Multibody System Dynamics **12:** 133–145, 2004.

Control and Virtual Reality Simulation of Tendon Driven Mechanisms

FABIO LONDI, ETTORE PENNESTRÌ, PIER PAOLO VALENTINI, and LEONARDO VITA
Department of Mechanical Engineering, University of Tor Vergata, Rome, Italy;
E-mail: pennestri@mec.uniroma2.it

(Received: 24 July 2003; accepted in revised form: 12 April 2004)

Abstract. In this paper the authors present a control strategy for tendon driven mechanisms. The aim of the control system is to find the correct torques which the motors have to exert to make the end effector describe a specific trajectory. In robotic assemblies this problem is often solved with closed loop algorithm, but here a simpler method, based on a open loop strategy, is developed. The difficulties in the actuation are in keeping the belt tight during all working conditions. So an innovative solution of this problem is presented here. This methodology can be easily applied in real time monitoring or very fast operations. For this reason several virtual reality simulations, developed using codes written in Virtual Reality Markup Language, are also presented. This approach is very efficient because it requires a very low cpu computation time, small size files, and the manipulator can be easily put into different simulated scenarios.

Keywords: tendon driven manipulator, control, virtual reality

1. Introduction

In many mechanical applications there is the need of transmitting the motion using open loop kinematic chains with actuators fixed in space instead of on moving parts (see Figure 2). Because of their topology and low inertia, tendon mechanisms [1–4] offer a solution to the problem. The design of these devices requires the search of optimal working parameters. For example, the increase of preload contributes to the reduction of sliding and backlash between pulleys and belt, but reduces the overall mechanical efficiency. Since flexible belts can transmit tension forces only, the number of transmission lines must be greater than the number of degrees-of-freedom (e.g., in Figure 1 there is a manipulator with 3 degrees-of-freedom and 4 transmission lines). Thus, the control algorithm must handle such redundancy.

The equation of dynamics for a generic n-link manipulator has been deduced by means of a Lagrangian approach [5, 6]. A new procedure, called λ-*corrector* [6], to compute tension of belts has been proposed and compared with the *torque rectifier method* of Jacobsen et al. [7, 8].

The dynamical model, together with inverse kinematics equations, has been included in a control algorithm, that, starting from the desired end-effector

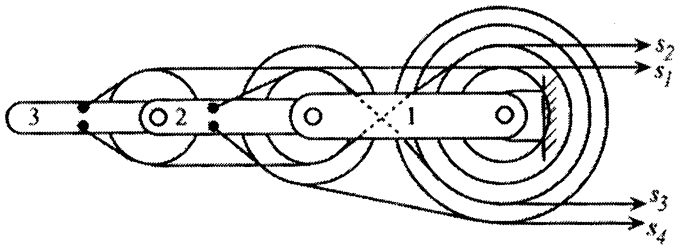

Figure 1. Planar scheme of Stanford/JPL manipulator, a typical 3-links tendon mechanism, with 4 independent transmission chains *s1...s4*.

trajectory, computes the actuating torques which the motors have to exert. The λ-*corrector* procedure has been coded in Maple V and the software developed can solve the inverse dynamics problem for a wide class of tendon driven manipulators.

2. General Considerations

In order to deduce the equations of motion for the entire tendon driven system we can split the mechanism into several subassemblies that are:

- The belt-pulley connections
- Rotor assembly
- Serial manipulator

With this approach, the entire mechanism can be viewed as a serial manipulator constrained by several belt–pulley connections, and the belt actuation is due to some motors which exert the right torques.

Before deducing the equations governing kinematics and dynamics of a generic tendon driven mechanism, let us introduce the following assumptions:

- There is no sliding between belts and pulleys; in particular we assume that each belt is wound several times around the grooves and the Eulerian arc of contact is long enough to avoid sliding.
- The belts are massless.
- Belts' elasticity can be neglected.
- The pulleys are rigid.
- The belts can exert only positive tension.
- The number of the belts (m) is greater than the number of degree-of-freedom (n) of the mechanism.

In all the presented formulas the dotted superscript $\dot{x}$ mean the time derivative of the correspondent variable x.

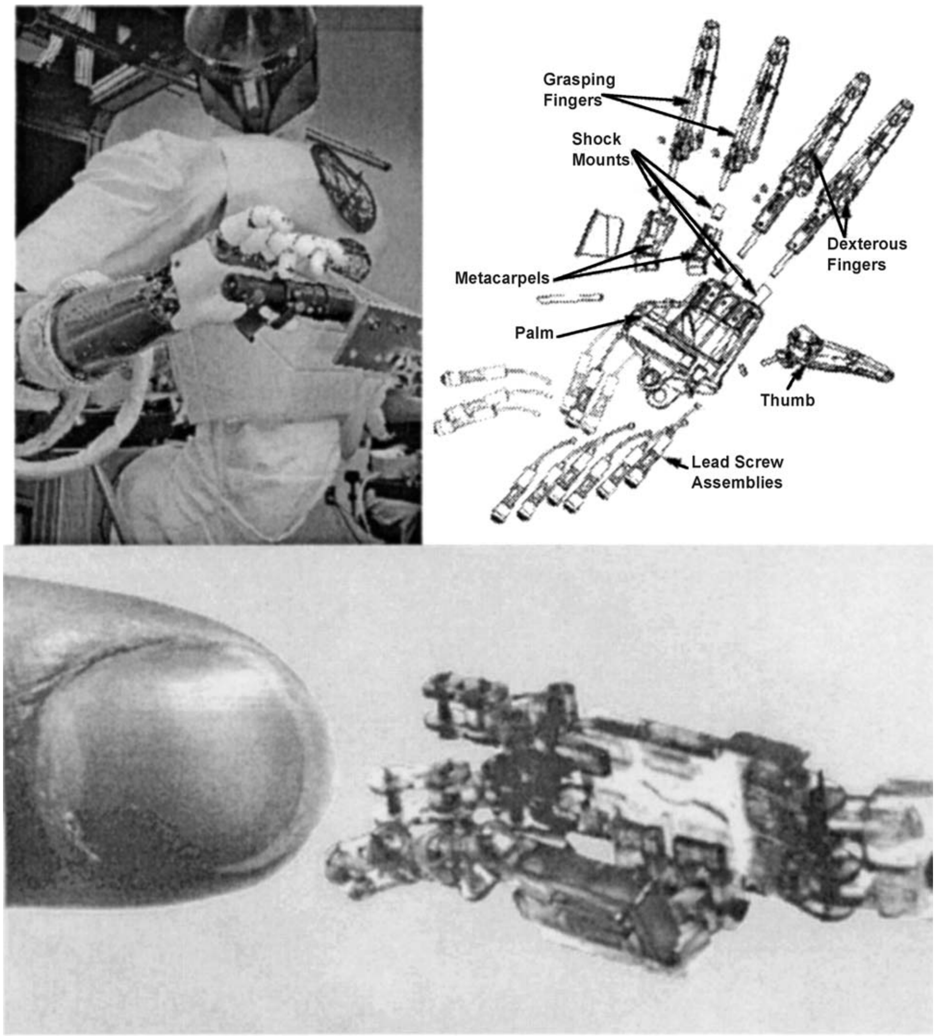

Figure 2. Some examples of modern application of tendon driven mechanisms: the first is the hand of robonaut project developed at NASA's Johnson Space Center [9]; the second is a robotic hand for medical application developed at Berkeley University.

3. Belt–Pulley Kinematics

The kinematics of a tendon driven manipulator is ruled by the belt–pulley connections kinematics equations [5]:

$$
\begin{aligned}
\{s\} &= [A]\{\theta\};\\
\{\dot{s}\} &= [A]\{\dot{\theta}\};\\
\{\ddot{s}\} &= [A]\{\ddot{\theta}\},
\end{aligned}
\tag{1}
$$

where $\{s\}$ is the linear position vector of the belts; $\{\theta\}$ is the angular rotation vector of the pulleys; and [A] is a $m \times n$ (with $m > n$) matrix, whose elements depend on the topology of the mechanisms and on the dimensions of the pulleys [4].

The equations in Equation (1) could be deduced observing that, without sliding, the following condition has to be satisfied:

$$r_j \alpha_{j,i} = \pm r_{j+1} \alpha_{j+1,i}, \tag{2}$$

where $\alpha_{i,j}$ is the angle between the pulley i and the link j. Thus considering the belt winds around one pulley, its displacement is

$$s_h = \pm r_1 \alpha_{1,0}, \tag{3}$$

where $\alpha_{1,0} = \alpha_{1,1} + \theta_1$ as shown in Figure 3. Finally considering the general case of the belt wound around n pulleys, its displacement according to Equations (3) and (2) is

$$s_h = \pm r_1 \theta_1 + \pm r_2 \theta_2 + \cdots + \pm r_n \theta_n. \tag{4}$$

The matrix expression of Equation (4) is represented by the first equation in Equation (1) by means of matrix [A]. It is worth to be said that the elements of this matrix, depending only on the topology and dimensions of the mechanism, in the time derivation of Equation (1), should be kept constant.

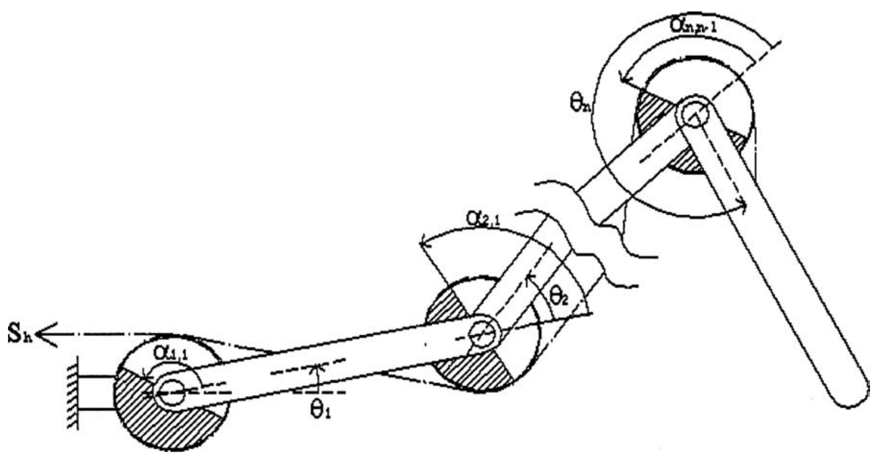

Figure 3. Nomenclature of the transmission chain.

4. Motors Dynamics

The motors' dynamics can be easily investigated using the following expression [2]:

$$[J_m]\{\ddot{\theta}_m\} + [C_m]\{\dot{\theta}_m\} = \{\tau_m\} - [R_m]\{\xi\}, \tag{5}$$

where $\{\theta_m\}$ is the angular position of each rotor of the motors; $[J_m]$ is the diagonal matrix whose elements are the moments of inertia of each rotor; $[C_m]$ is the diagonal matrix whose elements are the damping coefficients of each rotor; $[\tau_m]$ is the vector whose elements are the torques applied to the rotors; $[R_m]$ is the diagonal matrix whose elements are the transmission ratio of gears which connect the motors to the primary pulleys; and $[\xi]$ is the vector of tensions exerted by the belts winding around the rotors. As a consequence the relationship between the displacements of the pulleys and the angular position of the rotors of the motors is defined by the equation

$$\{s\} = [R_m]\,\{\theta_m\}. \tag{6}$$

This one is deduced considering, as shown in Figure 4, the belt winds around just one pulley and there is no sliding:

$$\begin{cases} s = r_p\theta \\ r_m\theta_m = r_f\theta \end{cases} \rightarrow s = \frac{r_m}{r_f} r_p\theta_m. \tag{7}$$

Applying Equation (7) to the general case of m belts wind around n pulleys and rearranging it in a matrix form, leads to Equation (6).

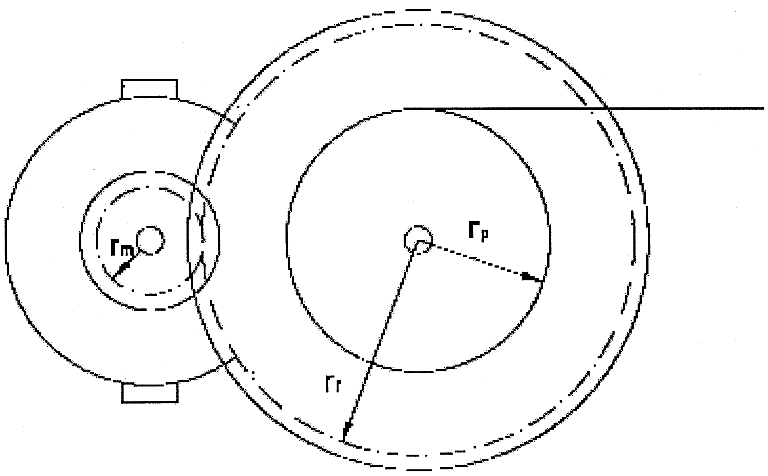

Figure 4. Nomenclature of the motor.

5. Tendon Driven Mechanism Kinematics and Dynamics

The literature presents some works about serial manipulator dynamics [16–20]. These approaches have to be integrated with the belt–pulley and motor subassemblies in order to deduce the equations of motion which are able to describe the entire system. The equations of motion for a tendon driven manipulator can be obtained from a Lagrangian approach as deduced by Londi [6]. After lengthy computation reported in literature [6] we can get:

$$[M(\theta)]\{\ddot{\theta}\} + \{h(\theta, \dot{\theta})\} + \{g(\theta)\} + [C]\{\dot{\theta}\} + [A]^T\{\xi\} = [J]^T\{F\}, \tag{8}$$

where $[M(\theta)]$ is the inertia matrix of the manipulator; $\{h(\theta, \dot{\theta})\}$ is the vector containing the contributions of torques due to centrifugal and Coriolis' accelerations; $\{g(\theta)\}$ is the vector containing the contributions of torques due to gravitational field; $[C]$ is the matrix containing the damping coefficients of the joints; $[J]$ is the Jacobian matrix of the manipulator with respect to the end effector location (considering the constraint equations due to revolute joints); and $\{F\}$ external load vector.

The elements of matrix $[M]$ can be computed using:

$$M_{i,k} = \sum_{j=\max(i,k)}^{n} tr\left(\frac{\partial\left[T_j^0\right]}{\partial\theta_k}\left[\hat{H}_i\right]\frac{\partial\left[T_j^0\right]}{\partial\theta_i}\right) \tag{9}$$

where the pseudo inertia matrix $\left[\hat{H}_i\right]$ can be computed solving:

$$\left[\hat{H}_i\right] = \int_{i-\mathrm{th\,link}} \{r_{p-i}\}\{r_{p-i}\}^T \rho_{i\mathrm{th}}\, dv \tag{10}$$

where ρ_i is mass density of i-th link, $\{r_{p-i}\} = \{x\ y\ z\ 1\}^T$ location vector of generic point on the i-th link.

$[T_j^0]$ is the Denavit–Hartenberg [10, 11] matrix which for two consecutive links of a generic manipulator with revolute joints assumes the expression (see Figure 5 for nomenclature):

$$[T]_i^j = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\cos(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & -\cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

To obtain the Denavit–Hartenberg matrix for two generic links of a manipulator we can perform a simply matrix multiplication:

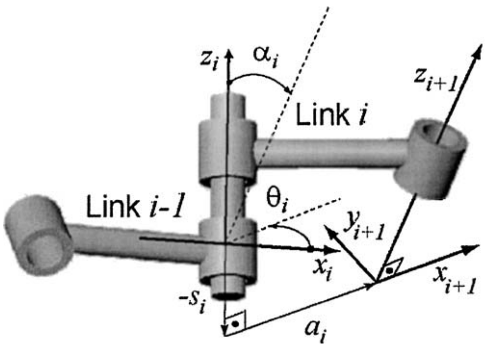

Figure 5. Nomenclature for Denavit–Hartenberg kinematics approach.

$$[T_0^i] = [T_0^1][T_1^2]\dots[T_{i-1}^i] \tag{12}$$

These matrices allow to transform the coordinate of a point from one reference frame to another, using the following formula:

$$\{r_p\}^j = [T]_i^j \{r_p\}^i \tag{13}$$

Assuming only revolute joints, the time derivative $\frac{\partial [T_i^{i-1}]}{\partial \theta_k}$ in Equation (9) has the form

$$\frac{\partial [T_i^{i-1}]}{\partial \theta_l} = [T_i^{i-1}] \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{14}$$

To compute the Jacobian matrix $[J]$ we can differentiate the equations obtained from Equation (12) with respect to angles θ_i.

6. The λ-Corrector Algorithm to Compute Belt Tension

One of the most important parameters which has to be computed in order to solve the control problem is the vector of belts tension. The correct computational strategy has to take into account the physical property that the belt can exert only positive tension.

Several authors in literature [7, 8] proposed for this purpose the *torque rectifier* method. The great disadvantage of this approach is that it cannot be applied for all manipulator, but it depends on kinematics structure of transmission chains, and the study of the topological graph of the manipulator is not sufficient to deduce automatically the equations to compute the unknown tensions.

For this reason, after solving for dynamics equations, these unknown tensions can be found following the method of the *λ-corrector*, proposed by the authors [6].

According to Equation (8), the torque acting on the joints can be computed solving

$$\{\tau_a\} = [J]^T \{F\} - [M(\theta)]\{\ddot{\theta}\} - \{h(\theta, \dot{\theta})\} - \{g(\theta)\} - [C]\{\dot{\theta}\}. \tag{15}$$

We can evaluate the tensions of the belts using the following expression [9]:

$$\{\xi\} = [A]^{+T}\{\tau_a\} + [H]\{\lambda\}, \tag{16}$$

for manipulator with $m = 2n$, otherwise

$$\{\xi\} = [A]^{+T}\{\tau_a\} + \{h\}\lambda \tag{17}$$

for a manipulator with $m = n + 1$.
The matrix $[A]^{+T}$ is the pseudo inverse of the matrix $[A]^T$, [H] is the matrix whose columns are the kernel of matrix $[A]^T$ and the vector $\{\tau_a\}$ can be computed using Equation (15).

For a manipulator with $m = n + 1$, the dimensions of matrix $[H]$ are $m \times 1$ (and so it is simply a vector of length m).

In general, $\{\lambda\}$ is an arbitrary vector of length $m - n$ chosen to satisfy the condition of having only positive tensions of the belts (to describe the physical property already explained). For a manipulator with $m = n + 1$ it is a single value.

According to Londi [6] we can choose the value of $\{\lambda\}$ making equal to zero the higher negative value of the tension obtained solving

$$\lambda_i = \left|\left([A]^{+T}_{2i}\right)\{\tau\}\right| i = 1, 2, \ldots, m \tag{18}$$

for manipulator with $m = 2n$, otherwise

$$\lambda = \min\left(\frac{([A]^{+T})_i\{\tau_a\}}{h_i}\right)\ i = 1, 2, ..., m \tag{19}$$

for a manipulator with $m = n + 1$, where h_i are the elements of $[H]_{\mathrm{m}\times 1}$.

Then, the unknown tensions vector can be found solving the following expression:

$$\{\xi\} = [A]^{+T}([J]^T\{F\} - ([M]\{\ddot{\theta}\} + \{g\} + \{h\} + [C]\{\dot{\theta}\})) + [H]\{\lambda\} \tag{20}$$

The whole procedure to compute belt tension can be summarized in the simple three step algorithm shown in Figure 6.

The comparison between the values of belts tensions obtained by means of the λ-*corrector* and the *torque rectifier* method is shown in Figure 7.

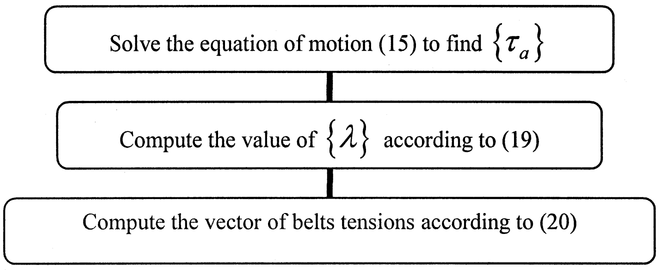

Figure 6. λ-*corrector* computational scheme.

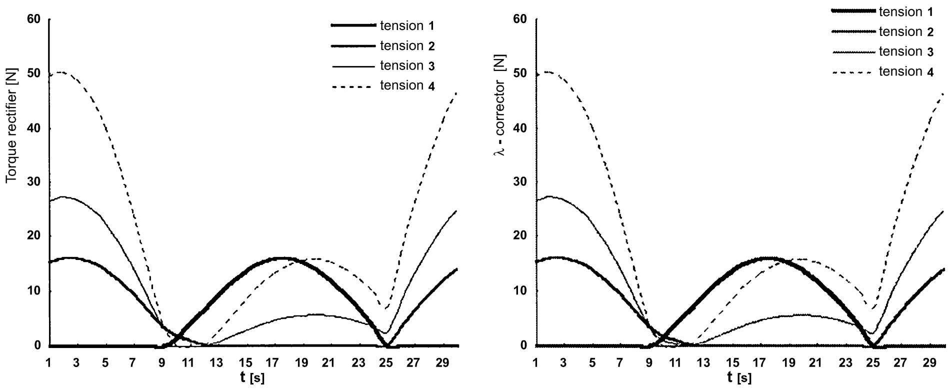

Figure 7. Comparison between λ-corrector and the torque rectifier method.

7. Virtual Reality Simulation

In order to visualise the whole behaviour of the tendon driven mechanisms several simulations using Virtual Reality Mark-up Language (VRML) have been performed [15].

This kind of approach is useful to introduce the mechanism in a virtual environment which can simulate the real working space. In literature there are other similar approaches in different fields of research [12–14].

VRML is a format to describe and transmit 3D objects and worlds composed of geometry and multimedia in a network environment. VRML targets web applications such as computer-aided design, engineering and science visualization, multimedia products, entertainments and educational offerings and shared virtual worlds.

In our models the geometry of the manipulator's links and the geometry of the surrounding world are included in different files. Both geometries can be also imported from a commercial CAD program. Moreover the manipulator has been described with an object oriented programming technique. For each degree-of-freedom has been defined an independent structure of animation (*nodes*), so that they

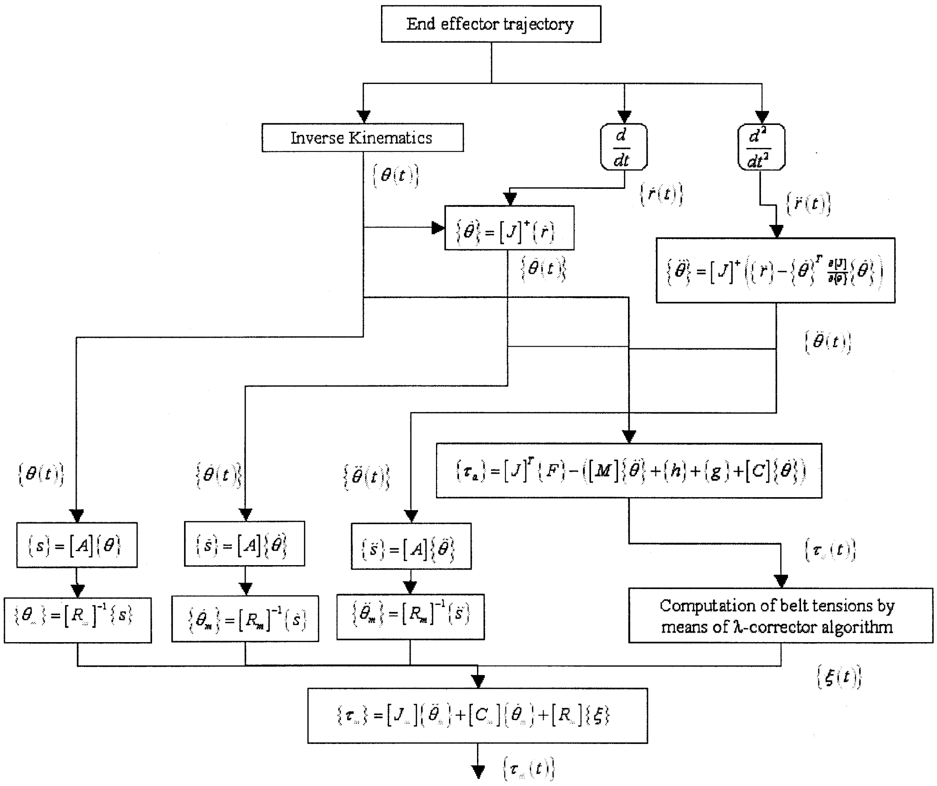

Figure 8. Flow chart of the proposed control algorithm.

can be animated with different time history (*events*). Every structure is constrained to another with a parent–child relation. It means that if a parent moves, the child follows it, but if a child moves, the parent does not.

The authors have developed two different kinds of simulations. The VRML can be successfully used to visualise the results coming from a dynamic analysis importing the externally computed time history for every link. The second application of VRML is to simulate the motion of the manipulator interactively (see Figure 10). In this case the user can drive the manipulator using special virtual sensors. The interactive prescribed movement of these sensors is translated into manipulator motion using Denavit–Hartenberg matrices in Equation (12). In Figure 9, some animation screenshots are presented.

The VRML code can be easily visualized with a common internet browser[1] (such as Internet Explorer) and this assures that the analysis can be viewed on many computer without installing any specific software.

Moreover during animation the user can interactively change the point of view (as a virtual camera) to explore the movement and the interaction of the manipulator

[1] It is only the need for installing a plug in which can be freely downloaded at Blaxxun web site: http://www.blaxxun.com/services/support/download/index.shtml.

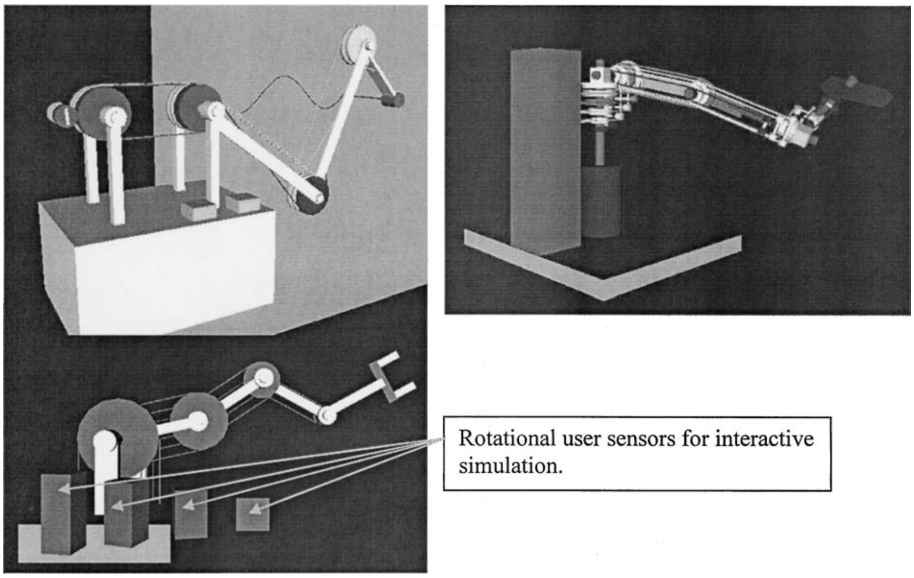

Figure 9. Screenshots from animation in VRML: two 3 degrees-of-freedom in plane manipulators (on the left) and a 6 degree of freedom spatial manipulator (on the right).

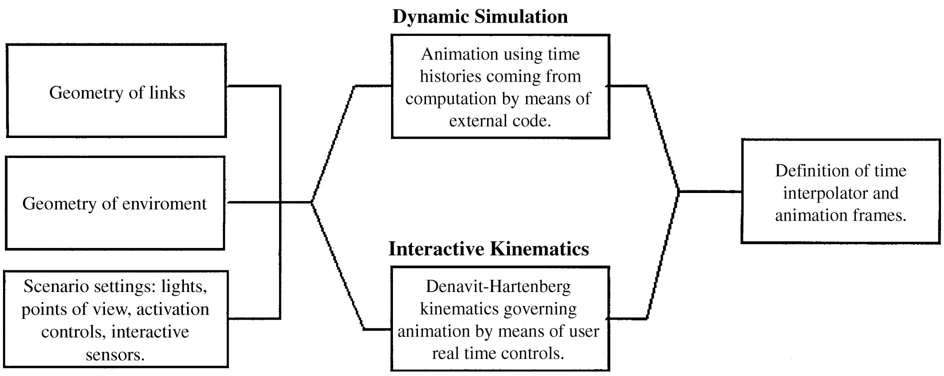

Figure 10. VRML code scheme.

with surrounding environment to check possible incompatibilities with it. The generated file dimensions are very small, so it is very easy to exchange this file from one computer to another.

We can summarize the advantages of VRML approach to simulation

- This approach is completely general and can successfully be applied to different classes of manipulators.
- It is easy to manage real time animation.
- The animation can be easily exchanged between different pc and it does not require any specific software.

- There is a good interaction between the virtual world and the user.
- It is easy to link virtual environments directly generated from CAD application.
- The files require low hardware resources to run.
- The render quality is acceptable.

The disadvantages are

- Render quality is lower than those produced by dedicated computer graphics software (Maya, 3D Studio Max, Lightwave 3D, etc.), because VRML is not able to use hardware or software acceleration.
- Results plotting cannot be viewed or analysed, as it happens in multibody simulation software (Working Model, ADAMS)
- Computational time increases if special textures are included in the model. These textures have to be loaded into the user browser.
- User interaction is limited by the usage of keyboard and mouse commands.

8. Conclusions

The authors in this paper have studied in depth the kinematics and dynamics of tendon driven manipulators in order to set up a control strategy based on an open loop algorithm (see Figure 8). This approach leads to a simpler law of control, but it can easily be used in real time monitoring. The equations of dynamics, in particular as regards the computation of belts tension, make use of λ-*corrector* algorithm, an innovative solution proposed by the authors themselves. Moreover, to perform a virtual simulation of the entire manipulator system (links, belts, pulley and motors) a methodology to generate VRML codes has been also described. This approach has shown a good efficiency as regard many aspects, and so it can be a good way to visualize the mechanism and its interaction with a virtual word. The VRML has also been used to perform real time analysis in a virtual reality environment, and also in this case it has shown great capabilities.

References

1. Lee, J.J., Tendon-driven manipulators, analysis, synthesis and control. Doctoral Dissertation, University of Maryland, Maryland 1991.
2. Tsai, L.W., 'Kinematics analysis of tendon driven mechanism using graph theory', *ASME Journal of Mechanisms, Transmissions and Automation in Design* **111**, 1989, 59–65.
3. Lee, J.J. and Tsai, L.W., 'Torque revolver design for tendon driven manipulators', *ASME Journal of Mechanical Design* **115**, 1993, 877–883.
4. Lee, J.J. and Tsai, L.W., 'On structural synthesis of tendon-driven manipulators having a pseudo-triangular matrix', *International Journal of Robotics Research* **10**(3), 1991, 225–262.
5. Pennestrì E., *Dinamica Tecnica e Computazionale*, Vol. 2, Casa Editrice Ambrosiana, 2002 [in Italian].
6. Londi, F., *Cinematica e Dinamica dei Meccanismi a Puleggia*. Degree Thesis in Mechanical Engineering, Univerisity of Rome Tor Vergata, 2001/2002 [in Italian].

7. Knutti, D.F, Jacobsen, S.C, Wood, J.E. and Biggers, K.B., 'The utah-mit dexterous hand: Work in progress'. Technical Report, *International Journal of Robotics Research* **3**(4), 1984.
8. Iversen, E.K., Jacobsen, S.C., Ko, H. and Davis, C.C., 'Antagonistic control of a tendon driven manipulator', Technical Report in: *Proceeding of IEEE International Conference on Robotics and Automation,* 1989, pp. 1334–1339.
9. The Robonaut Project, official site: http://vesuvius.jsc.nasa.gov/er_er/html/robonaut/Robonaut1.htm.
10. Denavit, J. and Hartenberg, R.S., 'A kinematic notation for lower pair mechanism based on matrices', *ASME Journal of Applied Mechanics* **77**, 1995, 215–221.
11. Denavit, J. and Hartenberg, R.S., *Kinematic Synthesis of Linkages*, McGraw-Hill Book Co., New York, 1964.
12. Agustinov, N., 'VRML as means of expressive 4D illustration in CAM education', *Future Generation Computer System* **17**, 2000, 39–48.
13. Zinzani, C., 'Virtual computer project "Delivering Content in a VRML World', *Future Generation Computer System*. **7**, 2000, 27–37.
14. Honjo, T. and Lim, E.M., *Visualization of Landscape by VRML Systems, Landscape and Urban Planing*, Vol. 55, 2001, pp. 175–183.
15. VRML (Virtual Reality Modelling Language), ISO/IEC DIS 14772, 1997, Online Paper, http://vag.vrml.org/VRML97/DIS/.
16. Salisbury, J.K., Kinematics and force analysis of articulated hands. Doctoral Dissertation, University of Stanford, California, 1982.
17. Salisbury, J.K. and Craig, J.J., 'Articulated hands: Force control and kinematic issues', *The International Journal of Robotics Research* **1**(1), 1982, 4–17.
18. Salisbury, J.K. and Roth, B., 'Kinematic and force analysis of articulated mechanical hands', *ASME Journal of Mechanism, Transmission and Automation in Design* **105**(1), 1983, 35–41.
19. Mason, M.T. and Salisbury, J.K., *Robot Hands and the Mechanics of Manipulations*, MIT Press, Cambridge, MA, 1985.
20. Tsai, L.W., *Robot Analysis – The mechanics of Serial and Parallel Manipulators*, John Wiley & Sons, 1999.