The function $f(t)$ is a polynomial function of time that allows to start and finish at $q_j = 0$ and $\dot{q}_j = 0$ after a half period $t = t_f$, then from t_f to $2t_f$, we apply the reverse motion. Here, we fix $q_r = 0$. The robot will thus bend and then recover its initially straight configuration. The corresponding position of the center of mass is reported on Fig. 5 during the robot bending. We observe that the displacement of the center of mass remains small with 1×10^{-4}m compared with the robot length ($L = 1.24m$) and thus the linear momentum is almost conserved.

B. Rolling motion vs undulatory swimming

In this section, we report the first study of the impact of swimming onto stability by considering the coupling between inertial effects and hydrostatic forces as the dominant effects. More precisely, we analyse the rolling motion exhibited by the robot when it performs the undulatory swimming. This gait consists in propagating a lateral deformation wave from the head to the tail, and can be generated by the following body deformation law without torsion ($q_r = 0$)

$$q_j = Qe^{\alpha s_j}.\sin\left[2\pi\left(\frac{ns_j}{L} - F.t\right)\right] \text{ for } j = [0, 3, ..., 8] \quad (18)$$

with $n = 1.5$ the relative spatial frequency, F the temporal frequency, L the robot length, and Q the bending amplitude. To model realistic snake swimming [20], we have considered an exponential wave envelop with $\alpha = 0.1\text{m}^{-1}$. Here, t and s_j are the time and the curvilinear position of j-th module along robot's backbone at rest ($s_j = 0$ is the snout).

First, we investigate the response of the robot for a frequency range $F \in [0.5 - 3]$Hz and a constant amplitude $Q = 0.25$ rad. The resulting rolling amplitude θ_x of the robot head is determined from the Euler angles expressed in the frame $\mathcal{F}_G$. Then the standard deviations are reported on Fig.6.

At low and high frequencies (black circles), the non-linear response of the rolling motion (Figure A) increases as the frequency approach the frequency range $[1.5 - 2]$Hz. This behaviour suggests a resonant mechanism. Then, the rolling motion exhibit beat, i.e. the superimposition of two components with very close frequencies (Figure B, triangles) similar to interference. Surprisingly, one observes an instability (Figure C) in the frequency range $F \in [1.33, 1.41]Hz$. In this regime, the rolling angle reaches an amplitude of 180 deg. Since our measure is modulo 180 deg, this motion is actually unbounded which implies that the robot capsizes. Concerning the origin of the instability, the beats suggest the presence of a progressive frequency locking of two components, which is known to be a precursory element of an instability [21]. Yet, we don't know precisely the underlying mechanism of this original instability displayed by the robot.

We then perform a parametric investigation of the amplitude-frequency (Q, F) plane. The instability is still present for various amplitude Q (stars symbols) and always occurs in a finite frequency range in between beat regimes (dotted area). As the amplitude increases, there is a shift of this band to higher frequencies. This figure is similar to stability diagrams exhibiting Arnold's tongues [22], confirming our intuition that a synchronization phenomenon is at play. We

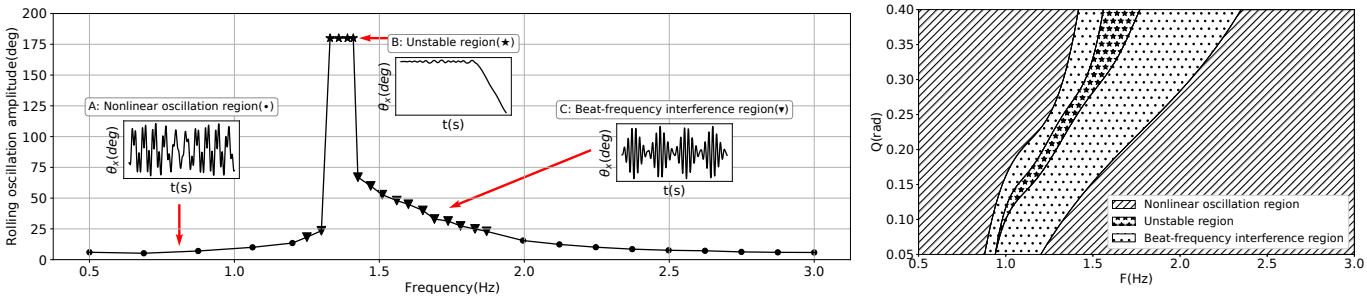

Fig. 6. Left: rolling oscillation amplitude as a function of F with $Q = 0.25rad$; circles correspond to nonlinear oscillations; triangles to the beat phenomenon; stars to the robot capsizing. Right: parametric investigation of the rolling motion in the $(Q - F)$ plane with $F \in [0.5 - 3]Hz$ and $Q \in [0.05 - 0.4]rad$.

may also notice that these frequency ranges are relevant for swimming (see [5]) and can be explored by the robot. Further investigations are required to better characterize this phenomenon.

VI. Conclusion

In this paper, we have reported the design of new snake like robot with a tree-like structure. We proposed a N-E algorithm to solve the dynamics of this robot, and benchmark the accuracy of our simulators. Yet, only the inertial, gravity and buoyancy forces are taken into account. We demonstrated that the robot can capsize in a narrow but relevant range of frequency swimming. This preliminary result will motivate a numerical and experimental parametric investigation of the stability of the robot. Besides, we don't know if the hydrodynamic effects such as wave damping or added mass, can prevent this instability. Consequently, an identification of the hydrodynamic coefficients is required to improve the simulator paving the way to controllers stabilizing the robot gaze.

VII. Acknowledgement

This work was supported by the French National Research Agency ANR (grant no. ANR-2019-CE33-0004- 01).

References

[1] G.-Z. Yang, *et al.*, The grand challenges of science robotics, Science robotics 3 (2018) p. eaar7650.

[2] D. Grémillet, *et al.*, Robots in ecology: welcome to the machine, Open J. Ecol. 2 (2012) 49–57.

[3] B. Bayat, *et al.*, Environmental monitoring using autonomous vehicles: a survey of recent searching techniques, Current opinion in biotechnology 45 (2017) 76–84.

[4] P. Liljebäck, R. Mills, Eelume: A flexible and subsea resident imr vehicle, in: Oceans 2017-Aberdeen, IEEE, 2017, pp. 1–4.

[5] M. Porez, F. Boyer, A. J. Ijspeert, Improved lighthill fish swimming model for bio-inspired robots, IJRR 33 (2014) 1322–1341.

[6] B. Bayat, A. Crespi, A. Ijspeert, Envirobot: A bio-inspired environmental monitoring platform, in: 2016 Ieee/Oes (Auv), 2016, pp. 381–386.

[7] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, J. T. Gravdahl, A review on modelling, implementation, and control of snake robots, Robotics and Autonomous systems 60 (2012) 29–40.

[8] J. Herault, *et al.*, Standing on the water: Stability mechanisms of snakes on free surface, in: Conf. on Biomimetic and Biohybrid Systems, 2020, pp. 165–175.

[9] X. Xie, *et al*, Quasi-static motion of a new serial snake-like robot on a water surface: a geometrical approach, in: 2021 IEEE/RSJ (IROS), 2021, pp. 7372–7377.

[10] J. Herault, A geometrically exact approach for floating slender bodies with finite deformations, Appl. Ocean Res. 101 (2020) 102220.

[11] W. Khalil, G. Gallot, F. Boyer, Dynamic modeling and simulation of a 3-d serial eel-like robot, IEEE Trans. Syst. Man Cybern. Syst. 37 (2007) 1259–1268.

[12] E. Kelasidi, *et al.*, Modeling of underwater snake robots, in: 2014 IEEE (ICRA), 2014, pp. 4540–4547.

[13] W. Khalil, E. Dombre, Modeling identification and control of robots, CRC Press, 2002.

[14] R. M. Murray, Z. Li, S. S. Sastry, A mathematical introduction to robotic manipulation, CRC press, 2017.

[15] T. I. Fossen, Handbook of marine craft hydrodynamics and motion control, John Wiley & Sons, 2011.

[16] J. Y. Luh, M. W. Walker, R. P. Paul, On-line computational scheme for mechanical manipulators (1980) 69–76.

[17] R. Featherstone, Rigid body dynamics algorithms, Springer, 2014.

[18] F. Boyer, M. Porez, Multibody system dynamics for bio-inspired locomotion: from geometric structures to computational aspects, Bioinspiration & biomimetics 10 (2015) p. 025007.

[19] G. Gallot, Modélisation Dynamique et Commande d'un robot Anguille, Ph.D. thesis, (ECN), 2007.

[20] B. C. Jayne, What defines different modes of snake locomotion?, Integrative and comparative biology 60 (2020) 156–170.

[21] J. Herault, *et al*, Instability of precession driven kelvin modes, Phys. Rev. Fluids 4 (2019) p.033901.

[22] E. Ott, Chaos in dynamical systems, Cambridge university press, 2002.