2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A hybrid dynamic model for bio-inspired soft robots - Application to a flapping-wing micro air vehicle.

Mathieu Porez, Frédéric Boyer and Ayman Belkhiri

***Abstract*— The paper deals with the dynamic modeling of bio-inspired robots with soft appendages such as flying insect-like or swimming fish-like robots. In order to model such soft systems, we propose to use the Mobile Multibody System framework introduced in [1], [2], [3]. In such a framework, the robot is considered as a tree-like structure of rigid bodies where the evolution of the position of the joints is governed by stress-strain laws or control torques. Based on the Newton-Euler formulation of these systems, we propose a new algorithm able to compute at each step of a time loop both the net and passive joint accelerations along with the control torques supplied by the motors. To illustrate, based on previous work [4], the proposed algorithm is applied to the simulation of the hovering flight of a soft flapping-wing insect-like robot (see the attached video).**

I. INTRODUCTION.

As revealed by works in biology such as those of Alexander [5], animals have developed soft organs to improve their locomotion performances. As an example, in the case of flying insects, such as hawk moth, the twisting strain of the wing along the leading edge generates a phase lag between the stroke and the pitch which is at the origin of the lift during flight [6]. Another relevant example of the benefits of compliances in animal locomotion is illustrated by the dead fish in a wake. In fact, recent experiments [7] and simulations [8] reveal that a dead trout placed in the wake downstream from obstacles can extract energy passively from large-scale coherent vortices and ascend flow. Based on these two examples, it appears that, soft organs allow animals: 1) to add useful degrees of freedom for locomotion without adding muscles; 2) to cyclically accumulate and restore kinetic energy in order to minimize the power consumption during the locomotion. From the roboticist's view point, the implementation of these concepts would allow to design simpler, lighter and cheaper robots. As a result, the reproduction of compliant wings of flying insects is the key of success of the new generation of Micro Air Vehicles (MAV) [9], [10]. To help researches to study soft locomotion, we propose, in this paper, a Mobile Multibody Systems (MMS) framework devoted to the dynamic modelling and simulation of locomotion systems which use soft appendages. The proposed algorithm allows to solve numerically the three following coupled dynamics: 1) the external forward dynamics ruling the net motions of the MMS produced by locomotion through a model of the contact forces with environment; 2) the internal inverse dynamics ruling the internal control torques produced by the shape motions of the MMS; 3) the internal forward dynamics ruling the strains of the compliant organs. While being applicable to a wide diversity of systems, the algorithm is illustrated on the case of the hovering flight of a soft MAV inspired of big moths of *Sphyngidae* family like *Manduca sexta* (see the attached video). In order to present this framework, the article is structured as follows. The modelling of a MMS is first presented in section II. In section III, a hybrid algorithm dedicated to compute the forward and the inverse dynamics of the MMS is introduced. Then, the resulting simulator is exploited in section IV, in the context of the flapping flight. Lastly, the article ends by some concluding remarks.

Mathieu Porez, Frédéric Boyer and Ayman Belkhiri are with IRCCyN, EMN - La Chantrerie - 4, rue Alfred Kastler,B.P. 20722, 44307 Nantes, Cedex 3, France.
Contact: mathieu.porez@mines-nantes.fr

II. THE PROBLEM STATEMENT.

A. Parametrisation and notations.

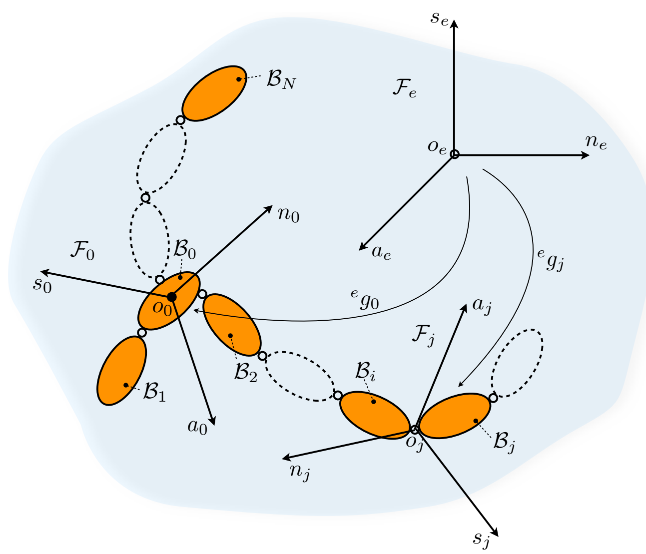

Fig. 1. Schematic view of a Mobile Multibody System.

In accordance with figure 1, let us consider a MMS with a tree-like structure, in a 3-D space of an unbounded volume filled of an initially quiescent fluid (e.g. air, water, etc ...). We attach to the ambient geometric space a fixed spatial orthonormed frame denoted by $\mathcal{F}_e = (O_e, s_e, n_e, a_e)$, where a_e supports the vertical axis and the plane (O_e, s_e, n_e) defines the ground. The considered MMS is composed of a sequence of $N+1$ rigid bodies interconnected through N passive or actuated 1-DoF angular joints. These bodies are denoted $\mathcal{B}_0, \mathcal{B}_1, ..., \mathcal{B}_N$, where $\mathcal{B}_0$ stands for the reference body. Moreover, the bodies are numbered from $\mathcal{B}_0$ toward the tips of the branches in increasing order. In the following, we denote by j and i, the indices of the current body and its antecedent respectively. Moreover, we defined by $\mathcal{J}_a$ the index set of actuated joints and by $\mathcal{J}_p$ the index set of passive joints. We attach to each body $\mathcal{B}_j$ of density ρ_j a mobile frame $\mathcal{F}_j = (O_j, s_j, n_j, a_j)$, where the center O_j coincides with the center of the joint j, and a_j supports the joint axis. Let us note that s_j and n_j are directed as required. At any time t, the robot configuration is defined by the vector of joint positions $r = (r_1, ..., r_n)^T$ defining the relative angles around the joint axis between the bodies, together with the orientation matrix ${}^eR_0$ and the position vector ${}^eP_0$ of the mobile frame attached to the reference body $\mathcal{F}_0 = (O_0, s_0, n_0, a_0)$ with respect to $\mathcal{F}_e$. The time evolution of $({}^eR_0, {}^eP_0)$ defines the rigid net motion of the MMS. Finally, throughout this article, we will use the following notation convention. For any physical variable modelled by a tensor, the right lower index will represent the body index (to which it is related) while the left upper exponent will indicate the index of the projection frame (e.g. ${}^eR_0$, ${}^eP_0$). When the tensor related to a body is expressed in the mobile frame of this body, the upper index is omitted. Moreover, the temporal derivative $\partial./\partial t$ will be sometimes denoted by a 'dot'.

B. Mobile Multibody System model.

To model the MMS presented previously, we use the Newton-Euler (N-E) framework proposed in [1], [2], [3]. This general setting is devoted to the modelling of MMS, i.e. Multibody Systems with a mobile basis (here $\mathcal{B}_0$) whose motion is governed by the locomotion dynamics. Let us start by introducing the geometric model of the MMS which relates the posture of any frame $\mathcal{F}_j$ with that of the antecedent frame $\mathcal{F}_i$, both expressed in the earth frame $\mathcal{F}_e$ and represented by the two (4×4) matrices ${}^eg_i$ and ${}^eg_j$ of $SE(3)$. This model can be detailed as:

$$ {}^eg_j = {}^eg_i \; {}^ig_j(r_j) = {}^eg_i \begin{pmatrix} {}^iR_j(r_j) & {}^iP_j \\ 0 & 1 \end{pmatrix} , \qquad (1) $$

where ${}^iR_j$ and ${}^iP_j$ are the orientation matrix and the position vector of $\mathcal{F}_j$ with respect to $\mathcal{F}_i$.

Regarding the velocity of the body j, it is a (6×1) vector of $se(3)$ denoted η_j and related to the velocity of the antecedent body i through the recursive relation:

$$ \eta_j = (V_j^T, \Omega_j^T)^T = Ad_{{}^jg_i} \eta_i + \dot{r}_j A_j , \qquad (2) $$

where V_j and Ω_j are respectively the linear and angular Galilean velocities of the considered body, both expressed in its mobile frame, $A_j = (0_3^T, a_j^T)^T$ is the (6×1) unit vector supporting the joint axis j, and $Ad_{{}^jg_i}$ is the adjoint map operator which permits a change in velocity from $\mathcal{F}_i$ to $\mathcal{F}_j$ [11]:

$$ Ad_{{}^jg_i} = \begin{pmatrix} {}^jR_i & {}^jR_i \, {}^i\hat{P}_j^T \\ 0 & {}^jR_i \end{pmatrix} . \qquad (3) $$

Let us remark that in (3), we introduced the 'hat' notation which changes a (3×1) vector into its associated (3×3) skew-symmetric tensor. Thus, for any vectors A and B in $\mathbb{R}^3$, $\hat{A}$ is defined such that $\hat{A}B = A \times B$.

Once the Galilean velocities are defined, by time derivation of (2), the acceleration of $\mathcal{B}_j$ is given by the relation:

$$ \dot{\eta}_j = Ad_{{}^jg_i} \dot{\eta}_i + \zeta_j + \ddot{r}_j A_j , \qquad (4) $$

where ζ_j represents the component of accelerations in (4) which depends on velocities through the detailed expression:

$$ \zeta_j = \begin{pmatrix} ({}^jV_i + {}^jP_i \times {}^j\Omega_i) \times \dot{r}_j a_j \\ \dot{r}_j \, {}^j\Omega_i \times a_j \end{pmatrix} . \qquad (5) $$

Finally, by applying the Newton's law and the Euler's theorem on the j^{th} body, one obtains the dynamic equations of $\mathcal{B}_j$ in the Newton-Euler form:

$$ f_j = \mathcal{M}_j \dot{\eta}_j + \beta_j + f_{ext,j} + \sum_k Ad^T_{{}^kg_j} f_k , \qquad (6) $$

where k are the indices of all the successive bodies to $\mathcal{B}_j$. Moreover, in (6), we introduced the following notations:

- for any j, f_j is the (6×1) force vector (element of $se(3)^*$) exerted by $\mathcal{B}_i$ onto $\mathcal{B}_j$; and
- $\mathcal{M}_j$ is the (6×6) inertia tensor of $\mathcal{B}_j$ (element of $se(3)^* \otimes se(3)$), which can be detailed as:

$$ \begin{aligned} \mathcal{M}_j &= \begin{pmatrix} M_j & -MS_j \\ MS_j & I_j \end{pmatrix} \qquad (7) \\ &= \rho_j \int_{V_{\mathcal{B}_j}} \begin{pmatrix} 1_3 & -\hat{O_jQ} \\ \hat{O_jQ} & -\hat{O_jQ}\hat{O_jQ} \end{pmatrix} dV_{\mathcal{B}_j} , \end{aligned} $$

  where Q is a point of $\mathcal{B}_j$, 1_3 is the 3×3 unit matrix, while M_j, MS_j and I_j are the tensor of body mass (spherical in the rigid body case), the tensor of first inertia moments (skew-symmetric in the rigid body case) and the tensor of angular inertia of $\mathcal{B}_j$; and
- the (6×1) vector of Coriolis and centrifugal forces:

$$ \beta_j = \begin{pmatrix} -\Omega_j \times (MS_j \Omega_j) + \Omega_j \times (M_j V_j) \\ \Omega_j \times (I_j \Omega_j) + MS_j(\Omega_j \times V_j) \end{pmatrix} , \qquad (8) $$

  (we shall see that the fluid generates other velocity-dependent inertia forces that we will also denote β by extension); and
- the (6×1) vector of external forces denoted by $f_{ext,j}$ whose the model depends on the considered locomotion problem.

Let us note that, for $j = 0$, (6) describes the time evolution of the MMS net motion and it is named the external forward dynamic model.

III. THE HYBRID ALGORITHM.

A. The algorithm working process.

In accordance with the assumptions of section II, we address the following dynamic problem: knowing at each time t, the state of the MMS $({}^eg_0, {}^e\eta_0, r, \dot{r})$, the accelerations $\ddot{r}_j$ (for $j \in \mathcal{J}_a$) applied to the actuated joints through

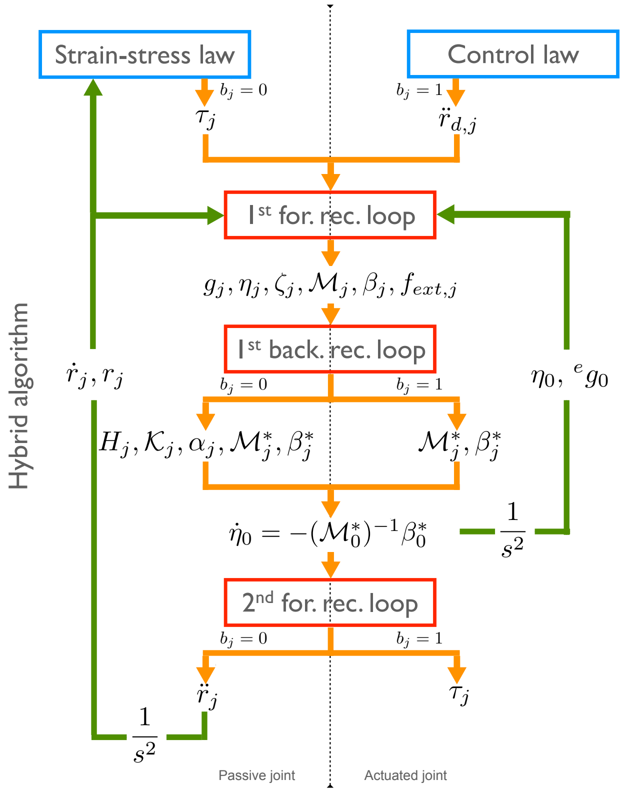

Fig. 2. Flow chart of the proposed hybrid algorithm.

motion control laws and the torques τ_j (for $j \in \mathcal{J}_p$) applied to the passive joints through stress-strain material laws or control torque; the dynamic problem consists in calculating the accelerations of the reference body $\dot{\eta}_0$ (describing the net motion of the MMS with respect to the Galilean frame $\mathcal{F}_e$), the torques τ_j (for $j \in \mathcal{J}_a$) applied on the actuated joints and the accelerations $\ddot{r}_j$ (for $j \in \mathcal{J}_p$) of passive joints. This dynamic problem is named mixed dynamic problem since it involves the forward and the inverse forms of the dynamic of a multi-body system [1]. To resolve such a problem, we propose here to extend to the MMS with a tree-like structure, the inverse algorithm of Luh et *al.* [12] and the forward algorithm of Featherstone [13] both dedicated initially to the manipulators. Due to its mixed (inverse, forward) nature, the resulting algorithm will be named "hybrid algorithm" and its flow chart is described in figure 2. From a computational point of view, this algorithm, resolves, at each time step of a time integration loop, three recursive sets of equations on the bodies index. The first loop is a forward recursive loop (from the reference body to the tips of the branches of the considered tree-like structure), which compute all the state dependent variables related to subsequent computing as the transformation matrices, velocities, inertia tensors, etc ... . It is followed by a backward loop (from the tips of the branches to the reference body) which computes $\dot{\eta}_0$, i.e. solves the external forward dynamic model. To do this, the recursive process computes the (6×6) inertia matrix of the whole MMS: $\mathcal{M}_0^*$, and β_0^* the (6×1) vector of all external and inertia forces applied on the MMS. Finally, once the variables are known, they allow to compute the current acceleration of the reference body as follows:

$$\dot{\eta}_0 = -\left(\mathcal{M}_0^*\right)^{-1} \beta_0^* \ , \tag{9}$$

which is used to initialize the last forward recursive loop (see the flow chart in figure 2) dedicated to the internal dynamics. This loop computes the accelerations of the passive joints and the torque applied on the actuated joints, which are the expected outputs allowing to update (after a time integration) the external state (i.e. $({}^e g_0,\ {}^e\eta_0)$ and the internal state ($r_j, \dot{r}_j$ for $j \in \mathcal{J}_p$) of the MMS before to increment the time and to begin the next iteration.

Before detailing the three loops previously presented, let us introduce the following Boolean variable defining the type of the j^{th} joint, i.e. for "actuated" or "passive" type:

$$\forall j,\ b_j = \begin{cases} 1 \text{ if } \ddot{r}_j(t) \text{ is imposed and } \tau_j(t) \text{ is unknown;} \\ 0 \text{ if } \tau_j(t) \text{ is imposed and } \ddot{r}_j(t) \text{ is unknown.} \end{cases}$$

B. *The first forward recursion on the kinematics*

Since the current robot's state $({}^e g_0,\ {}^e\eta_0, r, \dot{r})$ is known, the algorithm starts by the following forward recursion:

For $j = 0, 1, ..., N$, computes:

- ${}^i R_j$, ${}^i P_j$ and the body transformations ${}^e g_j$ from (1);
- the body velocities η_j from (2);
- the terms ζ_j of (4) from (5);
- the body inertia matrices $\mathcal{M}_j$ from (7);
- the body Coriolis and centrifugal forces β_j from (8);
- the external forces $f_{ext,j}$ whose the model depends on the studied problem;

and initializes:

- the generalized inertia matrix $\mathcal{M}_j^*$ from :

$$\mathcal{M}_j^* = \mathcal{M}_j \ ; \tag{10}$$

- the generalized forces β_j^* from :

$$\beta_j^* = \beta_j + f_{ext,j} \ . \tag{11}$$

End for.

C. *The backward recursion on the external forward dynamics*

Once all the state-dependent variables are known, the next step of the computational algorithm consists in executing the following recursion:

For $j = N, N-1, ..., 1$, computes:

- If $b_j = 1$:

$$\begin{aligned} \mathcal{M}_i^* &= \mathcal{M}_i^* + Ad_{{}^j g_i}^T \mathcal{M}_j^* Ad_{{}^j g_i} \ ; \\ \beta_i^* &= \beta_i^* + Ad_{{}^j g_i}^T (\mathcal{M}_j^* (A_j \ddot{r}_j + \zeta_j) + \beta_j^*) \ . \end{aligned}$$

- Else (if $b_j = 0$) :

$$
\begin{aligned}
H_j &= A_j^T \mathcal{M}_j^* A_j \ ; \\
\mathcal{K} &= \mathcal{M}_j^* - \mathcal{M}_j^* (A_j H_j^{-1} A_j^T) \mathcal{M}_j^* \ ; \\
\alpha &= \mathcal{K} \zeta_j + \mathcal{M}_j^* A_j H_j^{-1} (\tau_j - A_j^T \beta_j^*) + \beta_j^* \ ; \\
\mathcal{M}_i^* &= \mathcal{M}_i^* + Ad^T_{{}_jg_i} \mathcal{K} Ad_{{}_jg_i} \ ; \\
\beta_i^* &= \beta_i^* + Ad^T_{{}_jg_i} \alpha \ .
\end{aligned}
$$

- End if.

End for.

Once this recursion loop is carried out, the accelerations $\dot{\eta}_0$ of $\mathcal{B}_0$ are computed from (9).

D. The second forward recursion loop on the internal (inverse and forward) dynamics

Finally, the algorithm ends with a second forward recursion initialised by the current state and $\dot{\eta}_0$:

For $j = 1, 2, ..., N$, computes:

$$\dot{\eta}_j = Ad_{{}_jg_i} \dot{\eta}_i \ ;$$

- If $b_j = 1$:

$$
\begin{aligned}
\dot{\eta}_j &= \dot{\eta}_j + A_j \ddot{r}_j + \nu_j \ ; \\
\tau_j &= A_j (\mathcal{M}_j^* \dot{\eta}_j + \beta_j^*) \ .
\end{aligned}
$$

- Else (if $b_j = 0$) :

$$
\begin{aligned}
\ddot{r}_j &= H_j^{-1} (\tau_j - A_j^T (\mathcal{M}_j^* (\dot{\eta}_j + \nu_j) + \beta_j^*)) \ ; \\
\dot{\eta}_j &= \dot{\eta}_j + A_j \ddot{r}_j + \nu_j \ .
\end{aligned}
$$

- End if.

End for.

Lastly, in order to update the external state of the MMS, for the next iteration of the time loop, $\dot{\eta}_0$ is numerically integrated with a numerical integrator based on quaternions. As regards the internal state, i.e. $\dot{r}_j$ and r_j, they are updated by time integration of $\ddot{r}_j$.

IV. Application to the flapping flight.

In this section, we propose to apply the algorithm, introduced in section III, to the simulation of the hovering flight of a flapping-wing insect-like robot bio-inspired from the hawk moth *Manduca sexta* (see the attached video).

A. The robot parametrisation.

The considered robot is composed of a rigid thorax and two soft wings where the deformations are concentrated along the leading edge (see figure 3). For this numerical example, only the twisting around the leading edge and the bending in the plane perpendicular to the wing have been taken into account. In order to model such a soft system with the above general framework (see section III),

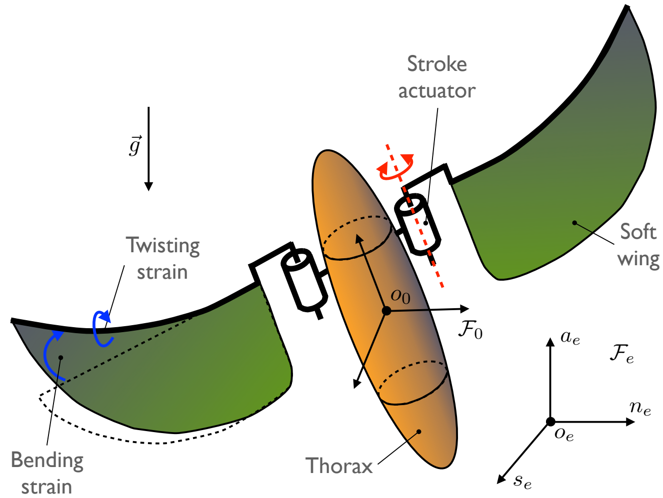

Fig. 3. Schematic view of the soft MAV.

we propose to discretize each wing of the robot into a serial assembly of rigid bodies. The discretization of each of the wings of span S consists in dividing them into M sections with a length $l = S/M$. Each section is composed of the following serial assembly: 1) a 1-DoF angular joint aligned with the leading edge (modeling the twisting); 2) a fictitious rigid body with no inertia; 3) a 1-DoF angular joint for which the axis is in the wing plane and orthogonal to the leading edge (modeling the bending); 4) a rigid body or "blade" whose size and inertia are the same as those of the considered section. Once so discretized, the virtual robot has $N = 4M + 1$ bodies and $4M$ angular joints. To illustrate this, figure 4 shows a virtual robot with $M = 2$ sections and $N = 9$ bodies.

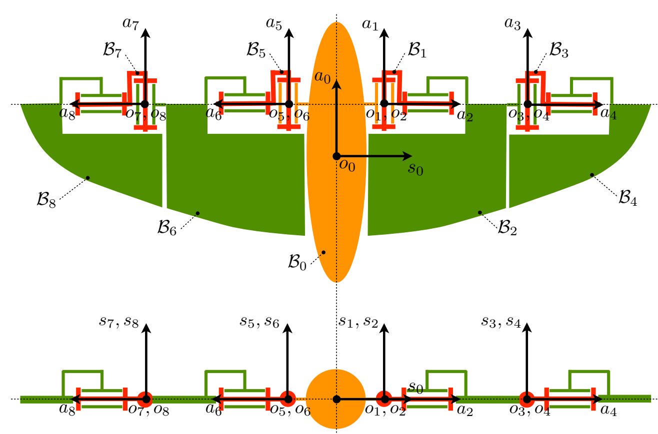

Fig. 4. Front and top views of a virtual robot at $N = 9$ bodies and $M = 2$ sections per wing.

In accordance with figure 5, the thorax, which is defined as the reference body, is denoted by $\mathcal{B}_0$ while the rigid bodies constituting the right and the left wings are denoted $\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_{2M}$ and $\mathcal{B}_{2M+1}, \mathcal{B}_{2M+2}, ..., \mathcal{B}_{4M}$ respectively. The body $\mathcal{B}_0$ of density ρ_t, is an ellipsoid of the half-axes

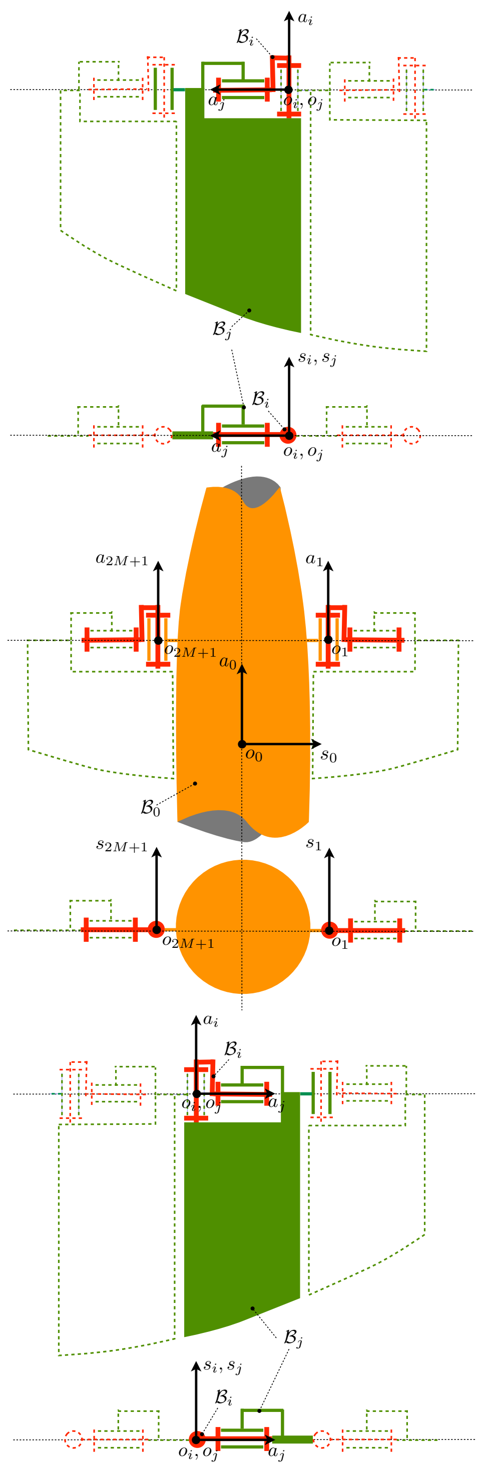

Fig. 5. Schematic view of the flapping-wing insect-like robot. On top: the left wing; in the middle: the thorax; at the bottom: the right wing.

a, b and c along the vectors s_0, n_0, and a_0 of the frame $\mathcal{F}_0 = (O_0, s_0, n_0, a_0)$ attached to the geometric center of $\mathcal{B}_0$. The wings are attached to $\mathcal{B}_0$ at d from O_0 along s_0. The blades, i.e. the bodies $j \in \{2, 4, ..., 4M\}$, have a length $l = S/M$ (along a_j), a cord c_j (along n_j), a thickness e (along s_j) and a density ρ_w. The wing having an elliptical shape, the cord c_j of $\mathcal{B}_j$ is a function of the position X_j of the section along the leading edge (from the root to the tip) which is defined as follow:

$$c_j = C\sqrt{1 - X_j^2/S^2} \ ,$$

where C is the cord of the wing at its roots. As far as the joints linking the wings to the thorax (i.e. $j \in \{1, 2M+1\}$) are concerned, they are actuated to generate the typical stroke observed in the hawk moth [14] according to the following time law:

$$\ddot{r}_{d,j} = \begin{cases} -A\omega^2 \cos(\omega t) \text{ , if } j = 1 \ , \\ A\omega^2 \cos(\omega t) \text{ , if } j = 2M+1 \ , \end{cases}$$

where A and ω are the amplitude and the frequency of the stroke respectively. As regards the other joints of the robot, they are all passive. Thus, the strain accelerations $\ddot{r}_j$ of these joints are unknown while the strain torques τ_j are imposed through strain-stress law function of the strain state, i.e. $(r_j, \dot{r}_j)$. More precisely, the strain torques applied to the joints numbered $j \in \{3, 5, ..., 2M-1\} \cup \{2M+3, 2M+5, ..., 4M-1\}$, modeling the twist of the wings, are ruled by the following viscous-elastic model:

$$\tau_j = -k_{t,j} r_j - \mu \dot{r}_j \ , \tag{12}$$

while the strain torques assigned to the joints numbered $j \in \{2, 4, ..., 2M\} \cup \{2M+2, 2M+4, ..., 4M\}$, which model the bending, are governed by:

$$\tau_j = -k_{b,j} r_j - \mu \dot{r}_j \ . \tag{13}$$

In (12)-(13), we introduced μ the structural damping together with $k_{t,j}$ and $k_{b,j}$ the stiffness of twisting and bending respectively defined along the leading edge by the following linear functions:

$$k_{t,j} = k_t^1 + \frac{X_j}{S}(k_t^2 - k_t^1) \ , \ k_{b,j} = k_b^1 + \frac{X_j}{S}(k_b^2 - k_b^1) \ ,$$

with k_t^1 and k_t^2 (k_b^1 and k_b^2) the stiffness of twisting (of bending) at the root and the tip respectively.

B. Model of the external forces.

Let us now specify the model of external forces chosen for this dynamic problem. It is the following:

$$f_{ext,j} = f_{g,j} + f_{aero,j} \ , \tag{14}$$

where $f_{g,j}$ is the (6×1) vector of gravity forces applied on $\mathcal{B}_j$ and $f_{aero,j}$ is the (6×1) vector of aerodynamic forces ($\neq 0$, only for the blade bodies of the wing). In this numerical example, based on the quasi-steady model of Dickinson & *al* [6], [15], we distinguish two types of aerodynamic forces: 1) the added mass forces due to the fluid inertia; 2) the quasi-steady forces of lift and drag whose time dependance is due to the body kinematics and not to the fluid flow history. Based on these considerations, $f_{aero,j}$ is defined as:

$$f_{aero,j} = f_{a,j} + f_{s,j} \ ,$$

where $f_{a,j}$ and $f_{s,j}$ are the (6×1) vector of added mass forces and the (6×1) vector of quasi-steady forces respectively.

To establish the model of aerodynamic forces, let us consider the wing blade $\mathcal{B}_j$. We define by ξ the abscissa of a cross-section of $\mathcal{B}_j$ along the leading edge. On each cross-section, along the cord, at a distance of $0.4c_j$ from the leading edge, we fixe the center of pressure C_p where the quasi-steady forces (i.e. the lift and drag forces) are applied. Moreover, on C_p, we attach two unit vectors t and w. t belongs to the blade plane and oriented from the trailing edge to the leading edge while w is orthogonal to the blade plane and oriented from the intrados to the extrados. Based on these definitions, the model of quasi-steady forces can be detailed as:

$$f_{s,j} = \int_0^l \begin{pmatrix} 1 & 0 \\ O_j\hat{C}_p & 1 \end{pmatrix} \begin{pmatrix} L+D \\ 0 \end{pmatrix} d\xi \ ,$$

where L et D are the lift and the drag forces respectively defined by:

$$L = \frac{1}{2}\rho_{air} c_j C_L ||V_{C_p(\xi)}||^2 v \ , \ D = \frac{1}{2}\rho_{air} c_j C_D ||V_{C_p(\xi)}||^2 u \ ,$$

where $V_{C_p(\xi)}$ is the linear speed of C_p, $v = V_{C_p(\xi)}/||V_{C_p(\xi)}||$, $u = v \times (t \times v)$, ρ_{air} is the density of the air while C_L and C_D are the coefficients of the lift and the drag respectively obtained from experiments [6]:

$$C_L = 1.8 \sin 2\beta \ , \text{and } C_D = 1.92 - 1.55 \cos 2\beta \ ,$$

with $\beta = atan2(-w^T.v, -t^T.v)$ the incidence angle of the ξ-cross-section and the air flow speed.

As regards the vector of added mass forces $f_{a,j}$, it can be simply derived from a kinetic momenta balance applied to the fluid which laterally bounds $\mathcal{B}_j$. Assuming that the fluid is perfect (inviscid and incompressible) and irrotational, we derive the fluid dynamics from the impulse-momentum theory of the fluid mechanic due to Kelvin and Kirchhoff [16]. Moreover, considering the aspect ratio of wings, $f_{a,j}$ can be detail as (for more details see appendix in [3]):

$$f_{a,j} = \mathcal{M}_{a,j}\dot{\eta}_j + \beta_{a,j} \ ,$$

where we introduced the following definitions:

- $\mathcal{M}_{a,j}$ is the (6×6) tensor of added inertia of the fluid accelerated by $\mathcal{B}_j$:

$$\begin{aligned} \mathcal{M}_{a,j} &= \begin{pmatrix} M_{a,j} & -MS_{a,j} \\ MS_{a,j} & I_{a,j} \end{pmatrix} \\ &= \int_0^l \begin{pmatrix} m_a & -m_a O_j\hat{C}_p \\ m_a O_j\hat{C}_p & -m_a O_j\hat{C}_p O_j\hat{C}_p \end{pmatrix} d\xi \ , \end{aligned}$$

  which only depends on the cross sectional added inertia tensors $m_a = \rho_{air}\pi c_j^2 w.w^T$; and
- $\beta_{a,j}$ is the (6×1) vector of added mass forces produced by the volume of fluid accelerated by the Coriolis and centrifugal accelerations of $\mathcal{B}_j$:

$$\begin{aligned} \beta_{a,j} &= \begin{pmatrix} -\Omega_j \times (MS_{a,j}\Omega_j) + \Omega_j \times (M_{a,j}V_j) \\ \Omega_j \times (I_{a,j}\Omega_j) + MS_{a,j}(\Omega_j \times V_j) \end{pmatrix} \\ &+ \begin{pmatrix} 0 \\ V_j \times (M_{a,j}V_j) \end{pmatrix} . \end{aligned}$$

Finally, (14) can be written as follows:

$$f_{ext,j} = f_{g,j} + \mathcal{M}_{a,j}\dot{\eta}_j + \beta_{a,j} + f_{s,j} \ . \tag{15}$$

Let us note that in (15), $f_{ext,j}$ is a function of $\dot{\eta}_j$ which is still unknown when (15) is evaluated by the hybrid algorithm. To overcome this problem, we replace (10) and (11) by:

$$\mathcal{M}_j^* = \mathcal{M}_j + \mathcal{M}_{a,j} \text{ and } \beta_j^* = \beta_j + f_{g,j} + \beta_{a,j} + f_{s,j} \ ,$$

respectively.

C. Results and Discussions

TABLE I

SIMULATION PARAMETERS.

Parameter	Value	Parameter	Value
M	4	ρ_t	800 Kg/m^3
N	17	ρ_w	1400 Kg/m^3
S	70×10^{-3} m	ρ_{air}	1.22 Kg/m^3
C	31.5×10^{-3} m	A	$41.3\pi/180$ rad
e	0.1×10^{-3} m	ω	50π rad/s
a	8.5×10^{-3} m	k_t^1	3×10^{-3} Nm/rad
b	8.5×10^{-3}m	k_t^2	9×10^{-3} Nm/rad
c	28×10^{-3} m	k_b^1	36×10^{-3} Nm/rad
d	18.2×10^{-3} m	k_b^2	9×10^{-3} Nm/rad
l	1.75×10^{-3} m	μ	1×10^{-5} Nm.s/rad

To illustrate the hybrid algorithm of section III, we realised a flapping flight simulator using the description of section IV-A and the numerical parameters of table I. For this simulation, the hybrid algorithm has been programmed using MATLAB ®. The developed simulator uses the predictor-corrector method (with a fourth-order explicit method for the prediction step and a fifth-order implicit method for the correction step) for the time integration (with a time step of 1×10^{-4} s) together with the Gaussian quadrature method (at 6 points) for the spatial integration of the external force model defined in section IV-B. With this model, these parameters and these numerical tools, we can simulate the dynamics of a flapping-wing insect-like robot during one flapping period, i.e. for $T = 2\pi/\omega = 0.04$ s, in 17 s on a laptop (CPU Intel ® Dual-Core I7 @2.66GHz). Finally, the initial conditions have been chosen to obtain the periodicity of the hovering flight.

Figure 6 shows a set of snapshots taken at regular time during one flapping cycle (see the attached video). On the snapshots numbered 0-1 and 5-6, we observe that during the transition between the downstroke and the upstroke (and vice-versa), thanks to their flexibility, the wings twist around the leading edge and bend in the opposite direction of the stroke. These twisting and bending deformations are characteristic of the flapping flight and are similar to those observed in the hawk moth *Manduca sexta* (see [14] and

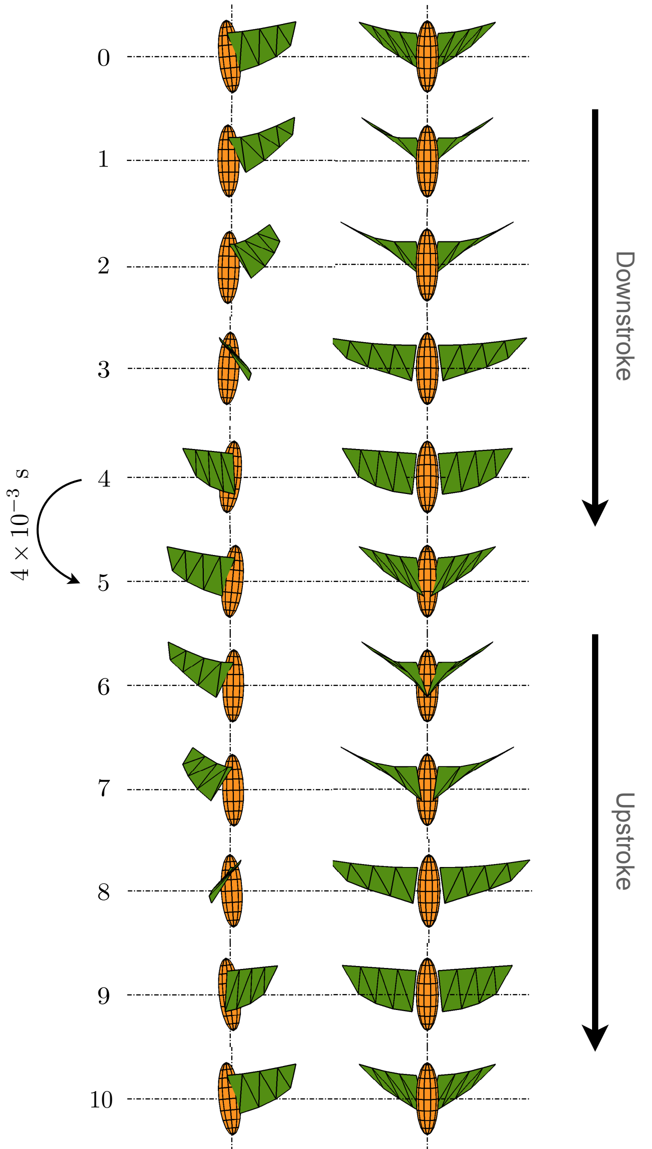

Fig. 6. Sagittal and coronal views of the flapping-wing insect-like robot for one stroke.

the video attached with [17]). As regards the net motion of the robot, the linear motions of the thorax along the vectors n_e and a_e have amplitudes of ± 2 mm and ± 0.5 mm respectively while the angular pitch motion has an amplitude of ± 7 degrees. Moreover, we have plotted in figure 7 the linear and angular speeds of $\mathcal{B}_0$ in the sagittal plane of the robot.

As far as the wing deformations are concerned, figure 8 shows them. We observe that the maximum twisting strain appears (in the middle of figure 8) when the stroke angle is equal to zero (i.e. when the wings are aligned with the thorax) while the maximum bending arises (on top of figure 8) after the stroke reversals (when the wings rotate and change direction). By adding all the relative stain angles, the twisting rotation of each wing tip has ± 79

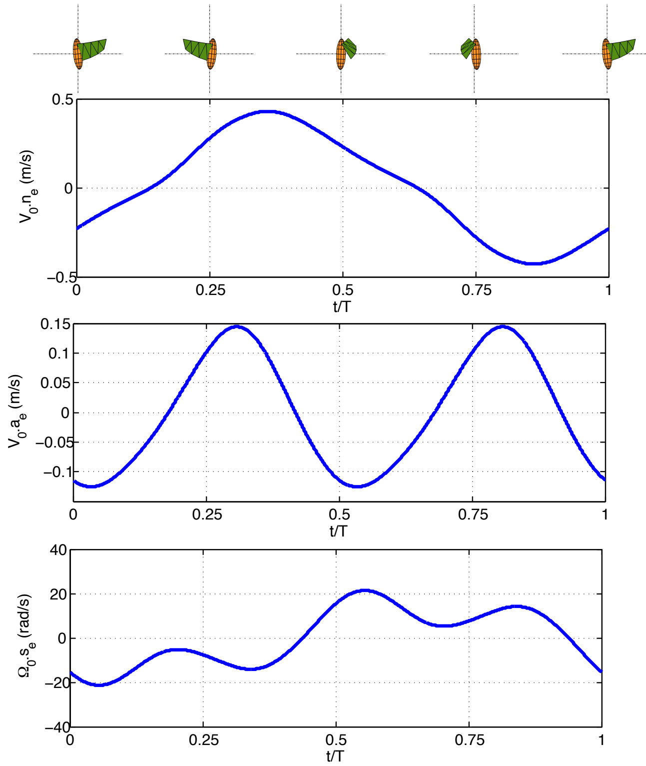

Fig. 7. The time evolution of the linear and angular speeds of $\mathcal{B}_0$ in the sagittal plane of the robot for one flapping cycle.

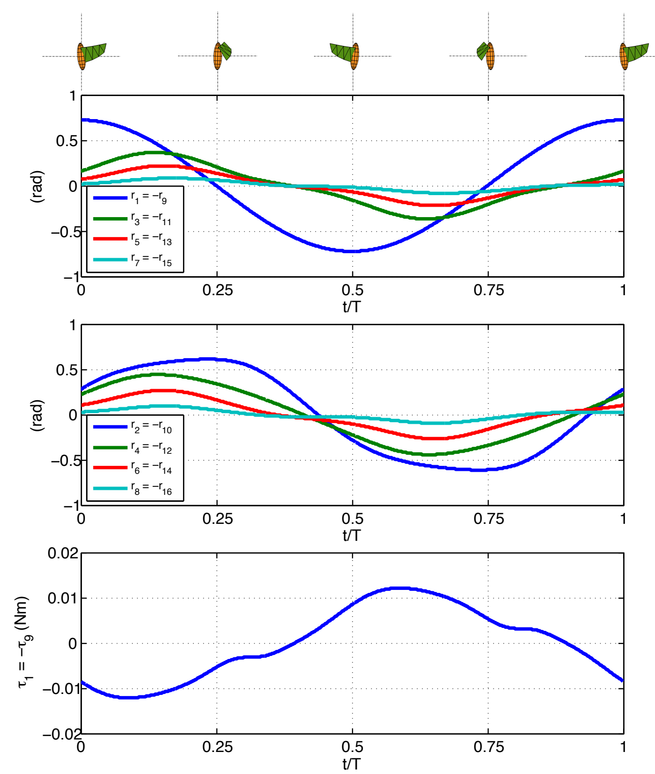

Fig. 8. For one flapping cycle, the time evolution of: on the top, the bending strain; in the middle, the twisting strain; at the bottom, the stroke torque.

degrees in amplitude and a phase lag with respect to the stroke of 58 degrees. For the bending strain, each wing tip rotates at ± 68 degrees in amplitude with a phase lag of 36 degrees. These numerical results are close to the observations from experimental biology [14]. From the view point of the actuation, as illustrated at the bottom in figure 8, the proposed hybrid algorithm allows to compute the torques required to ensure the desired flapping motion. For a flapping cycle, the maximal torque is 12.1 mNm and this peak appears after the stroke reversals. The mean power, during one flapping cycle, is equal to 0.42 W per wing and the total specific mechanical power is closed to 120 W/Kg which is less than the value measured for the hawk moth *Manduca sexta* given in [18].

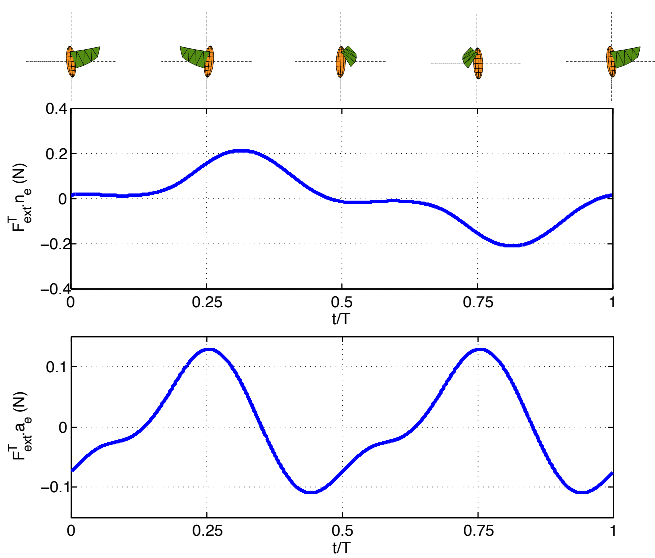

Fig. 9. The time evolution, for one flapping cycle, of the axial (on top) and the vertical (at the bottom) components of the external forces.

Finally, figure 9 shows the linear components of the external forces in the sagittal plane. The vertical component of the external force, image of the lift, is maximal when the wings are aligned with the thorax and minimal when the wings change their stroke direction. These observations are in agreement with the literature.

V. Conclusions

In this paper, we have presented a hybrid algorithm dedicated to the modeling of a Mobile Multibody System with a tree-like structure having both active and passive joints. Based on the Newton-Euler approach of robots dynamics [1], [2], [3], the proposed approach can solve the forward and inverse problems through a unique hybrid algorithm. Moreover, in the context of the locomotion of soft robots bio-inspired from animals, this algorithm allows to compute, through a model of contact forces with the environment: 1) the net motion; 2) the torques produced by the muscles or the actuators; 3) the body shape, i.e. the deformations of soft appendages after their discretisation in to serially connected rigid bodies. As illustrated in section IV-C, in the case of the flapping flight, the given solution is computationally efficient with observations of experimental biology [6], [14], [17]. In particular, we have been able to numerically recover the characteristic wing deformations of the hawk moth *Manduca sexta* during the stroke reversals.

VI. Acknowledgments

The authors would like to thank the French National Agency for Research (ANR) for supporting this work via EVA (Flying Autonomous Entomopter) project (2009-2013).

References

[1] W. Khalil, G. Gallot, and F. Boyer, "Dynamic modeling and simulation of 3-d serial eel-like robot," *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and reviews*, vol. 37, pp. 1259–1268, 2007.

[2] F. Boyer and S. Ali, "Recursive inverse dynamics of multibody systems with joints and wheels," *IEEE Transactions on Robotics*, vol. 27, pp. 215–228, 2011.

[3] M. Porez, F. Boyer, and A. J. Ijspeert, "Improved lighthill fish swimming model for bio-inspired robots - modelling, computational aspects and experimental comparisons." *To appear in International Journal of Robotics Research*, pp. 1–34, 2014.

[4] A. Belkhiri, M. Porez, and F. Boyer, "A hybrid dynamic model of an insect-like mav with soft wings," in *proceeding of IEEE International Conference on Robotics and Biomimetics, (ROBIO'2012)*, December 2012, pp. 108–115.

[5] R. M. Alexander, *Elastic mechanisms in animal movement*. Cambridge, UK: Cambridge University Press, 1988.

[6] M. H. Dickinson, F.-O. Lehmann, and S. P. Sane, "Wing rotation and the aerodynamic basis of insect flight," *Science*, vol. 284, pp. 1954–1960, 1999.

[7] D. Beal, F. Hover, M. Triantafyllou, J. Liao, and G. Lauder, "Passive propulsion in vortex wakes," *Journal of Fluid Mechanics*, vol. 549, pp. 385–402, 2006.

[8] F. Candelier, M. Porez, and F. Boyer, "Note on the swimming of an elongated body in a non-uniform flow," *Journal of Fluid Mechanics*, vol. 716, pp. 616–637, 2013.

[9] R. J. Wood, "The first takeoff of a biologically inspired at-scale robotic insect," *IEEE Transactions on Robotics*, vol. 24, pp. 341–347, 2008.

[10] N. O. Pérez-Arancibia, K. Y. Ma, K. C. Galloway, J. D. Greenberg, and R. J. Wood, "First controlled vertical flight of a biologically inspired microrobot," *Bioinspiration & Biomimetics*, vol. 6, pp. 036 009–1–11, 2011.

[11] R. M. Murray, S. S. Sastry, and L. Zexiang, *A mathematical introduction to robotic manipulation*. Boca Raton, USA: CRC Press, 1994.

[12] J. Luh, M. Walker, and R. Paul, "On-line computational scheme for mechanical manipulators," *ASME Transaction, Journal of Dynamic System, Measurements and control*, vol. 102, pp. 69–76, 1980.

[13] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *International Journal of Robotics Research*, vol. 2, pp. 13–30, 1983.

[14] A. Willmott and C. Ellington, "The mechanics of flight in the hawk-moth manduca sexta. i. kinematics of hovering and forward flight," *Journal of Experimental Biology*, vol. 200, pp. 2705–2722, 1997.

[15] J. A. Walker, "Rotational lift: something different or more of the same?" *Journal of Experimental Biology*, vol. 205, pp. 3783–3792, 2002.

[16] H. Lamb, *Hydrodynamics*, D. Publication, Ed. New York, USA: Dover Publication, 1932.

[17] T. L. Hedrick and T. L. Daniel, "Flight control in the hawkmoth manduca sexta: the inverse problem of hovering," *The Journal of Experimental Biology*, vol. 209, pp. 3114–3130, 2006.

[18] T. M. Casey, "Flight energetics of sphinx moths: power input during hovering flight," *The Journal of Experimental Biology*, vol. 64, pp. 529–543, 1976.