Dynamics of Tree Type Robotic Structures with Application to a Human Hand Prosthesis

André Carvalho and Afzal Suleman[♠]
University of Victoria
Department of Mechanical Engineering
Victoria, B.C., V8W 3P6 CANADA
andrerdc@gmail.com

Abstract

This paper describes the dynamics of an artificial hand dynamic model for application to a human hand prosthesis. The Articulated Body Model (ABM) algorithm was selected to model the dynamics of a tree type robotic structure. The ABM is a numerical Newton-Euler based algorithm that solves the forward dynamics for a joint-link model. The main advantage of this algorithm resides in the analysis of the system link by link rather than the entire system analysis. This feature enables the implementation of a computationally efficient code and makes the algorithm generic enough to be applied to almost any robotic structure, with minimal additional effort. Furthermore, as the basic algorithm only handles serial structures, it needed to be modified to include the effect of the gravity (the weight of the links), loads on the end-effector, elasticity and damping at the joints, and the generalization to tree-type structures.

Introduction

The work described in this article is part of a broader effort in which the principal objective is designing and manufacturing an artificial human hand for prosthetic applications. The scope of this paper includes static analysis and, more importantly, the dynamics behavior of the artificial hand.

The human hand is an extremely complex manipulator, not only because it has a large number of degrees of freedom (DoF), but also because there is elasticity, damping and non-linear saturations at the joints. To further increase the complexity of the hand, the actuators — the muscles — are by nature non-linear, because they are not single active materials, but instead, they are mechanisms [1]. The consequences of these non-linearities are: different responses obtained for contraction/expansion and for active/passive

[♠] Professor and Associate Dean Research, e-mail: Suleman@uvic.ca ; also Principal Investigator, IDMEC-IST, Lisbon, Portugal.

actuation, hysteresis, dead-zones, saturation and non-linear viscosity. The hand is also a special case, since all the actuators are connected by a network of tendons and ligaments unlike normal robotic systems. This means that it is a heavily uncoupled system — one actuator moves several joints simultaneously and one joint may need several actuators to be moved [2].

This article will focus on the first step of the model design — the establishment of a fast numerical algorithm to simulate the dynamic behavior of the hand in open loop for specified torques at the joints.

The Articulated Body Algorithm

The Articulated Body Method (ABM) is a numerical Newton-Euler-based forward dynamics method. The starting point of the ABM and its main principle is that any articulated structure, when approximated by a joint-link model, can be computed link by link instead of the system as whole. Therefore, the complexity and computational cost are reduced [3].

ABM principles

Considering a human hand approximated by a joint-link model as depicted in fig. 1.

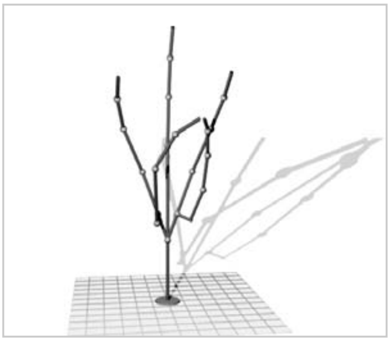


Fig. 0 — Joint-Link model of a human hand.

By selecting a link, the coordinate systems can be established in a way that the (*i*-1)-frame (at the base of the link) follows the link and the *i*-frame (at the top) has an additional DoF about the z-axis, fig. 2.

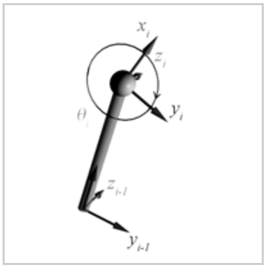


Fig. 0 — Link with its frames.

Since the selected link movement is unknown *a priori*, one must consider a generalized movement state with linear and angular accelerations, linear and angular velocities and, finally, loads and moments at the beginning and at the end of the link, fig. 3(a) and (b).

Using the principles of Newton-Euler Vector Dynamics and taking the base frame as reference, the following expressions can be derived for the linear velocity at the top frame, [4]:

$$\mathbf{v}_i^O = \mathbf{v}_{i-1}^O + \boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1}^L \tag{1}$$

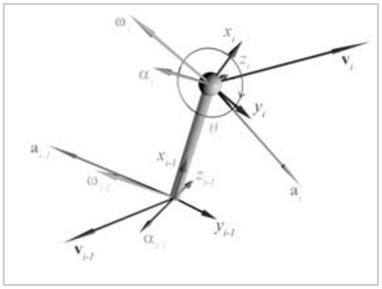


Fig. 0(a) — Link accelerations and velocities.

Angular velocity:

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \dot{\boldsymbol{\theta}}_i \tag{2}$$

Linear acceleration:

$$\mathbf{a}_i^O = \mathbf{a}_{i-1}^O + \boldsymbol{\alpha}_{i-1} \times \mathbf{r}_{i-1}^L + \boldsymbol{\omega}_{i-1} \times \left(\boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1}^L\right) \tag{3}$$

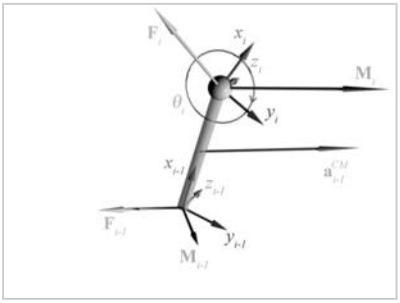


Fig. 3(b) — Link loads and moments.

Angular acceleration:

$$\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_{i-1} + \ddot{\boldsymbol{\theta}}_i + \boldsymbol{\omega}_i \times \dot{\boldsymbol{\theta}}_i \tag{4}$$

Linear acceleration of the Center of Mass (CM):

$$\mathbf{a}_i^{CM} = \mathbf{a}_{i-1}^O + \boldsymbol{\alpha}_{i-1} \times \mathbf{r}_{i-1}^{CM} + \boldsymbol{\omega}_{i-1} \times \left(\boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1}^{CM}\right) \tag{5}$$

Balance of forces:

$$\sum \mathbf{F} = \mathbf{F}_i + \mathbf{F}_{i-1} = m_{i-1}\mathbf{a}_{i-1}^{CM} \tag{6}$$

Balance of moments:

$$\sum \mathbf{M} = \mathbf{M}_{i-1} + \mathbf{M}_i + \mathbf{r}_{i-1}^L \times \mathbf{F}_i = \dot{\mathbf{H}}_{i-1}^O \tag{7}$$

where the superscript is the application point[*], e.g., $\mathbf{a}_i^O$ is the linear acceleration of the link base whereas $\mathbf{a}_i^{CM}$ is the linear acceleration of the CM; $\mathbf{r}_j^L$ and $\mathbf{r}_j^{CM}$ are, respectively, the position vectors of the top frame origin and Center of Mass of a link on its base frame coordinates; $\dot{\mathbf{H}}_i^O$ is the time derivative of the angular momentum about the rotation axis origin.

[*] All vectors are relative to the coordinate system at the base of the link.

In order to simplify (7), a new notation was introduced — Spatial Vectors[†] and Matrices. The spatial vectors are six-dimensional quantities created by assembling the linear and angular parts of a physical quantity. Thus, the Spatial Velocity, Acceleration and Force[‡] are defined by [3]:

$$\hat{\mathbf{v}}_i = \begin{bmatrix} \mathbf{v}_i^O \\ \boldsymbol{\omega}_i \end{bmatrix} \qquad \hat{\mathbf{a}}_i = \begin{bmatrix} \mathbf{a}_i^O \\ \boldsymbol{\alpha}_i \end{bmatrix} \qquad \hat{\mathbf{F}}_i = \begin{bmatrix} \mathbf{F}_i \\ \mathbf{M}_i^O \end{bmatrix}$$

Using the new notation, equations (1) to (4) become:

$$\hat{\mathbf{v}}_i = \begin{bmatrix} \mathbf{R}_{i-1}^i & -\mathbf{R}_{i-1}^i \mathbf{S}_{\mathbf{r}_{i-1}^L}^{i-1} \\ \mathbf{0} & R_{i-1}^i \end{bmatrix} \hat{\mathbf{v}}_{i-1} + \begin{bmatrix} \mathbf{0} \\ \dot{\boldsymbol{\theta}}_i \end{bmatrix} \tag{8}$$

$$\hat{\mathbf{a}}_i = \begin{bmatrix} \mathbf{R}_{i-1}^i & -\mathbf{R}_{i-1}^i \mathbf{S}_{\mathbf{r}_{i-1}^L}^{i-1} \\ \mathbf{0} & R_{i-1}^i \end{bmatrix} \hat{\mathbf{a}}_{i-1} + \begin{bmatrix} \mathbf{0} \\ \ddot{\boldsymbol{\theta}}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}_{i-1} \times \left(\boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1}^L \right) \\ \boldsymbol{\omega}_i \times \dot{\boldsymbol{\theta}}_i \end{bmatrix} \tag{9}$$

Where **R** and **S** are the rotation matrix from the (*i*-1)-frame to the *i*-frame and the skeew-symmetric matrix of the $\mathbf{r}^L$ vector, respectively. To further simplify the equations, two new entities can be introduced: the Spatial Transformation, $\hat{\mathbf{X}}$ [3]:

$$\hat{\mathbf{X}}_{i-1}^i = \begin{bmatrix} \mathbf{R}_{i-1}^i & -\mathbf{R}_{i-1}^i \mathbf{S}_{\mathbf{r}_{i-1}^L}^{i-1} \\ \mathbf{0} & \mathbf{R}_{i-1}^i \end{bmatrix} \tag{10}$$

and the Spatial Axis, $\hat{\boldsymbol{\phi}}$, which assembles the degrees of freedom of a joint into a spatial vector. For example, the spatial axis of a one-DoF rotational joint about the z-axis is:

$$\hat{\boldsymbol{\phi}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

If the joint has more than one DoF, the spatial axis becomes a matrix with 6 rows and a column for each DoF.

With (10) the spatial velocity and acceleration become:

$$\hat{\mathbf{v}}_i = \hat{\mathbf{X}}_{i-1}^i \hat{\mathbf{v}}_{i-1} + \hat{\boldsymbol{\phi}}_i \dot{\theta}_i \tag{11}$$

[†] Note that the introduction of the spatial vector is just a data rearrangement. All operations (e.g. addiction or multiplication by a scalar) remain unaltered, thus retaining the same mathematical properties and, therefore, remaining a Vector Space.
[‡] All spatial quantities are relative to the coordinate system origin at the base of the link.

$$\hat{\mathbf{a}}_i = \hat{\mathbf{X}}_{i-1}^{i}\hat{\mathbf{a}}_{i-1} + \hat{\boldsymbol{\phi}}_i\ddot{\theta}_i + \begin{bmatrix} \boldsymbol{\omega}_{i-1} \times \left(\boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1}^{L}\right) \\ \boldsymbol{\omega}_i \times \boldsymbol{\phi}_i\dot{\theta}_i \end{bmatrix} \tag{12}$$

where $\boldsymbol{\phi}_i$ is a three-dimensional vector defining the rotation axis of the joint.

The last term of the Spatial Acceleration expression is the centripetal acceleration and will be represented by $\hat{\mathbf{C}}$.

The new notation can be also applied to the Balance of Forces and Moments, thus giving:

$$\hat{\mathbf{F}}_i - \begin{bmatrix} \mathbf{R}_{i+1}^{i} & \mathbf{0} \\ \mathbf{S}_{\mathbf{r}_i^L}^{i}\mathbf{R}_{i+1}^{i} & \mathbf{R}_{i+1}^{i} \end{bmatrix} \hat{\mathbf{F}}_{i+1} = \begin{bmatrix} m_i\mathbf{a}_i^{CM} \\ \dot{\mathbf{H}}_i^{O} \end{bmatrix} \tag{13}$$

The algorithm requires that in previous equation the linear acceleration at the CM is replaced by the linear acceleration at the base of the link, $\mathbf{a}_i^{O}$. One advantage of using this acceleration is that, by (12), it can be directly related with the linear acceleration of the previous frame. Introducing this modification along with the angular momentum equation, the spatial force balance becomes:

$$\hat{\mathbf{F}}_i - \left(\hat{\mathbf{X}}_i^{i+1}\right)^T \hat{\mathbf{F}}_{i+1} = \begin{bmatrix} m_i\mathbf{1} & -m_i\mathbf{S}_{\mathbf{r}_i^{CM}}^{i} \\ m_i\mathbf{S}_{\mathbf{r}_i^{CM}}^{i} & \mathbf{I}_i^{O} \end{bmatrix} \hat{\mathbf{a}}_i + \begin{bmatrix} \boldsymbol{\omega}_i \times \left(\boldsymbol{\omega}_i \times \mathbf{r}_i^{CM}\right) \\ \boldsymbol{\omega}_i \times \mathbf{I}_i^{O}\boldsymbol{\omega}_i \end{bmatrix} \tag{14}$$

This form of the balance of forces introduces two new entities: the Spatial Inertia and Spatial Force Bias. The Spatial Inertia, $\hat{\mathbf{I}}$, given by the spatial acceleration pre-multiplying matrix, is an intrinsic constant property of the link and is the spatial counterpart of the mass and body inertia:

$$\hat{\mathbf{I}}_i = \begin{bmatrix} m_i\mathbf{1} & -m_i\mathbf{S}_{\mathbf{r}_i^{CM}}^{i} \\ m_i\mathbf{S}_{\mathbf{r}_i^{CM}}^{i} & \mathbf{I}_i^{O} \end{bmatrix} \tag{15}$$

The Spatial Force Bias is the result of the repositioning of the acceleration and angular momentum from the center of mass to the rotation axis origin:

$$\hat{\boldsymbol{\beta}}_i = \begin{bmatrix} \boldsymbol{\omega}_i \times \left(\boldsymbol{\omega}_i \times \mathbf{r}_i^{CM}\right) \\ \boldsymbol{\omega}_i \times \mathbf{I}_i^{O}\boldsymbol{\omega}_i \end{bmatrix} \tag{16}$$

Articular Inertia and Bias

The Articulated Body Method computes the structure dynamics treating each link as an independent system, but the effect of the moving upper links on the current link have to be known.

These effects are included in the algorithm through the Articulated Inertia and Articulated Force Bias. Both entities are obtained through the mathematical manipulation of the Spatial Balance of Forces. Considering that there is no tip force at the final link of a chain of links, the balance of forces is defined as follows:

$$\hat{\mathbf{F}}_i = \hat{\mathbf{I}}_i \hat{\mathbf{a}}_i + \hat{\boldsymbol{\beta}}_i \tag{17}$$

Since the abovementioned link is the last one, there are no dynamic influences from upper links. Hence, at the tip, the Articular Inertia and Force Bias are equal to the corresponding Spatial Inertia and Force Bias:

$$\hat{\mathbf{F}}_i = \hat{\mathbf{I}}_i^A \hat{\mathbf{a}}_i + \hat{\boldsymbol{\beta}}_i^A \tag{18}$$

Replacing the acceleration with (12), one obtains:

$$\hat{\mathbf{F}}_i = \hat{\mathbf{I}}_i^A \left(\hat{\mathbf{X}}_{i-1}^i \hat{\mathbf{a}}_{i-1} + \hat{\boldsymbol{\phi}}_i \ddot{\theta}_i + \hat{\mathbf{C}}_i \right) + \hat{\boldsymbol{\beta}}_i^A \tag{19}$$

Pre-multiplying both sides of the equation by the transpose of the Spatial Axis, the applied moment from the Spatial Force can be isolated. Solving for the joint angular acceleration:

$$\ddot{\theta}_i = \left(\hat{\boldsymbol{\phi}}_i^T \hat{\mathbf{I}}_i^A \hat{\boldsymbol{\phi}}_i \right)^{-1} \left(\left(\tau_i - \hat{\boldsymbol{\phi}}_i^T \hat{\boldsymbol{\beta}}_i^A \right) - \hat{\boldsymbol{\phi}}_i^T \hat{\mathbf{I}}_i^A \left(\hat{\mathbf{X}}_{i-1}^i \hat{\mathbf{a}}_{i-1} + \hat{\mathbf{C}}_i \right) \right) \tag{20}$$

Even though this expression represents the final objective of the algorithm, one still needs to derive the Articular Inertia and Force Bias. So, replacing the joint acceleration in (19) by (20) [5]:

$$\hat{\mathbf{F}}_i = \mathbf{N}_i \hat{\mathbf{X}}_{i-1}^i \hat{\mathbf{a}}_{i-1} + \mathbf{N}_i \hat{\mathbf{C}}_i + \mathbf{n}_i \mathbf{M}_i^{-1} \tau_i^A + \hat{\boldsymbol{\beta}}_i^A \tag{21}$$

where:

$$\mathbf{N}_i = \hat{\mathbf{I}}_i^A - \mathbf{n}_i \mathbf{M}_i^{-1} \mathbf{n}_i^T \tag{22}$$

$$\mathbf{M}_i = \hat{\boldsymbol{\phi}}_i^T \mathbf{n}_i \tag{23}$$

$$\mathbf{n}_i = \hat{\mathbf{I}}_i^A \hat{\boldsymbol{\phi}}_i \tag{24}$$

$$\tau_i^A = \tau_i - \hat{\boldsymbol{\phi}}_i^T \hat{\boldsymbol{\beta}}_i^A \tag{25}$$

The Spatial Balance of Forces for the next link (*i*-1) is given by (14). Substituting (21) in (14) for *i*-1:

$$\hat{\mathbf{F}}_{i-1} = \left(\hat{\mathbf{I}}_{i-1} + \left(\hat{\mathbf{X}}_{i-1}^{i}\right)^{T} \mathbf{N}_i \hat{\mathbf{X}}_{i-1}^{i}\right)\hat{\mathbf{a}}_{i-1} + \left(\hat{\boldsymbol{\beta}}_{i-1} + \left(\hat{\mathbf{X}}_{i-1}^{i}\right)^{T} \left(\hat{\boldsymbol{\beta}}_i^{A} + \mathbf{N}_i \hat{\mathbf{C}}_i + n_i M_i^{-1} \tau_i^{A}\right)\right) \quad (26)$$

Equation (26) is similar to (18), thus representing a fully connected link as an independent system:

$$\hat{\mathbf{F}}_{i-1} = \hat{\mathbf{I}}_{i-1}^{A} \hat{\mathbf{a}}_{i-1} + \hat{\boldsymbol{\beta}}_{i-1}^{A} \quad (27)$$

where the Spatial Inertia and Force Bias are defined by the following recursive formulas:

$$\hat{\mathbf{I}}_{i-1}^{A} = \hat{\mathbf{I}}_{i-1} + \left(\hat{\mathbf{X}}_{i-1}^{i}\right)^{T} \mathbf{N}_i \hat{\mathbf{X}}_{i-1}^{i} \quad (28)$$

$$\hat{\boldsymbol{\beta}}_{i-1}^{A} = \hat{\boldsymbol{\beta}}_{i-1} + \left(\hat{\mathbf{X}}_{i-1}^{i}\right)^{T} \left(\hat{\boldsymbol{\beta}}_i^{A} + \mathbf{N}_i \hat{\mathbf{C}}_i + \mathbf{n}_i \mathbf{M}_i^{-1} \tau_i^{A}\right) \quad (29)$$

Articulated Body Method for a serial structure

The ABM for a serial structure (chain of links that are connected to each other by a joint) is divided in three parts: the first forward kinematics loop, the backward dynamics loop and the second kinematics loop [5]. Knowing that the base spatial velocity is zero, in the first kinematics loop, the angular velocities, the centripetal accelerations and spatial force biases are computed as follows:

```
for i = 1 to final link do
    $\boldsymbol{\omega}_i = \mathbf{R}_{i-1}^{i} \boldsymbol{\omega}_{i-1} + \dot{\boldsymbol{\theta}}_i$
    $\hat{\mathbf{C}}_i = \begin{bmatrix} \boldsymbol{\omega}_{i-1} \times \left(\boldsymbol{\omega}_{i-1} \times \mathbf{r}_{i-1}^{L}\right) \\ \boldsymbol{\omega}_i \times \hat{\boldsymbol{\phi}}_i \dot{\theta}_i \end{bmatrix}$
    $\hat{\boldsymbol{\beta}}_i = \begin{bmatrix} \boldsymbol{\omega}_i \times \left(\boldsymbol{\omega}_i \times \mathbf{r}_i^{CM}\right) \\ \boldsymbol{\omega}_i \times \mathbf{I}_i^{O} \boldsymbol{\omega}_i \end{bmatrix}$
```

In the backward dynamics one computes the Articular Inertia and Articular Force Bias, knowing that the end-effector Articular Inertia and Force Bias are equal to the spatial Inertia and Spatial Force Bias, respectively:

```
for i = final link to 1 do
    $\mathbf{n}_i = \hat{\mathbf{I}}_i^A \hat{\boldsymbol{\phi}}_i$
    $\mathbf{M}_i = \hat{\boldsymbol{\phi}}_i^T \mathbf{n}_i$
    $\tau_i^A = \tau_i - \hat{\boldsymbol{\phi}}_i^T \hat{\boldsymbol{\beta}}_i^A$
    $\mathbf{N}_i = \hat{\mathbf{I}}_i^A - \mathbf{n}_i \mathbf{M}_i^{-1} \mathbf{n}_i^T$
    $\hat{\mathbf{I}}_{i-1}^A = \hat{\mathbf{I}}_{i-1} + \left(\hat{\mathbf{X}}_{i-1}^i\right)^T \mathbf{N}_i \hat{\mathbf{X}}_{i-1}^i$
    $\hat{\boldsymbol{\beta}}_{i-1}^A = \hat{\boldsymbol{\beta}}_{i-1} + \left(\hat{\mathbf{X}}_{i-1}^i\right)^T \left(\hat{\boldsymbol{\beta}}_i^A + \mathbf{N}_i \hat{\mathbf{C}}_i + \mathbf{n}_i \mathbf{M}_i^{-1} \tau_i^A\right)$
```

The spatial and joint accelerations are computed at the final loop, the second kinematics loop, where the base acceleration is equal to zero:

```
for i = 1 to final link do
    $\hat{\mathbf{a}}_i^{'} = \hat{\mathbf{X}}_{i-1}^i \hat{\mathbf{a}}_{i-1} + \hat{\mathbf{C}}_i$
    $\ddot{\theta}_i = M_i^{-1} \left(\tau_i^A - \mathbf{n}_i \hat{\mathbf{a}}_i^{'}\right)$
    $\hat{\mathbf{a}}_i = \hat{\mathbf{a}}_i^{'} + \hat{\boldsymbol{\phi}}_i \ddot{\theta}_i$
```

Gravity, Tip load and Dynamic Forces

The Articulated Body Algorithm, as presented in the previous section, is suitable for all serial structures but the simulated system is somewhat far from reality, since it does not include the effect of gravity, a load on the end-effector and the presence of dynamic forces on the joints, i.e., elastic joints with damping.

To add gravity to the system, the spatial acceleration of the base, rather than zero, must be the gravitational acceleration:

$$\hat{\mathbf{a}}_0 = \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix}$$

The effect of the tip force can be accounted for by adding it to the end-effector articular force bias equation:

$$\hat{\boldsymbol{\beta}}_{End}^{A} = \hat{\boldsymbol{\beta}}_{End} + \hat{\mathbf{F}}_{Tip} \tag{30}$$

Finally, to include the dynamic forces one must create an additional quantity (dynamic torque):

$$\tau_i^{Dyn} = c_i\dot{\theta}_i + k_i\left(\theta_i - \theta_i^{Initial}\right) \tag{31}$$

where c_i is the damping coefficient, k_i is the spring constant and $\theta_i^{Initial}$ is the resting position of the spring. Finally, this quantity must be subtracted to (25):

$$\tau_i^{A} = \tau_i - \tau_i^{Dyn} - \hat{\boldsymbol{\phi}}_i^{T}\hat{\boldsymbol{\beta}}_i^{A} \tag{32}$$

Articulated Body Method for a tree structure

The difference between serial and tree structures is that the joints in the latter can be connected to multiple links.

When adding this feature, one must change the formulation to properly handle the new configuration. Recalling the spatial balance of forces, (14):

$$\hat{\mathbf{F}}_i - \left(\hat{\mathbf{X}}_i^{i+1}\right)^T \hat{\mathbf{F}}_{i+1} = \hat{\mathbf{I}}_i\hat{\mathbf{a}}_i + \hat{\boldsymbol{\beta}}_i \tag{33}$$

It can be shown that when multiple links are connected to the current link, (33) becomes:

$$\hat{\mathbf{F}}_i - \sum_j\left(\left({}^{j}\hat{\mathbf{X}}_i^{i+1}\right)^T {}^{j}\hat{\mathbf{F}}_{i+1}\right) = \hat{\mathbf{I}}_i\hat{\mathbf{a}}_i + \hat{\boldsymbol{\beta}}_i \tag{34}$$

where the *j* indices represent the *n* upper links directly connected to the current link.

To begin the derivation of the new ABM, one must consider multiple end-effectors connected to a single link. In this case, although being independent, all end-effectors have the same base spatial acceleration. In order to simplify the computation they will be treated as a group, rather than one by one:

$$\sum_j\left({}^{j}\hat{\mathbf{F}}_i\right) = \sum_j\left({}^{j}\hat{\mathbf{I}}_i^{A}\right)\hat{\mathbf{a}}_i + \sum_j\left({}^{j}\hat{\boldsymbol{\beta}}_i^{A}\right) \tag{35}$$

Expanding (35) and solving it in order to get the joint acceleration one obtains:

$$\ddot{\theta}_i = \left(\hat{\boldsymbol{\phi}}_i^T \sum_j \left({}^j\hat{\mathbf{I}}_i^A \right) \hat{\boldsymbol{\phi}}_i \right)^{-1} \left(\tau_i - \hat{\boldsymbol{\phi}}_i^T \sum_j \left({}^j\hat{\mathbf{I}}_i^A \right) \left(\hat{\mathbf{X}}_{i-1}^i \hat{\mathbf{a}}_{i-1} + \hat{\mathbf{C}}_i \right) - \hat{\boldsymbol{\phi}}_i^T \sum_j \left({}^j\hat{\boldsymbol{\beta}}_i^A \right) \right) \tag{36}$$

Replacing (36) back into (35), an expression similar to (21) can be obtained.

$$\begin{aligned} {}^j\hat{\mathbf{F}}_i = {}&{}^j\mathbf{N}_i\, {}^j\hat{\mathbf{X}}_{i-1}^i \hat{\mathbf{a}}_{i-1} + {}^j\mathbf{N}_i \hat{\mathbf{C}}_i + \\ &{}^j\mathbf{n}_i\, {}^j\mathbf{M}_i^{-1}\, {}^j\tau_i^A + {}^j\hat{\boldsymbol{\beta}}_i^A \end{aligned} \tag{37}$$

Where:

$${}^j\mathbf{N}_i = {}^j\hat{\mathbf{I}}_i^A - {}^j\mathbf{n}_i\, {}^j\mathbf{M}_i^{-1}\, {}^j\mathbf{n}_i^T \tag{38}$$

$${}^j\mathbf{M}_i = \hat{\boldsymbol{\phi}}_i^T\, {}^j\mathbf{n}_i \tag{39}$$

$${}^j\mathbf{n}_i = {}^j\hat{\mathbf{I}}_i^A \hat{\boldsymbol{\phi}}_i \tag{40}$$

$${}^j\tau_i^A = \tau_i - \hat{\boldsymbol{\phi}}_i^T\, {}^j\hat{\boldsymbol{\beta}}_i^A \tag{41}$$

Moving to the *i*-1 link, replacing (37) into its spatial force balance and simplifying the result, (34) becomes:

$$\hat{\mathbf{F}}_{i-1} = \left(\hat{\mathbf{I}}_{i-1} + \sum_j \left(\left({}^j\hat{\mathbf{X}}_{i-1}^i \right)^T {}^j\mathbf{N}_i\, {}^j\hat{\mathbf{X}}_{i-1}^i \right) \right) \hat{\mathbf{a}}_{i-1} + \left(\hat{\boldsymbol{\beta}}_{i-1} + \sum_j \left(\left({}^j\hat{\mathbf{X}}_{i-1}^i \right)^T \left({}^j\mathbf{N}_i \hat{\mathbf{C}}_i + {}^j\mathbf{n}_i\, {}^j\mathbf{M}_i^{-1} \tau_i^A + {}^j\hat{\boldsymbol{\beta}}_i^A \right) \right) \right) \tag{42}$$

Again the expressions for the articular inertia and force bias can be taken from the previous equation:

$$\hat{\mathbf{I}}_{i-1}^A = \hat{\mathbf{I}}_{i-1} + \sum_j \left(\left({}^j\hat{\mathbf{X}}_{i-1}^i \right)^T {}^j\mathbf{N}_i\, {}^j\hat{\mathbf{X}}_{i-1}^i \right) \tag{43}$$

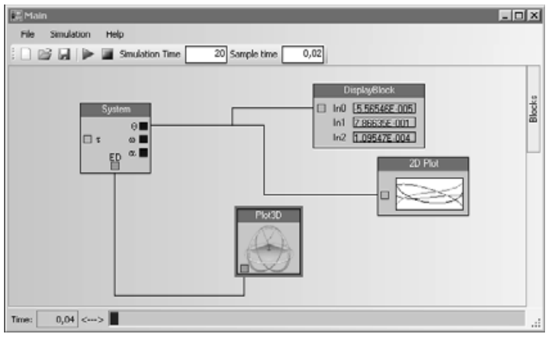
$$\hat{\boldsymbol{\beta}}_{i-1}^A = \hat{\boldsymbol{\beta}}_{i-1} + \sum_j \left(\left({}^j\hat{\mathbf{X}}_{i-1}^i \right)^T \left({}^j\mathbf{N}_i \hat{\mathbf{C}}_i + {}^j\mathbf{n}_i\, {}^j\mathbf{M}_i^{-1} \tau_i^A + {}^j\hat{\boldsymbol{\beta}}_i^A \right) \right) \tag{44}$$

The algorithm and additional forces

The implementation of the algorithm for a tree structure is very similar to the serial one. The only difference is the introduction of an extra loop to do the summations. As for the gravity, tip and dynamic forces, the required adaptations are the same as those made for the serial structure.

The Simulation Software

With the Articulated Body Algorithm defined and adapted to correctly model the system at hand, a computer program was developed. This piece of software can be divided into two parts: the core algorithm and the Graphical User Interface (GUI), Fig. 4. The core algorithm was developed under the Object-Oriented programming paradigm using ANSI C++. The use of ANSI C++ also enables the code to be system independent. The GUI was coded in C# and, thus limited to the Microsoft® Windows Operating System. The graphical interface allows direct user input and outputs the results either graphically (structural 3D plot and joint positions, velocities, accelerations and torques in 2D plots) or numerically.



Simulation Time 20 Sample time 0,02

In0 5.56546E-005
In1 7.86635E-001
In2 1.09547E-004

Time: 0,04 <--->

Fig. 4 — Application screen shot.

The link between the core algorithm in C++ and the GUI in C# is done using a Component Object Model (COM) object that wraps the core algorithm. The COM wrapper code provides the necessary interoperability to C++ code, making its objects useable in any COM supporting computer language, e.g.: C#, Managed and Unmanaged C++, Java and Visual Basic. The setback of the COM wrapper is a slight slowdown of the code, because of the existence of this extra layer.

Results

The developed computer application can handle both serial and tree structures, but since a serial structure is a particular and simpler case of the tree structure, in this paper it will only be presented results for tree structures.

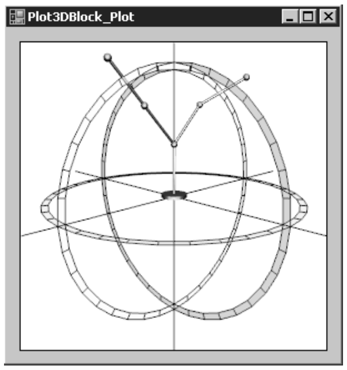


Fig. 5 — Initial position.

The first structure that was analyzed is a “Y” shaped tree with three joints: one at the branches point of union, and one in the middle of each branch. The initial position is the one show on Fig. 5. The test consists on letting the structure fall with its own weight. Note that for simplicity there are no saturations and, therefore, there will be interpenetration. The position vs. time plot is shown on Fig. 6.

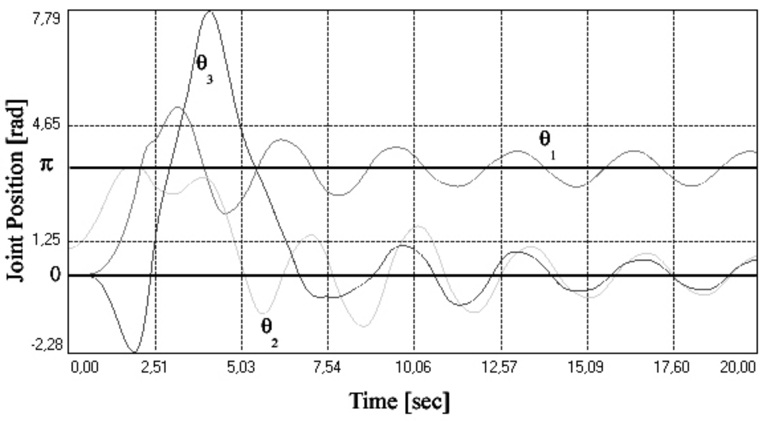


Fig. 6 — Joint Position plot.

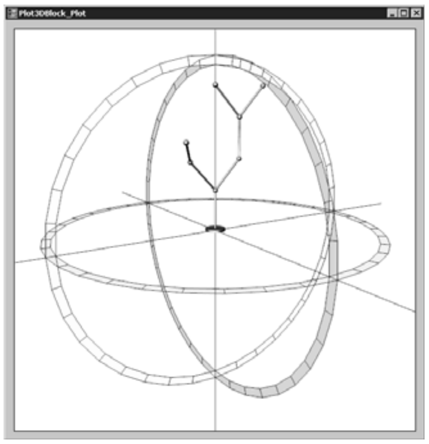


Fig. 7 — Second structure with its initial position.

Changing the previous structure into a more complex one, Fig. 7, with the addition of one extra degree of freedom and using a set of two links as the end-effector of one of the branches, one obtains the results on Fig. 8 by letting the structure fall because of its own weight and the results on Fig. 9 when a torsion spring, with a constant of 100 Nm, is added to the first DoF of longest branch.

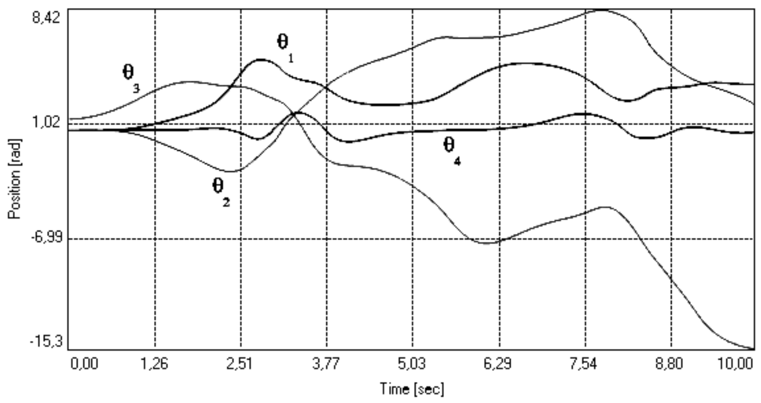


Fig. 8 — First test of the second structure.

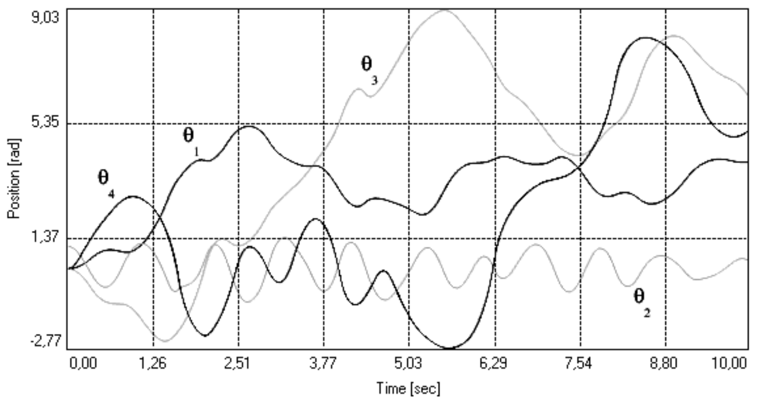


Fig. 9 — Test with a torsion spring.

Conclusions

This paper describes first step of the design of an artificial hand dynamic model. This model will help define the parameters and predict the dynamic behavior of the artificial hand before its construction. Because of the complexity of the system in study, a fast numerical algorithm must be used, which is why the Articulated Body Algorithm was chosen. The main advantage that stems from the usage of this algorithm resides on the analysis of the system link by link, instead of an entire system analysis. This particularity not only enables faster code, but also makes the algorithm generic enough to be applied to almost any robotic structure, needing but small changes.

The goal of this paper was to computationally implement the ABM algorithm for serial and tree robotic structures. Since the basic algorithm only handles serial structures, it needed some modifications so as to correctly describe the hand. These included the effect of the gravity, loads on the end-effectors, elasticity and damping at the joints and, finally, the generalization to tree structures.

The next step will be to expand the algorithm so that it is able to handle closed-loop tree structures, thus enabling the simulation of gripping and gestures requiring fingers to touch each other.

References

[1] Mow, V. C., Huiskes, R., *Basic Orthopaedic Biomechanics and Mechano-Biology*, 3rd ed., Lippincott Williams & Wilkins, pp. 42-50, 2005.

[2] *Gray's Anatomy: The Anatomical Basis of Medicine and Surgery*, 38th ed., Churchill-Livingstone, 1995.

[3] Featherstone, R., *Robot Dynamics Algorithms*, Kluwer Academic Publishers, 1984.

[4] Beer, F.P, Johnston Jr., E.R., *Vector Dynamics for Engineers*, 5th ed., McGraw-Hill, 1988.

[5] McMillan, S., Orin, D. E., McGhee, R. B., *Efficient Dynamic Simulation of an Underwater Vehicle with a Robotic Manipulator*, IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 3, pp. 1194-1206, 1995.