Comparison of Explicit and Implicit Numerical Integrations for a Tendon-Driven Robot

Nicolas J S Testard, Christine Chevallereau, Philippe Wenger

Comparison of explicit and implicit numerical integrations for a tendon-driven robot

Nicolas J.S. Testard, Christine Chevallereau, and Philippe Wenger

Nantes Université, Centrale Nantes, CNRS, LS2N, 44000 Nantes
{nicolas.testard, christine.chevallereau, philippe.wenger}@ls2n.fr

Abstract. A tendon-driven robot inspired from the bird neck is studied in this paper. The objective is to propose numerical integrations for this robot. The classical explicit and implicit Euler integration schemes can be used on this robot. When the elasticity in the tendons is not considered, these schemes are stable and give similar numerical integration results but the explicit implementation is faster. By considering the dynamics of the robot joints on one hand and the dynamics of the motors on the other hand, we propose two Euler integration schemes, explicit and implicit, that take into account the elasticity in the tendons. These schemes rely on the dynamics of the robot joints on the one hand, and on the dynamics of the motors on the other hand and it links them by the tendon elastic model. We observe that with the tendon elasticity model, the explicit integration becomes unstable, depending on the time step, while the implicit one stays stable. The numerical integration that uses this implicit scheme with tendon elasticity allows one to obtain more precise results, while the model of the system is more realistic. Moreover, the evolution of the tendon elongation during a desired motion can be obtained while they are computed in the integration scheme.

Keywords: Dynamics · Numerical integration · Explicit and Implicit Euler · Tendon-driven

1 Introduction

Robots are more and more complex in their design and the numerical integrations of their dynamic model must be more and more efficient and accurate. Two different methods exist for the numerical integration of the dynamic equations, namely, explicit and implicit ([6]). A tendon-driven robot inspired from the bird neck was first proposed in [8]. This paper proposes a study of dynamic numerical integration for this robot and a similar one. First, we study the model with no tendon elasticity and we compare the explicit and implicit integration. We then propose an integration of the dynamic equations in the presence of tendon elasticity. We show how to compute the explicit and implicit Euler integration scheme for both models and what results can be expected for these numerical integrations of our robot models.

As indicated in [10], the stability of the explicit Euler integration can depend on their time step. Furthermore, they indicate that the higher the highest natural

frequency of the structure is, the smaller the time step will have to be. In Section 3, in fact, this integration scheme such that it stays stable can be observed for these robots under study in the absence of tendon elasticity. When tendon elasticity is ignored, the dynamic equation of these robots can also be integrated with an implicit Euler method as shown in [4] and [12]. We will show that with this method, the numerical integration is also always stable.

However, tendon elasticity must be considered to have more realistic results. Softwares like Matlab® ([13]) or Maple® ([11]) exist that model tendon elasticity and use explicit and implicit integration but these softwares are not free. Matlab® proposes several explicit integration schemes like Runge-Kutta methods with different orders or the Heun's method. It also proposes an implicit Euler scheme in the Simulink® graphical programming environment. Furthermore, the tendon model is only present in the Simscape®/Multibody library in Simulink® which is an additional paying feature. Similarly, Maple® proposes several explicit Runge-Kutta integrations and the MapleSim® environment can use the implicit Euler or the Rosenbrock scheme. In contrast, SOFA ([1]) is a free framework that also models tendon elasticity and integrates the dynamic equations with a explicit and implicit scheme. Explicit Euler and Runge-Kutta 4 solvers are used among the explicit schemes and the implicit conjugate-gradient based Euler solver or the Newmark implicit solver are used among the implicit ones. However, this framework can be difficult to handle for those who are not familiar with it and the implicit integration uses more complicated schemes than the implicit Euler. Accordingly, section 4 proposes two integration methods, implicit and explicit, that take into account tendon elasticity. The proposed schemes integrate both the dynamics of the robot joints and the dynamics of the motors that pull the tendons, these two dynamics being linked by the tendon model. It will be observed that the implicit Euler integration scheme is still always stable. At the opposite, the explicit integration becomes unstable if the time step is not small enough. This limit time step for the explicit integration decreases with the complexity of the robot. All the computations presented in this article have been done with Matlab® and can be done with any other programming software.

2 Robot model

The robot is made of a stack of anti-parallelogramm joints or *X-joints* [15]. This joint is actuated through 2 tendons that pull on each side shown in Fig. 1. The tendons are pulled by motors and springs are added in parallel of the tendons to obtain a stable configuration at rest.

Different robots can be designed by arranging several X-joints in series. [8] proposed a robot with N_j= 3 joints and N_t=4 tendons, we also proposed to study a robot with N_j= 4 joints and N_t=5 tendons. One tendon pulls all the joints on the left while the N_t-1 other ones pull the different joints on the right (see Figure 2).

We first present the model with tendon elasticity. Let $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_{N_j}]^\top$ define the vector of joint configurations of the robot. Let $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_{N_t}]^\top$

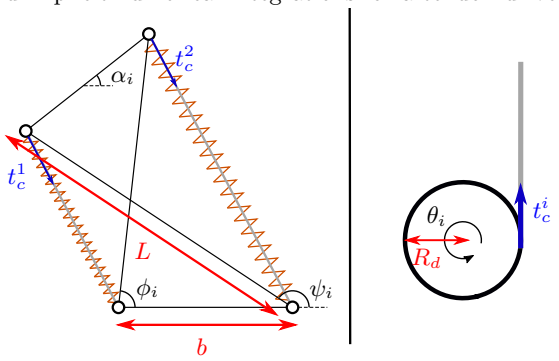

Fig. 1. X-joint (left) and motor pulling a tendon (right)

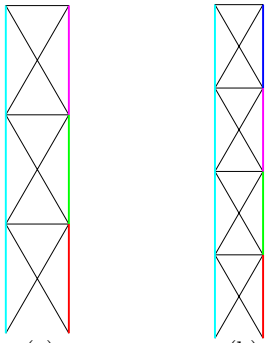

Fig. 2. Tendon-driven robot with 3 joints (a) and with 4 joints (b)

define the vector of motor positions. The dynamic model with tendon elasticity can be written as [3], [7]:

$$\begin{cases} \mathbf{M}_\alpha^s \ddot{\boldsymbol{\alpha}} + \mathbf{c}_s(\dot{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) + \mathbf{g}(\boldsymbol{\alpha}) = \mathbf{Z}\boldsymbol{t}_c - \boldsymbol{\Gamma}_{r\alpha}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) \\ \mathbf{M}_\theta^m \ddot{\boldsymbol{\theta}} = \boldsymbol{\Gamma}_m - \boldsymbol{\Gamma}_{r\theta}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{B}\boldsymbol{t}_c \end{cases} \tag{1}$$

where:

- $\mathbf{M}_\theta^m = diag(I_\theta)$ is the inertia matrix of the motors in the motor space (with I_θ the inertia of the motor around its axis);
- $\boldsymbol{\Gamma}_m$ are the motor torques;
- $\boldsymbol{\Gamma}_{r\theta}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ are the friction in the motors;
- $\mathbf{B} = diag(\frac{R_d}{r_g})$ is the link between the tendon lengths and the motor positions (R_d is the radius of the drum and r_g the gears ratio);
- $\boldsymbol{t}_c$ are the tension in the tendons;
- $\mathbf{M}_\alpha^s$ is the inertia matrix of the structure such that the kinetic energy is $T = \frac{1}{2}\dot{\boldsymbol{\alpha}}^\top \mathbf{M}_\alpha^s \dot{\boldsymbol{\alpha}}$,
- $\mathbf{c}_s(\dot{\boldsymbol{\alpha}}, \boldsymbol{\alpha})$ is the Coriolis effects;
- $\mathbf{g}(\boldsymbol{\alpha})$ corresponds to the forces associated to the potential energy (gravity and springs);

- $\mathbf{Z} = -\mathbf{J}_\theta^\top \mathbf{B}$ (such that $\dot{\boldsymbol{\theta}} = \mathbf{J}_\theta \dot{\boldsymbol{\alpha}}$ with no elasticity) ;
- $\boldsymbol{\Gamma}_{r\alpha}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}})$ is the friction in the structure.

As shown in [14] and [2], there are different tendon models. We will use the one proposed in [9]:

$$\boldsymbol{t}_c = k_c \mathbf{x}_c + d_c \dot{\mathbf{x}}_c \tag{2}$$

where $\mathbf{x}_c$ is the tendon elongation that can be expressed as $\mathbf{x}_c = \mathbf{B}(\boldsymbol{f}_\theta(\boldsymbol{\alpha}) - \boldsymbol{\theta})$, where $\boldsymbol{f}_\theta(\boldsymbol{\alpha})$ is the vector of the motor positions computed from the joint angles when there is no elasticity. k_c defines the link between the tension and the elongation in statics and d_c is a damping coefficient.

In the absence of tendon elasticity, this system of equations can be written as:

$$\mathbf{M}_\alpha \ddot{\boldsymbol{\alpha}} = \boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) + \mathbf{J}_\theta^\top \boldsymbol{\Gamma}_m \tag{3}$$

where:

- $\boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) = -\mathbf{c}(\dot{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) - \mathbf{g}(\boldsymbol{\alpha}) + \mathbf{Z}\left(\mathbf{B}^{-1}\boldsymbol{\Gamma}_{r\theta}\right) - \boldsymbol{\Gamma}_{r\alpha}$
- $\mathbf{M}_\alpha = \mathbf{M}_\alpha^s + \mathbf{J}_\theta^\top \mathbf{M}_\theta^m \mathbf{J}_\theta$
- $\mathbf{c}(\dot{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) = \mathbf{c}_s(\dot{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) + \mathbf{J}_\theta^\top \mathbf{M}_\theta^m \dot{\mathbf{J}}_\theta \dot{\boldsymbol{\alpha}}$

The diagonal and top bars of the X-joints are made with aluminium alloy with dimensions 0.1 m and 0.05 m, respectively. Their mass is 64 g and 32 g, respectively. The radius of the motor drum is R_d= 0.02 m and the gear ratio is r_g= 25. We consider a dry friction in the motors of the form $\boldsymbol{\Gamma}_{r\theta} = \text{diag}\left(\frac{2}{\pi}\arctan\left(c_s\dot{\theta}\right)\right)\boldsymbol{\Gamma}_s$ as proposed in [9] where $\boldsymbol{\Gamma}_s$= 0.008 N.m for all motors and c_s= 0.5 . We also consider a viscous friction in the pivots of the X-joints $\boldsymbol{\Gamma}_{r\alpha,j} = f_v\left(2\left(\frac{\partial\phi_j}{\partial\alpha_j}\right)^2 + 2\left(\frac{\partial\psi_j}{\partial\alpha_j}\right)^2\right)\dot{\boldsymbol{\alpha}}_j$ with f_v = 0.001 N.m/(rad/s). In the tendon model , we take $k_c = 10^5$ N/m and d_c = 100 N/m/(rad/s). Each X-joint is equipped with the same springs on each side. On each side, the spring constants for the 3-modules (resp. 4-modules) robot are [600,600,200] N/m (resp. [800,600,200,200] N/m,), from bottom to top. The free length is 46 mm for all springs.

3 Numerical integrations without elasticity

3.1 Explicit numerical integration without elasticity

The simplest way to integrate the system dynamics is to first compute the joint accelerations at step i with the dynamic equation. The joint velocities are then integrated at step $i+1$ with these accelerations. The joint orientations at step $i+1$ are finally integrated with the velocities at step i. The scheme of this simple explicit numerical integration is represented by the following Eq. (4):

$$\begin{cases} \ddot{\boldsymbol{\alpha}}_i & = \mathbf{M}_\alpha^{-1}\left(\boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i) + \mathbf{J}_\theta^\top(\boldsymbol{\alpha}_i)\boldsymbol{\Gamma}_m\right) \\ \dot{\boldsymbol{\alpha}}_{i+1} & = \dot{\boldsymbol{\alpha}}_i + \ddot{\boldsymbol{\alpha}}_i dt \\ \boldsymbol{\alpha}_{i+1} & = \boldsymbol{\alpha}_i + \dot{\boldsymbol{\alpha}}_i dt \end{cases} \tag{4}$$

Where dt is the time step and $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}(i.dt)$.

3.2 Implicit numerical integration without elasticity

For the implicit integration, the velocities (resp. orientations) at step $i+1$ are computed with the accelerations (resp. velocities) at step $i+1$ and not at step i, as described by Eq. (5):

$$\begin{cases} \mathbf{M}_\alpha \ddot{\boldsymbol{\alpha}}_{i+1} & = \boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}_{i+1}, \dot{\boldsymbol{\alpha}}_{i+1}) + \mathbf{J}_\theta^\top(\boldsymbol{\alpha}_{i+1})\boldsymbol{\Gamma}_m \\ \dot{\boldsymbol{\alpha}}_{i+1} & = \dot{\boldsymbol{\alpha}}_i + \ddot{\boldsymbol{\alpha}}_{i+1} dt \\ \boldsymbol{\alpha}_{i+1} & = \boldsymbol{\alpha}_i + \dot{\boldsymbol{\alpha}}_{i+1} dt \end{cases} \tag{5}$$

However, the accelerations at step $i+1$ also depend on the velocities and orientations at step $i+1$ that we want to compute. Therefore, we linearize the above equation as in [5].

First, we define $d\dot{\boldsymbol{\alpha}}_i = \dot{\boldsymbol{\alpha}}_{i+1} - \dot{\boldsymbol{\alpha}}_i = \ddot{\boldsymbol{\alpha}}_{i+1} dt$ and $d\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i = \dot{\boldsymbol{\alpha}}_{i+1} dt$. The first order linearization gives:

$$\mathbf{M}_\alpha d\dot{\boldsymbol{\alpha}}_i = \boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i)dt + \mathbf{K} d\boldsymbol{\alpha}_i dt + \mathbf{D} d\dot{\boldsymbol{\alpha}}_i dt + \mathbf{J}_\theta^\top(\boldsymbol{\alpha}_i)\boldsymbol{\Gamma}_m dt \tag{6}$$

Where $\mathbf{K} = \dfrac{\partial \boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i)}{\partial \boldsymbol{\alpha}} + \dfrac{\partial \mathbf{J}_\theta^\top(\boldsymbol{\alpha}_i)\boldsymbol{\Gamma}_m}{\partial \boldsymbol{\alpha}}$ and $\mathbf{D} = \dfrac{\partial \boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i)}{\partial \dot{\boldsymbol{\alpha}}}$.
To compute $d\dot{\boldsymbol{\alpha}}_i$ from this equation, we express $d\boldsymbol{\alpha}_i$ as a function of $d\dot{\boldsymbol{\alpha}}_i$ by:

$$d\boldsymbol{\alpha}_i = \dot{\boldsymbol{\alpha}}_{i+1} dt = (\dot{\boldsymbol{\alpha}}_i + d\dot{\boldsymbol{\alpha}}_i)dt \tag{7}$$

We then obtain:

$$\left(\mathbf{M}_\alpha - \mathbf{K}dt^2 - \mathbf{D}dt\right) d\dot{\boldsymbol{\alpha}}_i = \mathit{\Gamma}_{sys}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i)dt + \mathbf{K}\dot{\boldsymbol{\alpha}}_i dt^2 + \mathbf{J}_\theta^\top(\boldsymbol{\alpha}_i)\boldsymbol{\Gamma}_m dt \tag{8}$$

Thus, the implicit integration scheme is defined by:

$$\begin{cases} d\dot{\boldsymbol{\alpha}}_i & = \left(\mathbf{M}_\alpha - \mathbf{K}dt^2 - \mathbf{D}dt\right)^{-1}\left(\boldsymbol{\Gamma}_{sys}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i)dt + \mathbf{K}\dot{\boldsymbol{\alpha}}_i dt^2 + \mathbf{J}_\theta^\top(\boldsymbol{\alpha}_i)\boldsymbol{\Gamma}_m dt\right) \\ \dot{\boldsymbol{\alpha}}_{i+1} & = \dot{\boldsymbol{\alpha}}_i + d\dot{\boldsymbol{\alpha}}_i \\ \boldsymbol{\alpha}_{i+1} & = \boldsymbol{\alpha}_i + \dot{\boldsymbol{\alpha}}_{i+1} dt \end{cases} \tag{9}$$

3.3 Comparison between explicit and implicit numerical integration

For the integration of the equation, the dry friction in the motors will be considered as constant at step i.

For the robots, we can use a computed torque control on the joints as presented in [9]. An example of motion for the robot with 3 and 4 joints is observed in Figure 3 with a time step of dt=2ms. We can observe some difference in the evolution of the angles that are negligible w.r.t. the amplitude of the motion. Moreover, the time step can be increased and decreased and the 2 numerical integrations will remain stable.

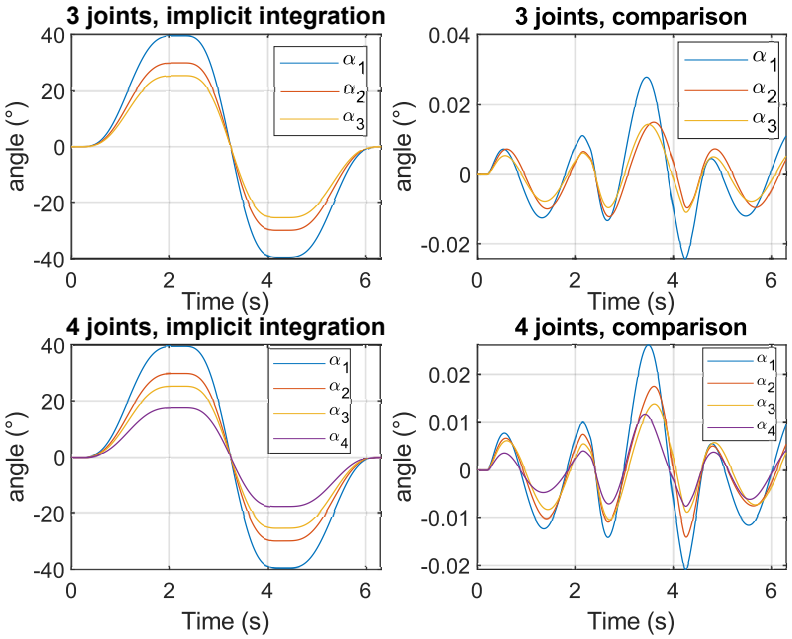

Fig. 3. Numerical integration of the robots (on the left) and comparison between the implicit and explicit numerical integration results (on the right)

	3 joints	4 joints
explicit integration	≈ 4 s	≈ 5 s
implicit integration	≈ 18 s	≈ 30 s

Table 1. Computation time for a motion of 6.3 s with dt=2 ms

Table 1 presents the computation time for these motions. It can be observed that the explicit integration is faster than the implicit one and that the computation time increases with the number of joints.

4 Numerical integrations with tendon elasticity

4.1 Explicit numerical integration with tendon elasticity

We now consider that the tendons are elastic. In this case, the evolution of motor positions is not defined by the joint angles only. Thus, the motor positions need to be integrated as well. Accordingly, the explicit integration scheme of the system with tendon elasticity is defined in the system of equations (10).

$$\begin{cases} \begin{cases} \ddot{\boldsymbol{\theta}}_i &= (\mathbf{M}_\theta^m)^{-1}\left(\boldsymbol{\Gamma}_{sys,\theta}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i, \boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i) + \boldsymbol{\Gamma}_m + \mathbf{B}\boldsymbol{t}_{c,i}\right) \\ \dot{\boldsymbol{\theta}}_{i+1} &= \dot{\boldsymbol{\theta}}_i + \ddot{\boldsymbol{\theta}}_i dt \\ \boldsymbol{\theta}_{i+1} &= \boldsymbol{\theta}_i + \dot{\boldsymbol{\theta}}_i dt \end{cases} \\ \begin{cases} \ddot{\boldsymbol{\alpha}}_i &= (\mathbf{M}_\alpha^s)^{-1}\left(\boldsymbol{\Gamma}_{sys,\alpha}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i, \boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i) + \mathbf{Z}(\boldsymbol{\alpha}_i)(\boldsymbol{t}_{c,i})\right) \\ \dot{\boldsymbol{\alpha}}_{i+1} &= \dot{\boldsymbol{\alpha}}_i + \ddot{\boldsymbol{\alpha}}_i dt \\ \boldsymbol{\alpha}_{i+1} &= \boldsymbol{\alpha}_i + \dot{\boldsymbol{\alpha}}_i dt \end{cases} \\ \begin{cases} \boldsymbol{t}_{c,i} &= k_c \mathbf{x}_{c,i} + d_c \dot{\mathbf{x}}_{c,i} \\ \mathbf{x}_{c,i} &= \mathbf{B}(\boldsymbol{f}_\theta(\boldsymbol{\alpha}_i) - \boldsymbol{\theta}_i) \\ \dot{\mathbf{x}}_{c,i} &= \mathbf{B}(\mathbf{J}_\theta(\boldsymbol{\alpha}_i)\dot{\boldsymbol{\alpha}}_{\boldsymbol{i}} - \dot{\boldsymbol{\theta}}_i) \end{cases} \end{cases} \tag{10}$$

where:

$$\begin{aligned} \boldsymbol{\Gamma}_{sys,\alpha}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) &= -\mathbf{c}_s(\dot{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) - \mathbf{g}(\boldsymbol{\alpha}) - \boldsymbol{\Gamma}_{r\alpha}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) \\ \boldsymbol{\Gamma}_{sys,\theta}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) &= -\boldsymbol{\Gamma}_{r\theta}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{aligned} \tag{11}$$

4.2 Implicit numerical integration with tendon elasticity

Similarly, the implicit integration of the system with tendon elasticity is defined by the system of equations (12).

$$\begin{cases} \begin{cases} \mathbf{M}_\theta^m \ddot{\boldsymbol{\theta}}_{i+1} &= \boldsymbol{\Gamma}_{sys,\theta}(\boldsymbol{\alpha}_{i+1}, \dot{\boldsymbol{\alpha}}_{i+1}, \boldsymbol{\theta}_{i+1}, \dot{\boldsymbol{\theta}}_{i+1}) + \boldsymbol{\Gamma}_m + \mathbf{B}\boldsymbol{t}_{c,i+1} \\ \dot{\boldsymbol{\theta}}_{i+1} &= \dot{\boldsymbol{\theta}}_i + \ddot{\boldsymbol{\theta}}_{i+1} dt \\ \boldsymbol{\theta}_{i+1} &= \boldsymbol{\theta}_i + \dot{\boldsymbol{\theta}}_{i+1} dt \end{cases} \\ \begin{cases} \mathbf{M}_\alpha^s \ddot{\boldsymbol{\alpha}}_{i+1} &= \boldsymbol{\Gamma}_{sys,\alpha}(\boldsymbol{\alpha}_{i+1}, \dot{\boldsymbol{\alpha}}_{i+1}, \boldsymbol{\theta}_{i+1}, \dot{\boldsymbol{\theta}}_{i+1}) + \mathbf{Z}(\boldsymbol{\alpha}_{i+1})(\boldsymbol{t}_{c,i+1}) \\ \dot{\boldsymbol{\alpha}}_{i+1} &= \dot{\boldsymbol{\alpha}}_i + \ddot{\boldsymbol{\alpha}}_{i+1} dt \\ \boldsymbol{\alpha}_{i+1} &= \boldsymbol{\alpha}_i + \dot{\boldsymbol{\alpha}}_{i+1} dt \end{cases} \\ \begin{cases} \boldsymbol{t}_{c,i+1} &= k_c \mathbf{x}_{c,i+1} + d_c \dot{\mathbf{x}}_{c,i+1} \\ \mathbf{x}_{c,i+1} &= \mathbf{B}(\boldsymbol{f}_\theta(\boldsymbol{\alpha}_{i+1}) - \boldsymbol{\theta}_{i+1}) \\ \dot{\mathbf{x}}_{c,i+1} &= \mathbf{B}(\mathbf{J}_\theta(\boldsymbol{\alpha}_{i+1})\dot{\boldsymbol{\alpha}}_{\boldsymbol{i+1}} - \dot{\boldsymbol{\theta}}_{i+1}) \end{cases} \end{cases} \tag{12}$$

By defining:

$$d\dot{\boldsymbol{\theta}}_i = \dot{\boldsymbol{\theta}}_{i+1} - \dot{\boldsymbol{\theta}}_i = \ddot{\boldsymbol{\theta}}_i dt, \quad d\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i+1} - \boldsymbol{\theta}_i = \dot{\boldsymbol{\theta}}_{i+1} dt = (d\dot{\boldsymbol{\theta}}_i + \dot{\boldsymbol{\theta}}_i)dt \tag{13}$$

we can develop at the first order:

$$\begin{aligned} d\mathbf{x}_{c,i} &= \mathbf{x}_{c,i+1} - \mathbf{x}_{c,i} = \mathbf{B}(\mathbf{J}_\theta(\boldsymbol{\alpha}_i)d\boldsymbol{\alpha}_i - d\boldsymbol{\theta}_i) \\ d\dot{\mathbf{x}}_{c,i} &= \dot{\mathbf{x}}_{c,i+1} - \dot{\mathbf{x}}_{c,i} = \mathbf{B}\left(\mathbf{J}_\theta(\boldsymbol{\alpha}_i)d\dot{\boldsymbol{\alpha}}_i + \frac{\partial \mathbf{J}_\theta(\boldsymbol{\alpha}_i)\dot{\boldsymbol{\alpha}}_i}{\partial \boldsymbol{\alpha}} d\boldsymbol{\alpha}_i - d\dot{\boldsymbol{\theta}}_i\right) \end{aligned} \tag{14}$$

Thus, we can obtain:

$$\boldsymbol{t}_{c,i+1} = \boldsymbol{t}_{c,i} + \frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\alpha}} d\boldsymbol{\alpha}_i + \frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\alpha}}} d\dot{\boldsymbol{\alpha}}_i + \frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta}_i + \frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\theta}}} d\dot{\boldsymbol{\theta}}_i \tag{15}$$

where:

$$\begin{aligned} &\frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\alpha}} = k_c \mathbf{B}\mathbf{J}_\theta(\boldsymbol{\alpha}_i) + d_c \mathbf{B}\frac{\partial \mathbf{J}_\theta(\boldsymbol{\alpha}_i)\dot{\boldsymbol{\alpha}}_i}{\partial \boldsymbol{\alpha}}, \quad \frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\alpha}}} = d_c \mathbf{B}\mathbf{J}_\theta(\boldsymbol{\alpha}_i) \\ &\frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\theta}} = -k_c \mathbf{B}, \quad \frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\theta}}} = -d_c \mathbf{B} \end{aligned} \tag{16}$$

Therefore, by applying the same computation as in Section 3.2, the linearization of the equations leads to:

$$\begin{pmatrix} (\mathbf{M}_\theta^m - \mathbf{K}_{\theta\theta}dt^2 - \mathbf{D}_{\theta\theta}dt) & (-\mathbf{K}_{\theta\alpha}dt^2 - \mathbf{D}_{\theta\alpha}dt) \\ (-\mathbf{K}_{\alpha\theta}dt^2 - \mathbf{D}_{\alpha\theta}dt) & (\mathbf{M}_\alpha^s - \mathbf{K}_{\alpha\alpha}dt^2 - \mathbf{D}_{\alpha\alpha}dt) \end{pmatrix} \begin{pmatrix} d\dot{\boldsymbol{\theta}}_i \\ d\dot{\boldsymbol{\alpha}}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Gamma}_\theta \\ \boldsymbol{\Gamma}_\alpha \end{pmatrix} \tag{17}$$

where:

$$\begin{aligned} &\mathbf{K}_{\theta\theta} = \frac{\partial \boldsymbol{\Gamma}_{sys,\theta}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i) + \mathbf{B}\frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\theta}}, \quad \mathbf{D}_{\theta\theta} = \frac{\partial \boldsymbol{\Gamma}_{sys,\theta}}{\partial \dot{\boldsymbol{\theta}}}(\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i) + \mathbf{B}\frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\theta}}} \\ &\mathbf{K}_{\theta\alpha} = \mathbf{B}\frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\alpha}}, \quad \mathbf{D}_{\theta\alpha} = \mathbf{B}\frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\alpha}}} \end{aligned} \tag{18}$$

$$\begin{aligned} &\mathbf{K}_{\alpha\alpha} = \frac{\partial \boldsymbol{\Gamma}_{sys,\alpha}}{\partial \boldsymbol{\alpha}}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i) + \frac{\partial \mathbf{Z}(\boldsymbol{\alpha}_i)\boldsymbol{t}_{c,i}}{\partial \boldsymbol{\alpha}} + \mathbf{Z}(\boldsymbol{\alpha}_i)\frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\alpha}} \\ &\mathbf{D}_{\alpha\alpha} = \frac{\partial \boldsymbol{\Gamma}_{sys,\alpha}}{\partial \dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i) + \mathbf{Z}(\boldsymbol{\alpha}_i)\frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\alpha}}}, \quad \mathbf{K}_{\alpha\theta} = \mathbf{Z}(\boldsymbol{\alpha}_i)\frac{\partial \boldsymbol{t}_c}{\partial \boldsymbol{\theta}} \\ &\mathbf{D}_{\alpha\theta} = \mathbf{Z}(\boldsymbol{\alpha}_i)\frac{\partial \boldsymbol{t}_c}{\partial \dot{\boldsymbol{\theta}}} \end{aligned} \tag{19}$$

$$\begin{aligned} \boldsymbol{\Gamma}_\theta &= \left(\boldsymbol{\Gamma}_{sys,\theta}(\boldsymbol{\theta}_i, \dot{\boldsymbol{\theta}}_i) + \boldsymbol{\Gamma}_m + \mathbf{B}\boldsymbol{t}_{c,i}\right)dt + \left(\mathbf{K}_{\theta\theta}\dot{\boldsymbol{\theta}}_i + \mathbf{K}_{\theta\alpha}\dot{\boldsymbol{\alpha}}_i\right)dt^2 \\ \boldsymbol{\Gamma}_\alpha &= \left(\boldsymbol{\Gamma}_{sys,\alpha}(\boldsymbol{\alpha}_i, \dot{\boldsymbol{\alpha}}_i) + \mathbf{Z}(\boldsymbol{\alpha}_i)\boldsymbol{t}_{c,i}\right)dt + \left(\mathbf{K}_{\alpha\theta}\dot{\boldsymbol{\theta}}_i + \mathbf{K}_{\alpha\alpha}\dot{\boldsymbol{\alpha}}_i\right)dt^2 \end{aligned} \tag{20}$$

Thus, the implementation of the implicit integration taking into account tendon elasticity is:

$$\begin{cases} \begin{pmatrix} d\dot{\boldsymbol{\theta}}_i \\ d\dot{\boldsymbol{\alpha}}_i \end{pmatrix} & = \begin{pmatrix} (\mathbf{M}_\theta^m - \mathbf{K}_{\theta\theta}dt^2 - \mathbf{D}_{\theta\theta}dt) & (-\mathbf{K}_{\theta\alpha}dt^2 - \mathbf{D}_{\theta\alpha}dt) \\ (-\mathbf{K}_{\alpha\theta}dt^2 - \mathbf{D}_{\alpha\theta}dt) & (\mathbf{M}_\alpha^s - \mathbf{K}_{\alpha\alpha}dt^2 - \mathbf{D}_{\alpha\alpha}dt) \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{\Gamma}_\theta \\ \boldsymbol{\Gamma}_\alpha \end{pmatrix} \\ \dot{\boldsymbol{\theta}}_{i+1} & = \dot{\boldsymbol{\theta}}_i + d\dot{\boldsymbol{\theta}}_i \\ \boldsymbol{\theta}_{i+1} & = \boldsymbol{\theta}_i + \dot{\boldsymbol{\theta}}_{i+1}dt \\ \dot{\boldsymbol{\alpha}}_{i+1} & = \dot{\boldsymbol{\alpha}}_i + d\dot{\boldsymbol{\alpha}}_i \\ \boldsymbol{\alpha}_{i+1} & = \boldsymbol{\alpha}_i + \dot{\boldsymbol{\alpha}}_{i+1}dt \\ \mathbf{x}_{c,i+1} & = \mathbf{B}(\boldsymbol{f}_\theta(\boldsymbol{\alpha}_{i+1}) - \boldsymbol{\theta}_{i+1}) \\ \dot{\mathbf{x}}_{c,i+1} & = \mathbf{B}(\mathbf{J}_\theta(\boldsymbol{\alpha}_{i+1})\dot{\boldsymbol{\alpha}}_{i+1} - \dot{\boldsymbol{\theta}}_{i+1}) \\ \boldsymbol{t}_{c,i+1} & = k_c\mathbf{x}_{c,i+1} + d_c\dot{\mathbf{x}}_{c,i+1} \end{cases} \tag{21}$$

4.3 Comparison between the numerical integrations with and without elasticity

We apply the same control on the robots as in the Section 3.3. We obtain motions similar to the one obtained with implicit integration without elasticity for the implicit integration with dt=2 ms (Fig. 3) for the robot with 3 joints and the one with 4 joints.

For the explicit integration, we observe that if the time step dt is greater than a certain value, approximately 0.4 ms, the numerical integration does not converge for the robot with 3 joints. Thus, a condition on the time step for the stability appears when tendon elasticity is taken into account. For the robot with 4 joints, the numerical integration does not converge even with a time step of dt=0.025 ms.

The difference between the numerical integration results with and without elasticity for 3 joints is presented in Fig. 4. We observe that the motion differences are similar between implicit and explicit and that these differences are negligible as compared with the amplitude of the motion. With these numerical integrations, moreover, it is also possible to compute the tendon elongation as presented in the Figure 5.

Table 2 presents the comparison of the computation time for the model with and without elasticity. We observe that the computation time does not increase significantly for the explicit and implicit integration. Table 3 compares the performance of the 2 numerical integrations. It can be observed that when the elasticity of the tendons is taken into account, the implicit integration can becomes necessary for more complex robots. However, these results can change depending on the value of stiffness and friction in the robot.

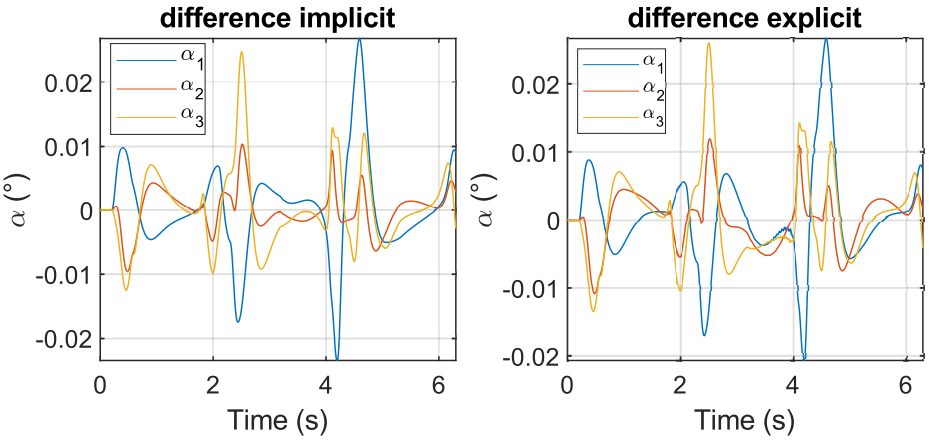

Fig. 4. Comparison of the numerical integration of the models with or without elasticity for the robot with 3 joints for the implicit (dt=2ms) and explicit (dt=0.4ms) methods

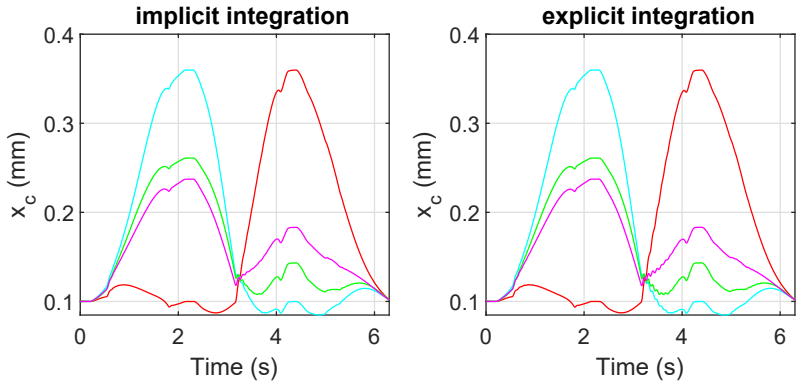

Fig. 5. Elongation of the tendons with the implicit (dt=2ms) and explicit (dt=0.4ms) integration for the robot with 3 joints

	without elastic	with elasticity
explicit integration (dt=0.4 ms)	$\approx$ 17 s	$\approx$ 20 s
implicit integration (dt=2 ms)	$\approx$ 18 s	$\approx$ 22 s

Table 2. Comparison of the computation time between models with or without elasticity for a motion of 6.3 s for 3 joints

5 Conclusion

Numerical integration of tendon-driven robots can be made with explicit or implicit integration. When tendon elasticity is not considered, both approaches are stable for the numerical integration. However, for a given time step, the

		limit time step	computation time for 6.3 s integrated
3 joints	explicit integration	$\approx$ 0.4 ms	$\approx$ 20 s (dt=0.4 ms)
	implicit integration	none	$\approx$ 22 s (dt=2 ms)
4 joints	explicit integration	$<$ 0.025 ms	$>$ 700s
	implicit integration	none	$\approx$ 35 s (dt=2 ms)

Table 3. Comparison of the explicit and implicit integrations with elasticity for a motion of 6.3 s

explicit integration is faster than the implicit integration. Thus, it is better to use explicit integration when there is no elasticity of the tendons. We have proposed an explicit and implicit integration that takes into account tendon elasticity and that integrates the dynamics of the joints and the dynamics of the motors. These numerical integrations give similar results on the robot joints evolution as the numerical integrations without elasticity, while they also allow computing the tendon elongation without increasing the computation time. We have observed that the implicit integration is still always stable with tendon elasticity. However, the explicit integration becomes unstable if the time step is not small enough. This limit time step becomes smaller when the robot becomes more complex. Accordingly, the minimal computation time that can be obtained with the explicit integration becomes much higher than the computation time that can be obtained with the implicit integration. Thus, when tendon elasticity is considered, the implicit integration is better for numerical integrations. In future work, we will take into account external efforts that can be applied on the robot, such as contacts against obstacles.

References

1. Allard, J., Cotin, S., Faure, F., Bensoussan, P.J., Poyer, F., Duriez, C., Delingette, H., Grisoni, L.: SOFA - an Open Source Framework for Medical Simulation
2. Baklouti, S., Courteille, E., Caro, S., Dkhil, M.: Dynamic and Oscillatory Motions of Cable-Driven Parallel Robots Based on a Nonlinear Cable Tension Model. Journal of Mechanisms and Robotics **9**(6) (Oct 2017)
3. Chalon, M., Friedl, W., Reinecke, J., Wimboeck, T., Albu-Schaeffer, A.: Impedance control of a non-linearly coupled tendon driven thumb. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4215–4221 (Sep 2011), iSSN: 2153-0866
4. Coevoet, E., Morales-Bieze, T., Largilliere, F., Zhang, Z., Thieffry, M., Sanz-Lopez, M., Carrez, B., Marchal, D., Goury, O., Dequidt, J., Duriez, C.: Software toolkit for modeling, simulation, and control of soft robots. Advanced Robotics **31**(22), 1208–1224 (Nov 2017)
5. Coevoet, E., Escande, A., Duriez, C.: Optimization-based inverse model of soft robots with contact handling. IEEE Robotics and Automation Letters **2**(3), 1413–1419 (jul 2017)
6. Efimov, D., Polyakov, A., Levant, A., Perruquetti, W.: Discretization of asymptotically stable homogeneous systems by explicit and implicit euler methods. In: 2016 IEEE 55th Conference on Decision and Control (CDC). pp. 5545–5550 (Dec 2016)

7. Fasquelle, B., Furet, M., Chevallereau, C., Wenger, P.: Dynamic modeling and control of a tensegrity manipulator mimicking a bird neck. In: Advances in Mechanism and Machine ScienceProceedings of the 15th IFToMM World Congress on Mechanism and Machine Science, pp. 2087–2097 (2019)
8. Fasquelle, B., Furet, M., Khanna, P., Chablat, D., Chevallereau, C., Wenger, P.: A bio-inspired 3-DOF light-weight manipulator with tensegrity X-joints. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 5054–5060 (May 2020), iSSN: 2577-087X
9. Fasquelle, B., Khanna, P., Chevallereau, C., Chablat, D., Creusot, D., Jolivet, S., Lemoine, P., Wenger, P.: Identification and Control of a 3-X Cable-Driven Manipulator Inspired From the Bird's Neck. Journal of Mechanisms and Robotics **14**(1) (Feb 2022), publisher: American Society of Mechanical Engineers
10. Gui, Y., Wang, J.T., Jin, F., Chen, C., Zhou, M.X.: Development of a family of explicit algorithms for structural dynamics with unconditional stability. Nonlinear Dynamics **77**(4), 1157–1170 (Sep 2014)
11. Hrebicek, J., Rezac, M.: Modelling With Maple And MapleSim. In: ECMS 2008 Proceedings edited by: L. S. Louca, Y. Chrysanthou, Z. Oplatkova, K. Al-Begain. pp. 60–66. ECMS (Jun 2008)
12. Jourdes, F., Valentin, B., Allard, J., Duriez, C., Seeliger, B.: Visual Haptic Feedback for Training of Robotic Suturing. Frontiers in Robotics and AI **9**, 800232 (Feb 2022)
13. Michelin, M., Baradat, C., Nguyen, D.Q., Gouttefarde, M.: Simulation and Control with XDE and Matlab/Simulink of a Cable-Driven Parallel Robot (CoGiRo). In: Pott, A., Bruckmann, T. (eds.) Cable-Driven Parallel Robots. pp. 71–83. Mechanisms and Machine Science, Springer International Publishing, Cham (2015)
14. Ottaviano, E., Castelli, G.: A Study on the Effects of Cable Mass and Elasticity in Cable-Based Parallel Manipulators. In: Parenti Castelli, V., Schiehlen, W. (eds.) ROMANSY 18 Robot Design, Dynamics and Control. pp. 149–156. CISM International Centre for Mechanical Sciences, Springer, Vienna (2010)
15. Snelson, K.D.: Continuous tension, discontinuous compression structures. US Patent 3169611 (Feb 1965)