Dynamic Modeling and Simulation of a 3-D Serial Eel-Like Robot

Wisama Khalil, *Senior Member, IEEE*, Guillaume Gallot, and Frédéric Boyer

***Abstract*—This paper presents the dynamic modeling of a 3-D-serial underwater eel-like robot using recursive algorithms based on the Newton–Euler equations. Both direct and inverse models are treated in the paper. The inverse dynamic model algorithm gives the head acceleration and the joint torques as a function of the joint positions, velocities, and accelerations. The direct dynamic model gives the head and joint accelerations as a function of the joint positions, velocities, and input torques. The proposed algorithms can be considered as a generalization of the recursive Newton–Euler dynamic algorithms of serial manipulators with fixed base. The algorithms are easy to implement and to simulate whatever the number of degrees of freedom of the robot. An example with 12 spherical joints is presented. The fluid forces have been taken into account using a simple model based on Morison's model.**

***Index Terms*—Autonomous structures, dynamic modeling, eel-like robot, fluid forces, swimming robot.**

I. Introduction

RECENTLY, many projects are devoted to the design and control of anguilliform robots owing to their potential in specific underwater applications including oceanic inspection and industrial endoscopy [1]–[3]. The work presented in this paper is done in the framework of the project "Robot Anguille" supported by the French Centre National de la Recherche Scientifique (CNRS). In this project, the modeling approach is based on three hierarchical levels: 1) a "macro-continuous" modeling based on a beam-like approach [16]; 2) a serial rigid bodies structure with revolute joints and 3) on a Navier–Stokes code modeling the fluid–structure interactions [11]. This paper is devoted to the second level, which deals with a hyperredundant serial system. To develop the dynamic models for such a structure, we propose to use efficient recursive Newton–Euler algorithms similar to those proposed for rigid and flexible manipulators [4]–[9]. The main difference between the two systems is that the acceleration of the base in the case of manipulators is equal to zero, whereas in the case of a swimming structure, the head is free and its angular and translational accelerations must be determined in both direct and inverse dynamic models. The proposed dynamic models are easy to implement and simulate using numerical calculation. The inverse dynamic model, which is used in general in the control problems, can be used in simulation too when assuming that, the joint positions, velocities, and accelerations are given. The direct dynamic model can be used in simulation when the joint torques are specified.

This paper is organized as follows. In Section II, we present the kinematic modeling of the robot. Then, in Section III, we recall the general form of inverse and direct dynamic model of swimming robots using the Lagrange form and explain our choice to use recursive Newton–Euler algorithms. In Sections IV and V, we describe the different steps to solve the inverse, and then, the direct dynamic model of our robot. In Section VI, we describe the model of fluid-structure interaction, which is used to simulate the eel-robot. Finally in Section VII, we present some simulation results obtained using our algorithms.

II. Kinematic Modeling of the Structure

The system treated in this paper is an eel-like robot with serial structure. It is composed of a sequence of $n+1$ links and n joints. The joints can either be revolute or prismatic. Since prismatic joints are not used for such robots, we limit the description and the modeling in this paper for revolute joints only. A complex joint (universal or spherical) can be represented by an equivalent combination of two or three revolute joints, respectively. We use the notations of Khalil and Kleinfinger [8], [9] to describe the structure. The links are numbered such that link 0 constitutes the head of the robot and link n is the tail. We assign a frame Σ_j attached to each link j, such that the $\mathbf{z}_j$-axis is taken along the axis of joint j, and the $\mathbf{x}_j$-axis is along the common normal between $\mathbf{z}_j$ and $\mathbf{z}_{j+1}$.

The transformation matrix from frame Σ_{j-1} to frame Σ_j is expressed as a function of the following parameters [8], [9]:

1) α_j: The angle between $\mathbf{z}_{j-1}$ and $\mathbf{z}_j$ about $\mathbf{x}_{j-1}$.
2) $d_{j:}$ The distance between $\mathbf{z}_{j-1}$ and $\mathbf{z}_j$ along $\mathbf{x}_{j-1}$.
3) θ_j: The angle between $\mathbf{x}_{j-1}$ and $\mathbf{x}_j$ about $\mathbf{z}_j$.
4) r_j: The distance between $\mathbf{x}_{j-1}$ and $\mathbf{x}_j$ along $\mathbf{z}_j$.

The variable of joint j, defining the relative orientation between links j – 1 and j, is given by

$$\theta_j = q_j + q_{0j} \tag{1}$$

where q_{0j} is a constant offset.

Manuscript received February 21, 2006; revised June 14, 2006. This work was supported by the French Centre National de la Recherche Scientifique (CNRS) under Project "Robot Anguille" of the Interdisciplinary Research Programme ROBEA. This work appeared in part in IEEE International Conference on Robotics and Automation, 2005. This paper was recommended by Associate Editor P. J. Sanz.

The authors are with the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), Ecole Centrale de Nantes, Nantes Cedex 03 44321, France (e-mail: wisama.khalil@irccyn.ec-nantes.fr; guillaume.gallot@irccyn.ec-nantes.fr; frederic.boyer@emn.fr).

Digital Object Identifier 10.1109/TSMCC.2007.905831

The homogeneous transformation matrix, which define frame Σ_j relative to frame Σ_{j-1} is given by the (4×4) matrix [8], [9]

$$ {}^{j-1}\mathbf{T}_j = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

where C* and S* denote cos(*) and sin(*), respectively.

In the following, the upper left exponent indicates the projection frame. We note that the (3×3) orientation matrix ${}^{j-1}\mathbf{R}_j$ of frame Σ_j with respect to frame Σ_{j-1} is the (3×3) upper left submatrix of ${}^{j-1}\mathbf{T}_j$, whereas the position vector ${}^{j-1}\mathbf{P}_j$ defining the origin of frame Σ_j with respect to frame Σ_{j-1} is equal to the upper right (3×1) submatrix.

The transformation matrix between the world fixed frame Σ_w and the frame fixed with the head Σ_0 is denoted by ${}^{w}\mathbf{T}_0$. This matrix is supposed to be known at $t = 0$, it will be updated by integrating the head acceleration.

The Cartesian velocities and accelerations of the links are calculated using the following recursive equations [8], [9]:

$$ {}^{j}T_{j-1} = \begin{bmatrix} {}^{j}\mathbf{R}_{j-1} & -{}^{j}\mathbf{R}_{j-1}{}^{j-1}\hat{\mathbf{P}}_j \\ \mathbf{0}_{3\times 3} & {}^{j}\mathbf{R}_{j-1} \end{bmatrix} \tag{3} $$

$$ {}^{j}V_j = {}^{j}T_{j-1}{}^{j-1}V_{j-1} + \dot{\mathrm{q}}_j{}^{j}a_j \tag{4} $$

$$ {}^{j}\gamma_j = \begin{bmatrix} {}^{j}\mathbf{R}_{j-1}\left[{}^{j-1}\omega_{j-1} \times \left({}^{j-1}\omega_{j-1} \times {}^{j-1}\mathbf{P}_j\right)\right] \\ {}^{j}\omega_{j-1} \times \dot{\mathrm{q}}_j{}^{j}\mathbf{a}_j \end{bmatrix} + \ddot{q}_j{}^{j}a_j \tag{5} $$

$$ {}^{j}\dot{V}_j = {}^{j}T_{j-1}{}^{j-1}\dot{V}_{j-1} + {}^{j}\gamma_j \tag{6} $$

where

${}^{j}T_{j-1}$ (6×6) screw transformation matrix;
$\hat{\mathbf{P}}$ (3×3) skew matrix of vector product associated with the vector $\mathbf{P}$, such that $\mathbf{P} \times \mathbf{u} = \hat{\mathbf{P}} \cdot \mathbf{u}$
V_j (6×1) kinematic screw vector of link j, given by

$$ V_j = [\,\mathbf{V}_j^T \quad \omega_j^T\,]^T ; \tag{7} $$

$\mathbf{V}_j$ linear velocity of the origin of frame Σ_j;
ω_j angular velocity of link j;
${}^{j}\mathbf{a}_j$ (6×1) matrix defined by

$$ {}^{j}\mathbf{a}_j = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ {}^{j}\mathbf{h}_j \end{bmatrix} ; $$

$\mathbf{0}_{m\times n}$ $(m \times n)$ zero matrix;
$\mathbf{h}_j$ unit vector along $\mathbf{z}_j$-axis, representing the joint axis j. Thus, ${}^{j}\mathbf{h}_j = [0 \quad 0 \quad 1]^T$.

III. General Form of the Dynamic Models

A. Introduction

The dynamic model of an eel-like robot can be represented by the following relation:

$$ \begin{bmatrix} \mathbf{0}_{6\times 1} \\ \Gamma \end{bmatrix} = \mathbf{A} \begin{bmatrix} {}^{0}\dot{V}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} + \mathbf{C} \tag{8} $$

where

$\mathbf{\Gamma}$ $(n \times 1)$ vector of joint torques;
$\mathbf{q}$ $(n \times 1)$ vector of joint positions;
$\mathbf{A}$ $(6+n) \times (6+n)$ inertia matrix of the robot, whose elements are functions of the joint angles; it can be partitioned as follows:

$$ \mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix}. \tag{9} $$

$\mathbf{A}_{11}$ is the (6×6) equivalent inertia matrix of the composed link 0, which is composed of the inertia of all the links referred to frame Σ_0 (the head).

$\mathbf{A}_{22}$ is the $(n \times n)$ inertia matrix of the robot when the head is fixed.

$\mathbf{A}_{12}$ is the $(6 \times n)$ coupled inertia matrix of the joints and the head. It reflects the effect of the joint accelerations on the head motion, and the dual effect of head accelerations on the joint motions. $\mathbf{C}$ is the $(n+6) \times 1$ vector representing the Coriolis, centrifugal, gravity, and external forces effect on the robot. Its elements are functions of the head and joint velocities and the external forces, in particular, the hydraulic forces. This vector can be partitioned as follows:

$$ \mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \tag{10} $$

where

$\mathbf{C}_1$ Coriolis, centrifugal, gravity, and external forces effect on the head;
$\mathbf{C}_2$ Coriolis, centrifugal, gravity, and external forces effect on the links $1, \ldots, n$.

B. Solution of the Inverse Dynamic Model

In the case of an anguiliform robot, the inverse dynamic model gives the joint torques and the head acceleration in terms of the head velocity and joint positions, velocities, and acceleration. Thus, using (8), the inverse dynamic model is solved as follows. At first, we use the first row of (8) to obtain the head acceleration

$$ {}^{0}\dot{V}_0 = -(\mathbf{A}_{11})^{-1}\left(\mathbf{C}_1 + \mathbf{A}_{12}\ddot{\mathbf{q}}\right). \tag{11} $$

Then, the second row of (3) will be used to find the joint torques

$$ \Gamma = \mathbf{A}_{12}^T\,\dot{V}_0 + \mathbf{A}_{22}\,\ddot{\mathbf{q}} + \mathbf{C}_2. \tag{12} $$

C. Solution of the Direct Dynamic Model

In the case of an anguiliform robot, the direct dynamic model gives the joint accelerations and the head acceleration in terms

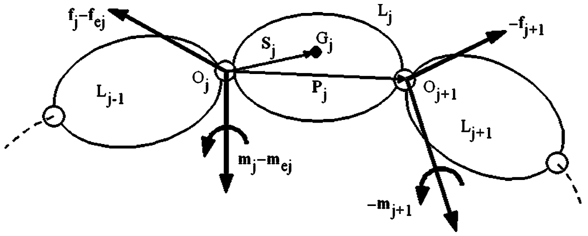

Fig. 1. Forces and moments on link j.

of the head velocity, and joint positions, velocities, and torques. Thus, using (8), the direct dynamic model is solved as follows:

$$\begin{bmatrix} \dot{V}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} -\mathbf{C}_1 \\ \Gamma - \mathbf{C}_2 \end{bmatrix}. \tag{13}$$

Solving the inverse and direct dynamic problems using the Lagrangian expanded form, as that given in (8), is very time consuming. This is because of the large number of degrees of freedom of the system; typically n is equal to 36 for our 12 spherical joint robots. This gives very complicated expressions for the elements of the matrices $\mathbf{A}$ and $\mathbf{C}$. Besides, inverting the inertia matrix $\mathbf{A}$ for the direct dynamic model will be tedious even for simulation. The researchers who tried such methods have limited their application for two-dimension (2-D) planar systems [1], [15], [19] with a small number of internal degrees of freedom. Therefore, we propose here to use a recursive method, which is easy to programme, and its computational complexity could be further improved using the techniques of customized symbolic method [7]–[9].

IV. Recursive Inverse Dynamic Mode

A. Introduction

In this section, we present the recursive Newton–Euler algorithm for computing the inverse dynamic model of an eel-like robot. With respect to manipulators, the main difference comes from the fact that the head is free, and its acceleration must be determined as an output of the algorithm.

The algorithm is based on the kinematic equations presented in Section II, giving the Cartesian velocity and acceleration of the links, and on the following Newton–Euler equation giving the total forces and moments on each link:

$$^jF_j = {}^jJ_j{}^j\dot{V}_j + \begin{bmatrix} ^j\omega_j \times \left(^j\omega_j \times {}^j\mathbf{MS}_j\right) \\ ^j\omega_j \times \left(^j\mathbf{J}_j{}^j\omega_j\right) \end{bmatrix}. \tag{14}$$

Hence, the equilibrium dynamic equation of each link is given by (Fig. 1)

$$^jf_j = {}^jF_j + {}^{j+1}T_j^T{}^{j+1}f_{j+1} + {}^jf_{ej} \tag{15}$$

where

F_j total external wrench on link j, given by

$$F_j = \begin{bmatrix} \mathbf{F}_j^T & \mathbf{M}_j^T \end{bmatrix}^T$$

$\mathbf{F}_j$ total external forces on link j;

$\mathbf{M}_j$ total moments of external forces on link j about the origin O_j;

J_j (6×6) inertia matrix of link j:

$$^jJ_j = \begin{bmatrix} M_j\mathbf{I}_3 & -{}^j\mathbf{M}\hat{\mathbf{S}}_j \\ ^j\mathbf{M}\hat{\mathbf{S}}_j & ^j\mathbf{J}_j \end{bmatrix} \tag{16}$$

$^j\mathbf{J}_j$ (3×3) inertia tensor of link j with respect to frame Σ_j;

$\mathbf{I}_3$ (3×3) identity matrix;

M_j mass of link j;

$\mathbf{MS}_j$ first moments of link j with respect to frame Σ_j;

f_j wrench exerted on link j by link $j-1$

$$f_j = \begin{bmatrix} \mathbf{f}_j^T & \mathbf{m}_j^T \end{bmatrix}^T$$

f_{ej} wrench exerted by link j on the environment, and representing in our case the fluid–structure interaction effect.

B. Recursive Calculation of the Inverse Dynamic Model Eel-Like Robot

The algorithm presented in the following can be considered as a generalization of that of Luh *et al.* [4] for rigid manipulators. The generalization concerns the calculation of the acceleration of the head. The inverse dynamic algorithm, in this case, consists of three recursive equations (a forward, then a backward, and then a forward), as for flexible manipulators, instead of two (forward and backward) for the rigid manipulators with fixed base.

1) Forward Recursive Calculation: In this step we calculate the screw transformation matrices, link velocities, and the elements of the accelerations and external wrenches on the links, which are independent of the acceleration of the robot head $(\dot{\mathbf{V}}_0, \dot{\omega}_0)$. Thus, we calculate for $j = 1, \ldots, n, {}^jT_{j-1}, {}^jV_j$ and $^j\gamma_j$ using (3), (4), and (5), respectively. We also calculate $^j\beta_j$ representing the elements of the Newton–Euler equations, which are independent of the head acceleration in (14) and (15) such that

$$^j\beta_j = {}^jf_{ej} + \begin{bmatrix} ^j\omega_j \times \left(^j\omega_j \times {}^j\mathbf{MS}_j\right) \\ ^j\omega_j \times \left(^j\mathbf{J}_j{}^j\omega_j\right) \end{bmatrix}. \tag{17}$$

2) Backward Recursive Equations: In this step, we obtain the head acceleration using the inertial parameters of the composite link 0, where the composite link j consists of the links $j, j+1, \ldots, n$.

We note that (15), giving the equilibrium equation of link j, can be rewritten using (14) and (17) as

$$^jf_j = {}^jJ_j{}^j\dot{V}_j + {}^j\beta_j + {}^{j+1}T_j^T{}^{j+1}f_{j+1}. \tag{18}$$

Applying the Newton–Euler equations on the composite link j, we obtain

$$^jf_j = {}^jJ_j{}^j\dot{V}_j + {}^j\beta_j + {}^{j+1}T_j^T\left(^{j+1}J_{j+1}{}^{j+1}\dot{V}_{j+1} + {}^{j+1}\beta_{j+1}\right) + \cdots + {}^nT_j^T\left(^nJ_n{}^n\dot{V}_n + {}^n\beta_n\right). \tag{19}$$

Substituting ${}^{j+1}\dot{V}_{j+1}, \ldots, {}^{n-1}\dot{V}_{n-1}, {}^{n}\dot{V}_{n}$ in terms of ${}^{j}\dot{V}_{j}$ using (6), we obtain

$$\begin{aligned} {}^{j+1}\dot{V}_{j+1} &= {}^{j+1}T_{j}{}^{j}\dot{V}_{j} + {}^{j+1}\gamma_{j+1}, \\ {}^{j+2}\dot{V}_{j+2} &= {}^{j+2}T_{j+1}{}^{j+1}\dot{V}_{j+1} + {}^{j+2}\gamma_{j+2} \\ &= {}^{j+2}T_{j+1}\left({}^{j+1}T_{j}{}^{j}\dot{V}_{j} + {}^{j+1}\gamma_{j+1}\right) + {}^{j+2}\gamma_{j+2} \quad (20) \end{aligned}$$

$$ {}^{n}\dot{V}_{n} = {}^{n}T_{j}{}^{j}\dot{V}_{j} + {}^{n}T_{j+1}{}^{j+1}\gamma_{j+1} + \cdots + {}^{n}\gamma_{n}. \quad (21)$$

From (19)–(21), we obtain

$$ {}^{j}f_{j} = {}^{j}J_{j}^{c\,j}\dot{V}_{j} + {}^{j}\beta_{j}^{c} \quad (22)$$

with

$$ {}^{j}J_{j}^{c} = {}^{j}J_{j} + {}^{j+1}T_{j}^{T\,j+1}J_{j+1}^{c\,j+1}T_{j} \quad (23)$$

$$ {}^{j}\beta_{j}^{c} = {}^{j}\beta_{j} + {}^{j+1}T_{j}^{T\,j+1}\beta_{j+1}^{c} + {}^{j+1}T_{j}^{T\,j+1}J_{j+1}^{c\,j+1}\gamma_{j+1} \quad (24)$$

where ${}^{j}J_{j}^{c}$ is the spatial inertial matrix of the composite link j. For $j = 0$, and since ${}^{0}f_{0}$ is equal to zero, we obtain using (22)

$$ {}^{0}\dot{V}_{0} = -\left({}^{0}J_{0}^{c}\right)^{-1}{}^{0}\beta_{0}^{c}. \quad (25)$$

To summarize, the recursive equations of this step consist of initialising ${}^{n}J_{n}^{c} = {}^{n}J_{n}, {}^{n}\beta_{n}^{c} = {}^{n}\beta_{n}$ and then calculating (23) and (24) for $j = n-1, \ldots, 0$.

At the end, ${}^{0}\dot{V}_{0}$ is calculated using (25).

Comparing (25) with (11), we can deduce that $\mathbf{A}_{11}$ is equal to ${}^{0}J_{0}^{c}$, whereas ${}^{0}\beta_{0}^{c}$ is equal to $(\mathbf{C}_{1} + \mathbf{A}_{12}\ddot{\mathbf{q}})$.

3) Forward Recursive Equations: After calculating ${}^{0}\dot{V}_{0}$, the wrench ${}^{j}f_{j}$ and the joint torques are then obtained using (6) and (22) for $j = 1, \ldots, n$ as

$$ {}^{j}\dot{V}_{j} = {}^{j}T_{j-1}{}^{j-1}\dot{V}_{j-1} + {}^{j}\gamma_{j} \quad (26)$$

$$ {}^{j}f_{j} = \begin{bmatrix} {}^{j}\mathbf{f}_{j} \\ {}^{j}\mathbf{m}_{j} \end{bmatrix} = {}^{j}J_{j}^{c\,j}\dot{V}_{j} + {}^{j}\beta_{j}^{c}. \quad (27)$$

The joint torque is calculated by projecting ${}^{j}f_{j}$ on the joint axis, and by taking into account the friction and the actuators inertia

$$\Gamma_{j} = {}^{j}f_{j}^{T\,j}a_{j} + F_{sj}\ \mathrm{sign}\left(\dot{q}_{j}\right) + F_{vj}\ \dot{q}_{j} + I_{aj}\ \ddot{q}_{j} \quad (28)$$

where

F_{sj} Coulomb friction parameter of joint j;
F_{vj} viscous friction parameter of joint j;
I_{aj} moment of inertia of the rotor of actuator j.

It is to be noted that the inverse dynamic model algorithm can be used in the dynamic simulation of the eel-like robot when assuming that the joint positions, velocities, and accelerations trajectories are given. At each sampling time, the acceleration of the head will be integrated to provide the angular and linear velocities for the next sampling time. The integration of the linear velocity will provide ${}^{w}\mathbf{P}_{0}$ representing the position of the origin of frame Σ_{0}. Concerning the orientation, the angular velocity is first transformed into quaternion velocity using the following equations [8]:

$$\dot{\mathbf{Q}} = \mathbf{\Omega}_{q}\mathbf{Q} \quad (29)$$

where

$$\mathbf{\Omega}_{q} = \frac{1}{2\,\|\mathbf{Q}\|}\begin{bmatrix} 0 & -\omega_{1} & -\omega_{2} & -\omega_{3} \\ \omega_{1} & 0 & -\omega_{3} & \omega_{2} \\ \omega_{2} & \omega_{3} & 0 & -\omega_{1} \\ \omega_{3} & -\omega_{2} & \omega_{1} & 0 \end{bmatrix}$$

$$\omega_{0} = \begin{bmatrix}\omega_{1} & \omega_{2} & \omega_{3}\end{bmatrix}^{T}$$

$$\mathbf{Q} = \begin{bmatrix}Q_{1} & Q_{2} & Q_{3} & Q_{4}\end{bmatrix}^{T}. \quad (30)$$

The integration of (29) gives the quaternion $\mathbf{Q}$. Then, the orientation matrix ${}^{w}\mathbf{R}_{0}$ is calculated using the relation

$$ {}^{w}\mathbf{R}_{0} = \begin{bmatrix} 2\left(Q_{1}^{2} + Q_{2}^{2}\right) - 1 & 2(Q_{2}Q_{3} - Q_{1}Q_{4}) & 2(Q_{2}Q_{4} + Q_{1}Q_{3}) \\ 2(Q_{2}Q_{3} + Q_{1}Q_{4}) & 2\left(Q_{1}^{2} + Q_{3}^{2}\right) - 1 & 2(Q_{3}Q_{4} - Q_{1}Q_{2}) \\ 2(Q_{2}Q_{4} - Q_{1}Q_{3}) & 2(Q_{3}Q_{4} + Q_{1}Q_{2}) & 2\left(Q_{1}^{2} + Q_{4}^{2}\right) - 1 \end{bmatrix} \quad (31)$$

where, for an orientation $\mathbf{rot}(\mathbf{u}, \alpha)$, denoting a rotation around the unit vector $\mathbf{u}$ by an angle α, the quaternion components are defined as

$$Q_{1} = \cos(\alpha/2) \qquad Q_{2} = u_{1}\ \sin(\alpha/2)$$
$$Q_{3} = u_{2}\ \sin(\alpha/2) \qquad Q_{4} = u_{3}\ \sin(\alpha/2)$$

where $\mathbf{u} = [u_{1}\ u_{2}\ u_{3}]^{T}$.

V. Direct Dynamic Model

A. Introduction

The direct dynamic model of an anguiliform robot must provide the head and joint accelerations in terms of the joint positions, velocities, and torques. The proposed algorithm can be considered as the generalization of that of Featherstone [5] which has been proposed for rigid manipulators with fixed base.

B. Calculation of the Direct Dynamic Model

The direct dynamic model consists of three recursive calculations in the same order as those of the inverse dynamic model (forward, backward, and forward).

1) Forward Recursive Equations: We calculate the link Cartesian velocities using (3) and (4), and the terms of Cartesian accelerations and equilibrium equations of the links that are independent of the acceleration of the head and of the joint accelerations. From (5), (14), and (15), we calculate the following recursive equations for $j = 1, \ldots, n$:

$$ {}^{j}\zeta_{j} = \begin{bmatrix} {}^{j}\mathbf{R}_{j-1}\left[{}^{j-1}\omega_{j-1} \times \left({}^{j-1}\omega_{j-1} \times {}^{j-1}\mathbf{P}_{j}\right)\right] \\ {}^{j}\omega_{j-1} \times \dot{q}_{j}{}^{j}\mathbf{h}_{j} \end{bmatrix} \quad (32)$$

$$ {}^{j}\beta_{j} = {}^{j}f_{ej} + \begin{bmatrix} {}^{j}\omega_{j} \times \left({}^{j}\omega_{j} \times {}^{j}\mathbf{MS}_{j}\right) \\ {}^{j}\omega_{j} \times \left({}^{j}\mathbf{J}_{j}{}^{j}\omega_{j}\right) \end{bmatrix}. \quad (33)$$

2) Backward Recursive Equations: In this second step, we first initialize ${}^{n}J_{n}^{*} = {}^{n}J_{n}, {}^{n}\beta_{n}^{*} = {}^{n}\beta_{n}$, and then, we calculate for $j = n, \ldots, 1$ the following elements, which permit to calculate ${}^{j}f_{j}$ and $\ddot{q}_{j}$ in terms of ${}^{j-1}\dot{V}_{j-1}$ and will be used in the third recursive equations [see (A17) and (A18) in Appendix A]:

$$H_{j} = {}^{j}a_{j}^{T}\,{}^{j}J_{j}^{*}\,{}^{j}a_{j} + Ia_{j} \tag{34}$$

$${}^{j}K_{j} = {}^{j}J_{j}^{*} - {}^{j}J_{j}^{*}\,{}^{j}a_{j}H_{j}^{-1}\,{}^{j}a_{j}^{T}\,{}^{j}J_{j}^{*} \tag{35}$$

$${}^{j-1}J_{j-1}^{*} = {}^{j-1}J_{j-1} + {}^{j}T_{j-1}^{T}\,{}^{j}K_{j}\,{}^{j}T_{j-1} \tag{36}$$

$$\tau_{j} = \Gamma_{j} - F_{sj}\,\mathrm{sign}(\dot{\mathrm{q}}_{j}) - F_{vj}\dot{q}_{j} \tag{37}$$

$${}^{j}\alpha_{j} = {}^{j}K_{j}\,{}^{j}\zeta_{j} + {}^{j}J_{j}^{*}\,{}^{j}a_{j}H_{j}^{-1}\left(\tau_{j} - {}^{j}a_{j}^{T}\,{}^{j}\beta_{j}^{*}\right) + {}^{j}\beta_{j}^{*} \tag{38}$$

$${}^{j-1}\beta_{j-1}^{*} = {}^{j-1}\beta_{j-1} + {}^{j}T_{j-1}^{T}\,{}^{j}\alpha_{j}. \tag{39}$$

3) Forward Recursive Equations: At first, the head acceleration is calculated by the following relation (see Appendix A):

$${}^{0}\dot{V}_{0} = -\left({}^{0}J_{0}^{*}\right)^{-1}\,{}^{0}\beta_{0}^{*}. \tag{40}$$

We note that ${}^{0}\beta_{0}^{*}$ is a function of τ, whereas ${}^{0}\beta_{0}^{c}$ in (25) is a function of $\ddot{q}$.

Then, $\ddot{q}_{j}$ and ${}^{j}f_{j}$ (if desired) are calculated for $j = 1, \ldots, n$ using the following equations (see Appendix A):

$$\ddot{q}_{j} = H_{j}^{-1}\left[-{}^{j}a_{j}^{T}\,{}^{j}J_{j}^{*}\left({}^{j}\dot{V}_{j-1} + {}^{j}\zeta_{j}\right) + \tau_{j} - {}^{j}a_{j}^{T}\,{}^{j}\beta_{j}^{*}\right] \tag{41}$$

$${}^{j}f_{j} = {}^{j}K_{j}\,{}^{j}T_{j-1}\,{}^{j-1}\dot{V}_{j-1} + {}^{j}\alpha_{j} \tag{42}$$

where

$${}^{j}\dot{V}_{j} = {}^{j}\dot{V}_{j-1} + {}^{j}a_{j}\ddot{q}_{j} + {}^{j}\zeta_{j}. \tag{43}$$

VI. Fluid–Structure Interaction Model

To simulate a swimming eel-like robot, we adopt a simple fluid mechanical model. We assume that the forces exerted by the fluid onto a given link are due only to the motion of that link. Moreover the links are assimilated to elliptic cross-sectional cylinders whose serial assembly builds a shape variable cylinder with one of its three characteristic dimensions (the axial one) longer (with a factor larger than 10) than the two others (the transverse ones). Hence, we can invoke the slender–body theory of fluid mechanics [20]. Based on this assumption, the 3-D fluid flow near the body can be replaced by a continuous slicing of planar flows transverse to the cylinders' axes. Hence, the fluid forces can be modeled by a density of wrenches applied onto each cross section of the links, which only depends of the transverse links' motion. This corresponds to the so-called strip-theory approach, commonly used in naval engineering [21]. It is worth noting that such a model neglects the influence onto the fluid flow due to the corners induced by the presence of the joints.

To simplify the writing, we assume that the mass per unit of volume of the robot is equal to that of water such that the robot is neutrally buoyant. Moreover, ${}^{j}\mathbf{V}_{j}(s)$ denotes the velocity of a cross section of the link j positioned at the distance s along the link axis from the point O_{j} (see Fig. 2). This velocity can be decomposed in the local frame $(\mathbf{e}_{j1}, \mathbf{e}_{j2}, \mathbf{e}_{j3})$ as

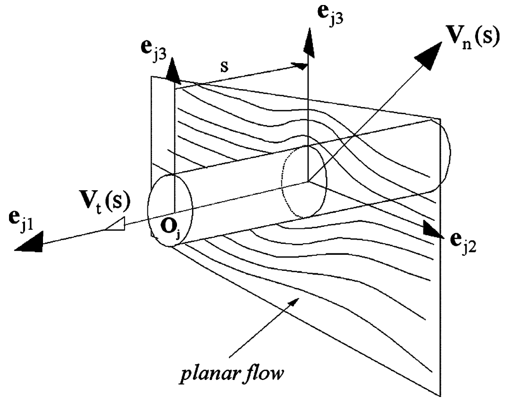

Fig. 2. Representation of fluid wrench in the cross-section frame.

$${}^{j}\mathbf{V}_{j}(s) = V_{tj}(s)\mathbf{e}_{j1} + V_{nj2}(s)\mathbf{e}_{j2} + V_{nj3}(s)\mathbf{e}_{j3} \tag{44}$$

where

V_{tj} the forward velocity component along axis $\mathbf{e}_{j1}$;
V_{nj2}, V_{nj3} the perpendicular velocities along axis $\mathbf{e}_{j2}$ and $\mathbf{e}_{j3}$.

We also define $\mathbf{V}_{nj} = V_{nj2}\mathbf{e}_{j2} + V_{nj3}\mathbf{e}_{j3}$, and $\|\mathbf{V}_{nj}\| = \sqrt{V_{nj2}^{2} + V_{nj3}^{2}}$. Similar relations to (44) can be written for ${}^{j}\dot{\mathbf{V}}_{j}, {}^{j}\omega_{j}$, and ${}^{j}\dot{\omega}_{j}$.

With the previous assumptions, the model of the contact forces between the fluid and the cylindrical links of the eel is that of Morison [10], and can be defined by the following field (along the links' axes) of hydraulic wrenches densities per unit of link axial length (with respect to the s cross-sectional center):

$${}^{j}f_{hj}(s) = \begin{bmatrix} {}^{j}\mathbf{f}_{hj}(s) \\ {}^{j}\mathbf{m}_{hj}(s) \end{bmatrix} = \begin{bmatrix} {}^{j}\mathbf{f}_{\text{drag}}(s) \\ {}^{j}\mathbf{m}_{\text{drag}}(s) \end{bmatrix} + \begin{bmatrix} {}^{j}\mathbf{f}_{\text{am}}(s) \\ {}^{j}\mathbf{m}_{\text{am}}(s) \end{bmatrix} \tag{45}$$

where, forces and moments are given respectively as

$$\begin{aligned} {}^{j}\mathbf{f}_{\text{drag}}(s) &= C_{ld1,j}\,|V_{tj}(s)|\,V_{tj}(s)\mathbf{e}_{j1} \\ &\quad + \sum_{i=2}^{3} C_{ldi,j}\,\|\mathbf{V}_{nj}(s)\|V_{nji}(s)\mathbf{e}_{ji} \end{aligned} \tag{46}$$

$${}^{j}\mathbf{m}_{\text{drag}}(s) = C_{ad1,j}\,|\omega_{tj}|\,\omega_{tj}\mathbf{e}_{j1}. \tag{47}$$

And the density per unit of link axial length of added mass forces and torques is given by

$${}^{j}\mathbf{f}_{\text{am}}(s) = \sum_{i=2}^{2} C_{lmi,j}\dot{V}_{nji}(s)\mathbf{e}_{ji} \tag{48}$$

$${}^{j}\mathbf{m}_{\text{am}}(s) = C_{am1,j}\dot{\omega}_{tj}\mathbf{e}_{j1} \tag{49}$$

where $\mathbf{f}_{\text{drag}}$ and $\mathbf{m}_{\text{drag}j}$ are due to the friction viscosity and pressure difference, whereas $\mathbf{f}_{\text{am}}$ and $\mathbf{m}_{\text{am}j}$ are in relation with the quantity of fluid accelerated by the movement. Finally the coefficients $C_{ldi,j}, C_{lmi,j}$, $C_{ad1,j}$ and C_{am1j} depend on the mass per unit of volume of the fluid, the shape and size of the profile (here elliptic), and the Reynolds number of the moving profile in the fluid (approximately 10^5). Their expressions are given in Section VII.

The translational velocity and acceleration are calculated by

$$ {}^j\mathbf{V}_j(s) = {}^j\mathbf{V}_j + {}^j\omega_j \times {}^j\mathbf{P}_j(s) \tag{50}$$

$$ {}^j\dot{\mathbf{V}}_j(s) = {}^j\dot{\mathbf{V}}_j + {}^j\dot{\omega}_j \times {}^j\mathbf{P}_j(s) + {}^j\omega_j \times {}^j\omega_j \times {}^j\mathbf{P}_j(s) \tag{51}$$

where ${}^j\mathbf{P}_j(s)$ is the position of the cross section of distance s from the origin of the link fixed frame along the eel's backbone.

Then, by superimposing all the "slice-by-slice" contributions from $s = 0$ to $s = L_j$ (the axial length of the jth link), we find the global wrench exerted by link j on the fluid, expressed at O_j as

$$ {}^j f_{hj} = \begin{bmatrix} {}^j\mathbf{f}_{hj} \\ {}^j\mathbf{m}_{hj} \end{bmatrix} = \begin{bmatrix} {}^j\mathbf{f}_{\text{drag}j} \\ {}^j\mathbf{m}_{\text{drag}j} \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{\text{am}j} \\ {}^j\mathbf{m}_{\text{am}j} \end{bmatrix} = \int_0^{L_j} {}^j f_{hj}(s)\, ds \tag{52}$$

where

$$ {}^j f_{hj}(s) = \begin{bmatrix} {}^j\mathbf{f}_{hj}(s) \\ {}^j\mathbf{m}_{hj}(s) \end{bmatrix} = {}^sT_j^T \left(\begin{bmatrix} {}^j\mathbf{f}_{\text{drag}}(s) \\ {}^j\mathbf{m}_{\text{drag}}(s) \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{\text{am}}(s) \\ {}^j\mathbf{m}_{\text{am}}(s) \end{bmatrix} \right) \tag{53}$$

and

$$ {}^sT_j = \begin{bmatrix} \mathbf{I}_3 & -{}^j\hat{\mathbf{P}}(s) \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix}.$$

The first term (drag and viscous wrench) of (52) is integrated numerically at each sample time of the algorithm from $s = 0$ to $s = L_j$, while the second contribution (added mass) can be explicitly computed in the local frame $(\mathbf{O}_j, \mathbf{e}_{j1}, \mathbf{e}_{j2}, \mathbf{e}_{j3})$ as

$$ {}^j f_{\text{am}j} = \begin{bmatrix} {}^j\mathbf{f}_{\text{am}j} \\ {}^j\mathbf{m}_{\text{am}j} \end{bmatrix} = {}^jJ_j^{aj}\, {}^j\dot{V}_j + {}^j\beta_{\text{am}j} \tag{54}$$

where ${}^jJ_j^{aj}$ is the (6×6) added inertia matrix and ${}^j\beta_{\text{am}j}$ the (6×1) matrix of Coriolis-centrifugal forces, both produced by the added fluid masses. In order to compute these two matrices, we first rewrite (54) in the local frame $(\mathbf{G}_j, \mathbf{e}_{j1}, \mathbf{e}_{j2}, \mathbf{e}_{j3})$ at the center of mass of the link $\mathbf{G}_j$ as

$$ {}^j f_{\text{am}Gj} = \begin{bmatrix} {}^j\mathbf{f}_{\text{am}Gj} \\ {}^j\mathbf{m}_{\text{am}Gj} \end{bmatrix} = {}^jJ_{Gj}^{aj}\, {}^j\dot{V}_{Gj} + {}^j\beta_{\text{am}Gj} \tag{55}$$

where we introduced the notations

$$ {}^jJ_{Gj}^{aj} = \begin{bmatrix} \mathbf{m}_{aGj} & -\mathbf{m}\hat{\mathbf{s}}_{aGj} \\ \mathbf{m}\hat{\mathbf{s}}_{aGj} & \mathbf{I}_{aGj} \end{bmatrix} \tag{56}$$

$$ {}^j\beta_{\text{am}Gj} = \begin{bmatrix} 0_{3\times 1} \\ \frac{1}{3} {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} \left({}^j\omega_j \times {}^j\omega_j \times {}^j\mathbf{S}_j \right) \end{bmatrix} \tag{57}$$

where ${}^j\mathbf{S}_j$ is the position of the center of mass of link j with respect to the origin of the link fixed frame. Furthermore, with the expressions of the local (slice-by-slice) added mass coefficients (48) and (49), ${}^jJ_{Gj}^{aj}$ can be detailed as follows:

1) the (3×3) matrix of added linear inertia

$$\text{a)}\quad {}^j\mathbf{m}_{aGj} = \text{diag}_{i=1,2,3}(m_{aij})$$

where $m_{a1j} = 0, m_{a2j} = C_{lm2,j}.L_j, m_{a3j} = C_{lm3,j}L_j$;

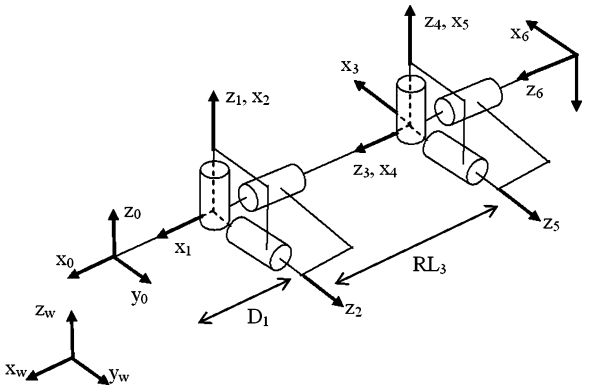

Fig. 3. Three-dimensional eel-like robot.

2) the (3×3) matrix of added linear-angular coupled inertia

$$\text{b)}\quad {}^j\mathbf{m}\hat{\mathbf{s}}_{aGj} = \mathbf{0}_{3\times 3}$$

3) the (3×3) matrix of added angular inertia

$$\text{c)}\quad {}^j\mathbf{I}_{aGj} = \text{diag}_{i=1,2,3}(\mathbf{I}_{aij})$$

where $I_{a1j} = C_{am1,j}.L_j, I_{a2j} = C_{lm3,j}.L_j^3/12, I_{a3j} = C_{lm2,j}.L_j^3/12$.

At this point, let us remark that all these data correspond to the (6×6) added mass matrix of a thin ellipsoid computed at its mass center [12].

Thus, (55) can be detailed as

$$ \begin{bmatrix} {}^j\mathbf{f}_{\text{am}Gj} \\ {}^j\mathbf{m}_{\text{am}Gj} \end{bmatrix} = \begin{bmatrix} {}^j\mathbf{m}_{aGj} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & {}^j\mathbf{I}_{aGj} \end{bmatrix} \begin{bmatrix} {}^j\dot{\mathbf{V}}_{Gj} \\ {}^j\dot{\omega}_j \end{bmatrix} + {}^j\beta_{\text{am}Gj}. \tag{58}$$

Transforming this wrench to the origin of frame j gives the detailed expression of (52)

$$ {}^j f_{hj} = \begin{bmatrix} {}^j\mathbf{m}_{aGj} & -{}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \\ {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} & {}^j\mathbf{I}_{aGj} - {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \end{bmatrix} {}^j\dot{V}_j + \begin{bmatrix} {}^j\mathbf{m}_{aGj} \left({}^j\hat{\boldsymbol{\omega}}_j {}^j\hat{\boldsymbol{\omega}}_j {}^j\mathbf{S}_j \right) \\ \left({}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} + \frac{1}{3} {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} \right) \left({}^j\hat{\boldsymbol{\omega}}_j {}^j\hat{\boldsymbol{\omega}}_j {}^j\mathbf{S}_j \right) \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{\text{drag}j} \\ {}^j\mathbf{m}_{\text{drag}j} \end{bmatrix}. \tag{59}$$

The introduction of (59) into (17) yields the elements corresponding to ${}^j f_{ej}$ due to the fluid contact forces and a constant term to be added to the (6×6) inertia matrix of each link called "added mass" due to fluid. We finally obtain

$$ {}^jJ_j = {}^jJ_j + J_j^{aj} \tag{60}$$

with

$$ {}^jJ_j^{aj} = \begin{bmatrix} {}^j\mathbf{m}_{aGj} & -{}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \\ {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} & {}^jI_{aGj} - {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \end{bmatrix} \tag{61}$$

$$ {}^j f_{ej} = \begin{bmatrix} {}^j\mathbf{f}_{\text{drag}j} \\ {}^j\mathbf{m}_{\text{drag}j} \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{m}_{aGj} \left({}^j\hat{\omega}_j {}^j\hat{\omega}_j {}^j\mathbf{S}_j \right) \\ \frac{4}{3} {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} \left({}^j\hat{\omega}_j {}^j\hat{\omega}_j {}^j\mathbf{S}_j \right) \end{bmatrix}. \tag{62}$$

TABLE I
GEOMETRIC PARAMETERS OF THE STRUCTURE

j	α_j	d_j	θ_j	r_j	q_{0j}
1	0	0.4	θ_1	0	0
2	$-\pi/2$	0	θ_2	0	$\pi/2$
3	$-\pi/2$	0	θ_3	0.14	$\pi/2$
4	$\pi/2$	0	θ_4	0	$-\pi/2$
…					
35	$-\pi/2$	0	θ_{35}	0	$-\pi/2$
36	$-\pi/2$	0	θ_{36}	0.14	$\pi/2$

VII. Simulation Examples

In this section, we present some simulation results obtained for an eel-like robot using Matlab and Simulink. The robot is composed of 13 rigid bodies connected by 12 spherical joints. Each joint is represented by three intersecting revolute joints (Fig. 3). This gives a system with 36 revolute joints and 37 links. The first link (link 0), which is the head of the robot, is composed of a half of a spheroid and an elliptic cylinder. The links 3 $j-2$ and 3 $j-1$, for $j=1,\ldots,12$, are virtual zero length and zero mass links; thus, there are no fluid forces acting on them. The links $3j$, for $j=1,\ldots,12$ are elliptic cylinders and the last link, the tail, is ended by a half of an ellipsoid.

The geometric parameters of the robot are given in Table I. The offset values q_{0j} are defined such that with $q_j = 0$, the x_0-axis is aligned with the z_{3j}-axis of the elliptical cylinder bodies (with j being the number of the spherical joint, $j=1,\ldots,12$).

From Fig. 3, we deduce that the local frames ($\mathbf{e}_{j1}$, $\mathbf{e}_{j2}$, $\mathbf{e}_{j3}$) taken to express the fluid–structure model for the link 0 and $3j (j=1,\ldots,12)$ are as follows.

1) For the head, $\mathbf{e}_{01}=\mathbf{x}_0 \quad \mathbf{e}_{02}=\mathbf{y}_0 \quad \mathbf{e}_{03}=\mathbf{z}_0$.
2) For the links $3j$, $\mathbf{e}_{3j1}=\mathbf{z}_{3j} \quad \mathbf{e}_{3j2}=\mathbf{x}_{3j} \quad \mathbf{e}_{3j3}=\mathbf{y}_{3j}$.

The total length of the robot is of 2.08 m. The cross section is of elliptic shape (whose great axis length is equal to 18 cm and its small axis length is equal to 13 cm). Denoting that the half small and great axis's lengths of the elliptic section are a and b, the coefficients of the fluid–structure model (46)–(47) are

$$C_{\mathrm{ld1}}=(1/2)\rho C_1\pi(a+b)/2 \quad C_{\mathrm{ld2}}=(1/2)\rho C_2 2b$$
$$C_{\mathrm{ld3}}=(1/2)\rho C_3 2a \quad C_{\mathrm{ad1}}=(1/2)\rho C_4(b^2-a^2)^2$$
$$C_{\mathrm{lm2}}=\rho\pi b^2 C_5 \quad C_{\mathrm{lm3}}=\rho\pi a^2 C_6 \quad C_{\mathrm{am1}}=\rho\pi C_7(b^2-a^2)^2/8$$

where ρ is the robot volume density taken equal to 1.

From [12], $C_1=0.01, C_2=C_3=C_4=1$ and $C_5=C_6=C_7=1$ which correspond to the values of a cylindrical obstacle plunged in a flow with a Reynolds number of approximately $R_e \simeq 10^5$, meaning an eel velocity of approximately 1 m $\cdot$ s^{-1}.

The simulation is carried out firstly using the inverse dynamic model by giving $\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t)$ as inputs and ${}^0V_0(0)=0$ as initial condition. Then, to validate the direct dynamic model, the simulation has been carried out using the inputs $\mathbf{q}(t), \dot{\mathbf{q}}(t)$, and $\Gamma(t)$ obtained from the first simulation using the inverse dynamic model, and the initial condition ${}^0V_0(0)=0$. The two simulations give the same results, which are given in the following sections.

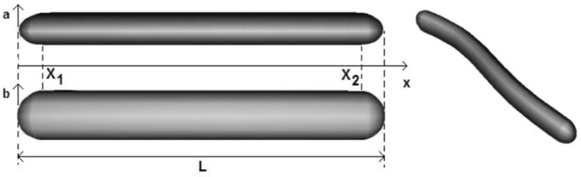

Fig. 4. Geometry of the eel.

A. First Example: A Planar Propulsion

In this example, we study the planar forward propulsion. Following biomechanic's literature about anguilliform locomotion [13]–[15], we start with a planar forward propulsion motion. It is produced by a continuous motion law of the following form:

$$Q(s,t)=f(t).A.e^{\alpha.s}\,\sin\left[2\pi\left(\frac{s}{\lambda}-\frac{t}{T}\right)\right] \tag{63}$$

where A is the amplitude of the motion, α is introduced to increase the amplitude when going from the head to the tail [14], $1/T$ is the frequency of the wave, λ represents the length of the wave, and s the curvilinear coordinate along eel's backbone (Fig. 4). The function $f(t)$ is a smoothing function, which is taken equal to

$$f(t)=f_1(t)=a_0.t^5+a_1.t^4+a_2.t^3+a_3.t^2+a_4.t+a_5 \; t\le t_f$$
$$f(t)=1 \qquad t>t_f \tag{64}$$

where the coefficients satisfy the following conditions:

$$f_1(0)=0 \quad f_1(t_f)=1 \quad \dot{f}_1(0)=\ddot{f}_1(0)=0$$
$$\dot{f}_1(t_f)=\ddot{f}_1(t_f)=0$$

and t_f is the ending time of $f_1(t)$.

To apply this continuous motion law to our model, we have to discretize (63) to have the corresponding value on each joint [18]. The values of the first joints of the spherical joints (joints numbered $j=3i-2$, with $i=1,\ldots,12$ denotes the number of the spherical joint) are

$$q_j(t)=Q(X_{j+1},t)-Q(X_j,t)+O_{\mathrm{ff}_1} \tag{65}$$

where $X_j=\sum_{k=1}^{j} r_k$ is the distance of each joint from the head, and O_{ff_1} is a constant value used to carry out the deviation of the eel.

The other joint angles are taken equal to the constant offset given in Table I. The following numerical values are used:

$$A=0.06 \quad \alpha=1.2 \quad t_f=2\,\mathrm{s} \quad \lambda=1.8\,\mathrm{m}$$
$$T=2\,\mathrm{s}\,O_{\mathrm{ff}_1}=0.$$

Fig. 5 shows the head trajectory of the eel in the $\mathbf{x}_w$–$\mathbf{y}_w$ plane of the fixed world frame for a simulation of 30 s.

Fig. 6 gives the linear forward velocity of the robot in the fixed-world frame.

These results show that the eel has a straight line trajectory with a small transversal oscillatory motion. With the proposed motion law, the eel needs approximately 30 s to reach its final

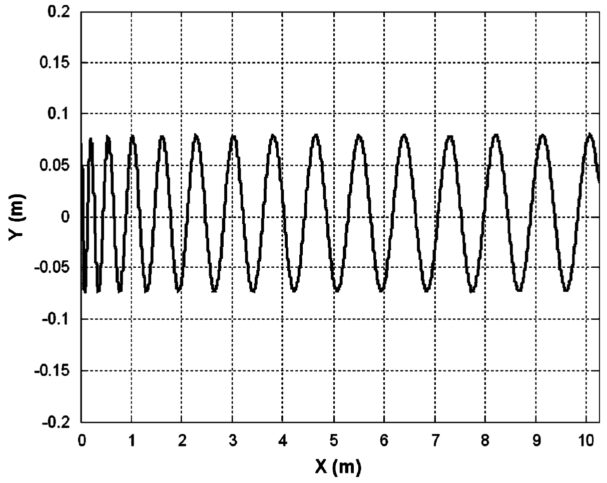

Fig. 5. Head's trajectory in the $\mathbf{x}$–$\mathbf{y}$ plane of the fixed-world frame.

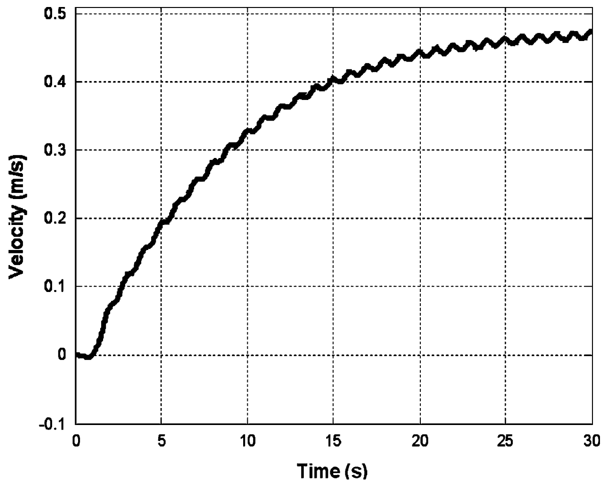

Fig. 6. Velocity of the head with respect to time.

speed, in order to satisfy the following limits of our actuators for $j = 1, \ldots, 12$:

$$\Gamma_{\max} = \pm 4N \cdot m \text{ for joints } 3j{-}2 \text{ and } 3j{-}1$$

$$\Gamma_{\max} = \pm 0.7N \cdot m \text{ for joints } 3j.$$

B. Second Example: A Three-Dimensional Looping

The goal of this example is to make the eel follow a vertical circular trajectory in the $\mathbf{x}_w$–$\mathbf{z}_w$ plane of the fixed frame. This is achieved by adding to the propulsive law of the first example, the following evolution on the second variables of the spherical joints $(j = 3i - 1)$ for $i = 1, \ldots, 12$:

$$q_j = f(t) O_{\mathrm{ff}_2} \tag{66}$$

where $f(t)$ is the same function defined in the planar example. The last joints ($j = 3i$ for $i = 1, \ldots, 12$) of the spherical joints are taken equal to the constant offset given in Table I. The values of the propulsive law A,α, t_f, λ, T, and O_{ff_1} are the same as in the first example, whereas $O_{\mathrm{ff}_2} = \pi/40$.

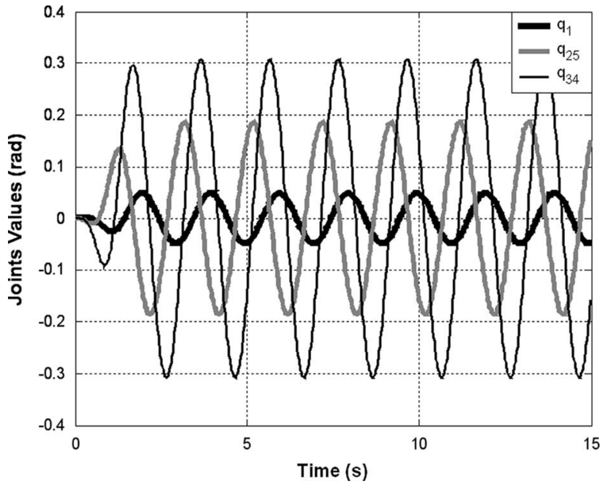

Fig. 7. Trajectories of joints 1, 25, and 34.

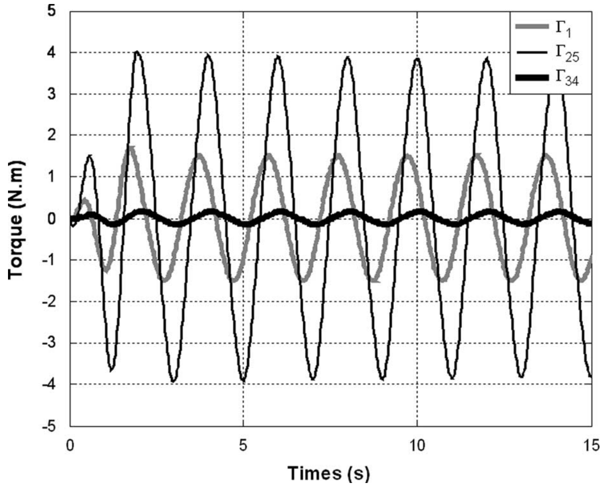

Fig. 8. Torques of joints 1, 25, and 34.

Figs. 7–10 show the joint trajectories and torques of three spherical joints. Fig. 11 shows the trajectory of the head in the $\mathbf{x}_w$–$\mathbf{z}_w$ plane of the world fixed frame and ten configurations of the eel during the simulation. The motion presented is carried out in 60 s. This second example has validated the possibility of 3-D motion.

VIII. Conclusion

This paper presents the inverse and direct dynamic modeling of a swimming eel-like robot. The dynamic models are developed using the recursive Newton–Euler formalism. The inverse model provides the acceleration of the head of the robot and the torque of the joints whereas the direct model provides the acceleration of the head and of the joints. These algorithms constitute the generalization of the algorithms of articulated manipulators to the case where the base is not fixed. The generalization takes into account the mobility of the base, which constitutes here the eel's head. Moreover, based on the literature of fluid mechanics, a simplified model of the fluid–structure contact is adopted. The presented research is currently being used to develop an

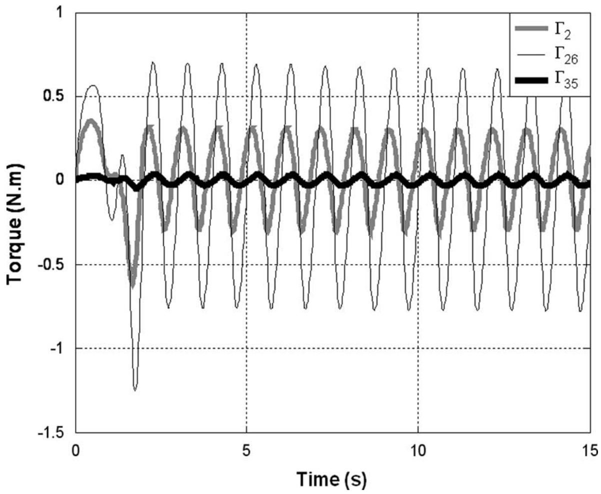

Fig. 9. Torques of joints 2, 26, and 35.

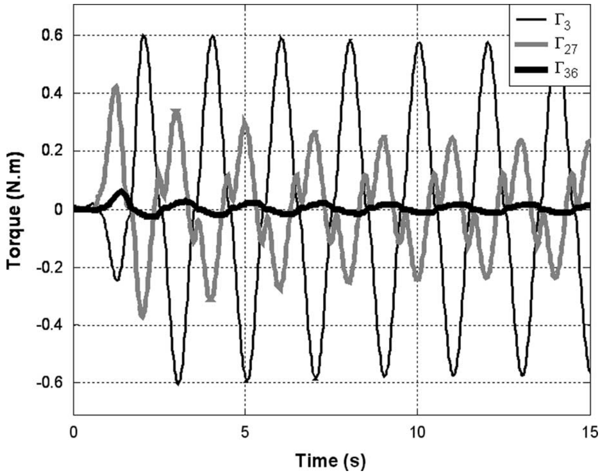

Fig. 10. Torques of joints 3, 27, and 36.

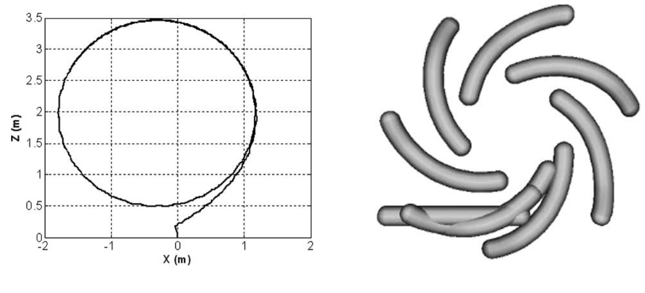

Fig. 11. Trajectory of the head origin frame in the $\mathbf{x}_w$–$\mathbf{z}_w$ world fixed frame and eel's configuration each of 6 s.

eel-like robot with the same dimension and with the same number of degrees of freedom as given in the simulated examples. However, the proposed algorithm can be used for other type of systems (snake robots, flying robots, etc.) by taking an appropriate external force model. Future work will involve the following.

1) The generalization of the given algorithms for hybrid structure, where the robot is composed of parallel modules, which are connected in series.
2) The development of a more accurate fluid contact model, which takes into account the articulated structure of the robot.

APPENDIX

OBTAINING $\ddot{q}_j$ AND ${}^{j}f_j$ IN TERMS OF ${}^{j-1}\dot{V}_{j-1}$

In this appendix, we detail the equations of the second and third recursive step of the direct dynamic model. The principal idea is to calculate $\ddot{q}_j$ and ${}^{j}f_j$ in terms of ${}^{j-1}\dot{V}_{j-1}$. Using (14), (15), and (17), we can write the equilibrium equation of motion of link j as

$$ {}^{j}f_{j} = {}^{j}J_{j}\,{}^{j}\dot{V}_{j} + {}^{j}\beta_{j} + {}^{j+1}T_{j}^{T}\,{}^{j+1}f_{j+1} \tag{A1}$$

with

$$ {}^{j}\dot{V}_{j} = {}^{j}\dot{V}_{j-1} + {}^{j}a_{j}\ddot{q}_{j} + {}^{j}\zeta_{j}. \tag{A2}$$

Thus

$$ {}^{j}f_{j} = {}^{j}J_{j}\left({}^{j}\dot{V}_{j-1} + {}^{j}a_{j}\ddot{q}_{j} + {}^{j}\zeta_{j}\right) + {}^{j+1}T_{j}^{T}\,{}^{j+1}f_{j+1} + {}^{j}\beta_{j}. \tag{A3}$$

As link n is a terminal link, then ${}^{n+1}f_{n+1} = 0$, using (A3) for $j = n$, we can obtain

$$ {}^{n}f_{n} = {}^{n}J_{n}\left({}^{n}\dot{V}_{n-1} + {}^{n}a_{n}\ddot{q}_{n} + {}^{n}\zeta_{n}\right) + {}^{n}\beta_{n}. \tag{A4}$$

To obtain $\ddot{q}_n$ in terms of ${}^{n-1}\dot{V}_{n-1}$, we use (28) giving

$$ {}^{j}a_{j}^{T}\,{}^{j}f_{j} = \tau_{j} - Ia_{j}\ddot{q}_{j} \tag{A5}$$

where

$$ \tau_{j} = \Gamma_{j} - F_{sj}\,\mathrm{sign}(\dot{q}_{j}) - F_{vj}\dot{q}_{j}. \tag{A6}$$

So, introducing (A4) into (A5), we obtain

$$ \ddot{q}_{n} = H_{n}^{-1}\left(-{}^{n}a_{n}^{T}\,{}^{n}J_{n}\left({}^{n}\dot{V}_{n-1} + {}^{n}\zeta_{n}\right) + \tau_{n} - {}^{n}a_{n}^{T}\,{}^{n}\beta_{n}\right). \tag{A7}$$

Then, by introducing (A7) into (A4)

$$ {}^{n}f_{n} = {}^{n}K_{n}\,{}^{n}\dot{V}_{n-1} + {}^{n}\alpha_{n} \tag{A8}$$

with

$$ H_{n} = {}^{n}a_{n}^{T}\,{}^{n}J_{n}\,{}^{n}a_{n} + Ia_{n} \tag{A9}$$

$$ {}^{n}K_{n} = {}^{n}J_{n} - {}^{n}J_{n}\,{}^{n}a_{n}H_{n}^{-1}\,{}^{n}a_{n}^{T}\,{}^{n}J_{n} \tag{A10}$$

$$ {}^{n}\alpha_{n} = {}^{n}K_{n}\,{}^{n}\zeta_{n} + {}^{n}J_{n}\,{}^{n}a_{n}H_{n}^{-1}\left(\tau_{n} - {}^{n}a_{n}^{T}\,{}^{n}\beta_{n}\right) + {}^{n}\beta_{n}. \tag{A11}$$

Equations (A7) and (A8) give $\ddot{q}_n$ and ${}^{n}f_n$ in terms of ${}^{n-1}\dot{V}_{n-1}$. For link $n-1$, we can rewrite equation (A1) (after replacing j by $n-1$) and using (A8) as

$$ {}^{n-1}J_{n-1}\,{}^{n-1}\dot{V}_{n-1} = {}^{n-1}f_{n-1} - {}^{n}T_{n-1}^{T}\,{}^{n}f_{n} - {}^{n-1}\beta_{n-1} \tag{A12}$$

which can be rewritten after substituting for ${}^{n}f_n$ from (A8) as

$$ {}^{n-1}f_{n-1} = {}^{n-1}J_{n-1}^{*}\left({}^{n-1}\dot{V}_{n-2} + {}^{n-1}a_{n-1}\ddot{q}_{n-1} + {}^{n-1}\zeta_{n-1}\right) + {}^{n-1}\beta_{n-1}^{*} \tag{A13}$$

with

$$ {}^{n-1}J^*_{n-1} = {}^{n-1}J_{n-1} + {}^{n}T^T_{n-1}\,{}^{n}K_n\,{}^{n}T_{n-1} \tag{A14} $$

$$ {}^{n-1}\beta^*_{n-1} = {}^{n-1}\beta_{n-1} + {}^{n}T^T_{n-1}\,{}^{n}\alpha_n. \tag{A15} $$

Equation (A13) has the same form as (A4); thus we can obtain

$$ {}^{n-1}f_{n-1} = {}^{n-1}K_{n-1}\,{}^{n-1}\dot{V}_{n-2} + {}^{n-1}\alpha_{n-1}. \tag{A16} $$

By generalizing this approach for the other links $(j = n, \ldots, 1)$

$$ {}^{j}f_j = {}^{j}K_j\,{}^{j}T_{j-1}\,{}^{j-1}\dot{V}_{j-1} + {}^{j}\alpha_j \tag{A17} $$

$$ \ddot{q}_j = H_j^{-1}\left[-{}^{j}a_j^T\,{}^{j}J_j^*\left({}^{j}\dot{V}_{j-1} + {}^{j}\zeta_j\right) + \tau_j - {}^{j}a_j^T\,{}^{j}\beta_j^*\right] \tag{A18} $$

with

$$ H_j = {}^{j}a_j^T\,{}^{j}J_j^{*j}a_j + I_{aj} \tag{A19} $$

$$ {}^{j}K_j = {}^{j}J_j^* - {}^{j}J_j^{*j}a_j H_j^{-1j}a_j^T\,{}^{j}J_j^* \tag{A20} $$

$$ \tau_j = \Gamma_j - F_{sj}\,\mathrm{sign}(\dot{q}_j) - F_{vj}\dot{q}_j \tag{A21} $$

$$ {}^{j}\alpha_j = {}^{j}K_j\,{}^{j}\zeta_j + {}^{j}J_j^{*j}a_j H_j^{-1}\left(\tau_j - {}^{j}a_j^T\,{}^{j}\beta_j^*\right) + {}^{j}\beta_j^* \tag{A22} $$

$$ {}^{j-1}J^*_{j-1} = {}^{j-1}J_{j-1} + {}^{j}T^T_{j-1}\,{}^{j}K_j\,{}^{j}T_{j-1} \tag{A23} $$

$$ {}^{j-1}\beta^*_{j-1} = {}^{j-1}\beta_{j-1} + {}^{j}T^T_{j-1}\,{}^{j}\alpha_j. \tag{A24} $$

We notice that, for the head $(j = 0)$, (A19)–(A22) are not valid since there is no motor on it. By using (A1) and the fact that${}^0f_0$ is equal to zero, we obtain

$$ 0 = {}^{0}J_0\,{}^{0}\dot{V}_0 + {}^{0}\beta_0 + {}^{1}T_0^{T\,1}f_1. \tag{A25} $$

By using (A17) for $j = 1$, (A25) becomes

$$ \begin{aligned} \mathbf{0} &= {}^{0}J_0\,{}^{0}\dot{V}_0 + {}^{0}\beta_0 + {}^{1}T_0^T\left({}^{1}K_1\,{}^{1}T_0\,{}^{0}\dot{V}_0 + {}^{1}\alpha_1\right) \\ \mathbf{0} &= {}^{0}J_0^{*0}\dot{V}_0 + {}^{0}\beta_0^*. \end{aligned} \tag{A26} $$

The resolution of this equation gives

$$ {}^{0}\dot{V}_0 = -\left({}^{0}J_0^*\right)^{-1}\,{}^{0}\beta_0^*. \tag{A27} $$

References

[1] K. A. McIsaac and J. P. Ostrowski, "A geometric approach to anguilliform locomotion modelling of an underwater eel robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May. 1999, vol. 4, pp. 2843–2848.

[2] J. P. Ostrowski and J. W. Burdick, "The geometric mechanics of undulatory robotics locomotion," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 683–701, 1998.

[3] R. Mason and J. W. Burdick, "Experiments in caranguiform robotic fish locomotion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 428–435.

[4] J. Y. S. Luh, M. W. Walker, and R. C. P. Paul, "On-line computational scheme for mechanical manipulator," *Trans. ASME, J. Dyn. Syst., Meas., Control*, vol. 102, no. 2, pp. 69–76, 1980.

[5] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *Int. J. Robot. Res.*, vol. 2, no. 1, pp. 13–30, 1983.

[6] F. Boyer and W. Khalil, "An efficient calculation of flexible manipulator inverse dynamics," *Int. J. Robot. Res.*, vol. 17, no. 3, pp. 282–293, 1998.

[7] F. Boyer, N. Glandais, and W. Khalil, "Flexible multibody dynamics based on non-linear Euler–Bernoulli kinematics," *Int. J. Numer. Methods Eng.*, no. 54, pp. 27–59, 2002.

[8] W. Khalil and E. Dombre, *Modeling Identification and Control of Robots*. London, U.K.: Hermes, Penton-Sciences, 2002.

[9] W. Khalil and D. Creusot, "SYMORO+: A System for the symbolic modelling of robots," *Robotica*, vol. 15, pp. 153–161, 1997.

[10] J. R. Morison, M. P. O'Brien, J. W. Johnson, and S. A. Shaaf, "The force exerted by surface waves on piles," *Trans. AIME*, vol. 189, pp. 149–154, 1950.

[11] A. Leroyer and M. Visonneau, "Numerical methods for RANSE simulations of a self-propelled fish-like body," *J. Fluids Struct.*, vol. 20, no. 7, pp. 975–991, Oct. 2005.

[12] G. Susbielles and C. Bratuch, *Vagues et ouvrages pétroliers en mer*. Paris, France: Technip, 1981.

[13] G. B. Gillis, "Environmental effects on undulatory locomotion in the american eel anguilla rostrata: Kinematics in water and on land," *J. Exp. Biol.*, vol. 201, pp. 949–961, 1998.

[14] E. D. Tytell and G. V. Lauder, "The hydrodynamics of eel swimming: I. Wake structure," *J. Exp. Biol.*, vol. 207, pp. 1825–1841, 2004.

[15] E. J. Kim and Y. Youm, "Design and dynamic analysis of fish robot: PoTuna," in *Proc. IEEE Conf. Robot. Autom.*, vol. 5, New Orleans, LA, Apr. 2004, pp. 4887–4892.

[16] F. Boyer, M. Porez, and W. Khalil, "Macro-continuous torque algorithm for a three-dimensional eel-like robot," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 763–775, Aug. 2006.

[17] J. Lighthill, "Large-amplitude elongated-body theory of fish locomotion," in *Proc. R. Soc. Lond. Ser.*, 1971, vol. B179, pp. 125–138.

[18] J. Yu, S. Wang, and M. Tan, "A simplified propulsive model of bio-mimetic robot fish and its realization," *Robot.*, vol. 23, pp. 101–107, 2005.

[19] S. F. Galls, "Development of a computational model for an underwater autonomous vehicle" Ph.D. dissertation, Texas A&M Univ., College Station, 2001.

[20] J. Katz and A. Plotkin, *Low Speed Aerodynamics*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[21] E. V. Lewis, *Principles of Naval Architecture. Vol. III, Motion in Waves and Controllability*. Jersey City, NJ: The Society of Naval Architects and Marine Engineers, 1989.

Wisama Khalil (SM'86) received the Ph.D. and the "Doctorat d'Etat" degrees in robotics and control engineering from the University of Montpellier, France, in 1976 and 1978, respectively.

Since 1983, he has been a Professor at the Automatic Control and Robotics Department, Ecole Centrale de Nantes, France. He is carrying out his research within the Robotics team, Institut de Recherche en Communication et Cybernétique de Nantes (IRCCyN). His current research interests include modeling, control, and identification of robots.

Guillaume Gallot received the Diploma in engineering and the Master of Research in robotics, in 2004 from the Ecole Centrale de Nantes, France. He is currently working toward the Ph.D. degree of the Ecole Centrale de Nantes and Université de Nantes at the Institut de Recherche en Communication et Cybernétique de Nantes (IRCCyN) on the dynamic modeling and control of eel-like robots.

Frédéric Boyer received the Diploma in mechanical engineering and the Master of Research in mechanics, in 1991 from the Institut National Polytechnique de Grenoble, France. He received the Ph.D. degree in robotics from the University Paris VI, France, in 1994.

Since 1995, he has been an Assistant Professor at the Automatic Control Department, Ecole des Mines de Nantes, Nantes, France. He is carrying out his research within the Robotics team, Institut de Recherche en Communication et Cybernétique de Nantes (IRCCyN). His current research interests include structural dynamics, geometric mechanics, and biomimetic robotics.