



# Dynamic Splines for interactive simulation of elastic beams in Augmented Reality

P.P. Valentini<sup>(a)</sup>, E. Pezzuti<sup>(a)</sup>

<sup>(a)</sup> University of Rome "Tor Vergata", Department of Mechanical Engineering - Rome, Italy

## Article Information

### Keywords:

dynamic splines  
augmented reality  
interactive simulation  
real time simulation

### Corresponding author:

Pier Paolo Valentini  
Tel.: +390672597137  
Fax.: +39062021351  
e-mail: valentini@ing.uniroma2.it  
Address: University of Rome "Tor Vergata", Department of Mechanical Engineering - via del Politecnico 1, 0013 - Rome, Italy

## Abstract

### Purpose:

The purpose of the paper is to discuss a methodology for addressing the real-time accurate dynamics simulation of elastic beams using an interactive augmented reality environment.

### Method:

The equations of motion of the elastic beams have been deduced using the dynamic spline theory. According to this approach, the degrees of freedom of the beam are described using control points coordinates. The model has been integrated in an augmented reality environment where the user can interact with the elastic element by using a mechatronic tracker.

### Result:

The dynamic spline model has been successfully integrated in the augmented reality environment. This methodology allows to perform accurate interactive real-time simulations and to deal with large displacements as well.

### Discussion & Conclusion:

The implementation of dynamic spline theory reveals to be a valid formulation for simulation of the flexibility of one-dimension elements like beams. Due to the flexibility of the mathematical formulation, a small set of degrees of freedom is sufficient to simulate a very variable shape of the beam in order to take into account large displacement effects with accuracy. This feature is important for the integration in the augmented reality environment where the simulation has to be computed in real time.

## 1 Introduction

During the last decades computer-aided engineering (CAE) methodologies have deeply changed the way of designing and developing products, systems and services. Thank also to significant hardware and software improvements, CAE techniques are widely used by the designers since the early conceptual phases up to the final stages of engineering processes.

In particular, during the last years, many investigations focused on the implementation of interactive design methodologies based on the use of Augmented Reality (AR) [1-2] to support engineering activities. The scientific literature reports some contributions on the developing of interactive tools for geometrical modeling [3-4], assembling simulation [5-8] and behavioural analysis [9-11].

The augmented reality deals with the combination of real world images and computer generated data. Most AR research is concerned with the use of live video imagery which is digitally processed and augmented by the addition of computer generated graphics. The purpose is to extend the visual perception of the world, being supported by additional information and virtual objects.

One of the most important feature of AR is the possibility to ensure an high level of user's interaction with the augmented scene.

One of the current research topics in the field of Augmented Reality for engineering purposes is focused

on the development and review of interactive physical simulation methodologies able to reproduce the actual behaviour of the modelled objects. Interaction is concerned with both the definition of boundary conditions and initial parameters and the real-time control of the simulation. Two are the main requirements for integrating a physical simulation into an augmented reality system: accuracy and synchronization. The first one implies that the computation has to produce correct results, suitable for engineering purposes and subsequent processing. On the other hand, the synchronization requires a real-time interpretation of the user's intent and a real-time solution of the mathematical equations governing the simulation.

In a recent paper [11], Valentini et al. developed a methodology for implementing, solving and reviewing multibody simulations using augmented reality. Their idea was to use patterned markers in order to figure out the communication between the user and the augmented scene. By this way, the user can interact with the objects in the scene pushing, moving and setting boundary condition values. The implementation proposed by the authors concerned with the motion simulation of a collection of rigid bodies.

In many practical cases the simulation has to include the flexibility of some specific bodies. The use of flexible bodies in multibody simulations is necessary when the effects of deformations cannot be neglected for an accurate assessment of the dynamics of the system [12]. The most used type of flexible entity is the beam suitable

for addressing problems involving slender structures, cables, frames, etc.

A beam is a dynamic system with mass and elasticity continuously distributed. According to continuum mechanics, it has an infinite number of degrees-of-freedom. However, they are usually reduced through modelling strategies that aim to describe displacements and deformations using a finite set of variables. The choice of this set is a compromise between the accuracy of taking into account the flexibility and the simplification of the equations to be solved. Scientific literature reports different approaches and contributions in modeling of beams to be included into dynamics simulations [13]. The most of them are not suitable for a real-time processing because involves too many equations of motion and their computation is often onerous. Other methods, based on several important simplifications, are more suitable for real-time processing but achieve approximated results.

Many scientific contributions dealt with the implementation of beam formulations suitable for static deformation study of beams. Classical cable harnessing problems can be addressed using these methodologies [14-16].

In 1996, Quin and Terzopoulos introduced the basis of a mathematical formulation of the D-NURBS for the computer-aided design of NURBS entities [17].

The D-NURBS combine physics-based constraining equations with spline geometry representation. Their application has been focused on car and airplane design, virtual prototyping and assembly simulations [18-20]. Their main advantage is that even a complex and large displacement of a 1D element can be expressed in terms of a polynomial closed form expression (using spline representation), deducing geometrically exact expressions of kinetic and elastic energy.

In the 2011, Valentini et al. [21] proposed and tested a methodology based on the use of dynamic spline for addressing stand-alone multibody simulations with flexible beams undergoing to large displacements, including general constraint and force formulations.

Starting from this background, the purpose of this investigation is to develop a methodology for integrating the dynamic spline formulation proposed in [21] into an augmented reality environment in order to perform interactive simulations including flexible beams.

The original contributions of the works can be summarized in three different aspects:

- a numerical methodology for integrating the dynamic spline formulation presented in [21] into an augmented reality system, taking into account the real-time requirements of the simulation;
- a methodology for improving the precise interaction between the user and the simulated scene based on the use of a multi d.o.f. mechatronic arm and a specific dynamic implementation.
- a preliminary validation of the entire methodology based on the comparison with more standard structural analysis tools.

## 2 Theory of the dynamic splines

A spline is a piecewise polynomial description of a curve in parametric form. Considering a set of  $m$  control points  $\{P_0 \dots P_{m-1}\}$ , a spline can be expressed by means of the general formula [16]:

$$p(u) = \sum_{i=0}^{m-1} b_i(u) P_i \quad (1)$$

where  $b_i(u)$  are the blending functions (polynomial functions of the variable  $u$ ) fitting the control points.

In general, the blending functions depend on the degree of the polynomials, the numbers of control points and the knots sequence [22]. The control points can be considered as the degree of freedom of the spline. In the Bézier-like formulation, except for the first and the last control point, the curve doesn't pass through them, but their position in space influences the shape of the spline.

Before discussing the details about the theory of the dynamic splines, it is useful to review some key geometrical concepts about their geometrical representation. In particular, in order to simplify the computation of local geometrical properties of a curve (e.g. tangent vector, curvature, etc.) it is useful to introduce the Frenet frame, which is a local frame moving along the curve. Assuming that the curve is given in the algebraic form in (1), the versors associated with the Frenet coordinate system can be expressed as follows:

$$t(u) = \frac{p'(u)}{\|p'(u)\|} \text{ is the tangent unit vector} \quad (2)$$

$$m(u) = b(u) \wedge t(u) \text{ is the normal unit vector} \quad (3)$$

$$b(u) = \frac{p'(u) \wedge p''(u)}{\|p'(u) \wedge p''(u)\|} \text{ is the binormal unit vector} \quad (4)$$

where:

$$p'(u) = \frac{\partial p(u)}{\partial u} \text{ and } p''(u) = \frac{\partial^2 p(u)}{\partial u^2}.$$

Considering the changing of the Frenet frame along the curve, it is possible to define two scalar parameters: the Frenet curvature  $\kappa(u)$  and the Frenet torsion  $\tau(u)$ , defined as:

$$\kappa(u) = \frac{\|p'(u) \times p''(u)\|}{\|p'(u)\|^3} \quad (5)$$

$$\tau(u) = \frac{(p'(u) \times p''(u)) \cdot p'''(u)}{\|p'(u) \times p''(u)\|^2} \quad (6)$$

Consider now a physical beam with a cross section  $S$ . The neutral fiber or neutral axis is the oriented curve of length  $L$  that passes through the center of every cross-section. The neutral fiber can be geometrically expressed with a spline.

According to (1), the shape and attitude of this spline is regulated by the location of its control points. Thus, they can be considered as the degrees-of-freedom of the neutral fiber of the beam.

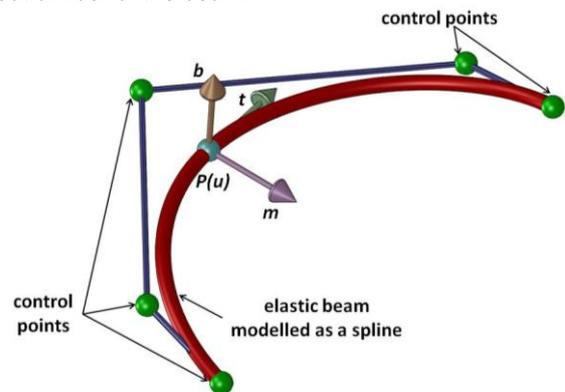


Fig. 1 Spline nomenclature

When the control points are displaced, the spline (and so the neutral fiber) moves and changes its shape

accordingly. For a geometrical 3D curve with  $m$  control points we have  $3 \cdot m$  degrees of freedom (3 translations). Since the real beams have a thickness, it is also necessary to include another degree of freedom which describe the roll of the cross section that represents the rotation of the cross section about the neutral fiber [21]. It can be introduced as the forth degree of freedom of the generic control point. Following this approach, the global degrees of freedom of the  $i$ th control point can be expressed using a four-parameter vector:

$$\begin{aligned} P_i &= \{q_{4i-3} \quad q_{4i-2} \quad q_{4i-1} \quad q_{4i}\}^T = \\ &= \{x_i \quad y_i \quad z_i \quad \theta_i\}^T = \{r_i \quad \theta_i\}^T \end{aligned} \quad (7)$$

## 2.1 Equations of motion

The equations of motion of a dynamic spline can be written in terms of the coordinates in Eq. (7) and can be deduced using Lagrange equations:

$$\left\{ \begin{array}{l} \frac{d}{dt} \frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} - \left\{ \frac{\partial \psi}{\partial q_i} \right\}^t \{\lambda\} = Q_i \quad i=1..4m \\ \{\psi\} = \{0\} \end{array} \right. \quad (8)$$

where:

$q_i$  is the  $i$ th generalized coordinate describing the  $i$ th degree of freedom of the system as in Eq. (7);

$\dot{q}_i = \frac{dq_i}{dt}$  is the time derivative of the  $i$ th generalized coordinate;

$T$  is the kinetic energy of the whole system;

$\{\psi\}$  is the vector containing the constraint equations written in terms of generalized coordinates;

$\{\lambda\}$  is the vector containing the Lagrange multipliers associated to each constraint;

$Q_i$  is the sum of all generalized forces acting on the  $i$ th coordinate  $q_i$ .

The generalized forces acting on the flexible beam have two contributions:

$$Q_i = Q_{i,elastic} + Q_{i,external} \quad (9)$$

The first one,  $Q_{i,elastic}$ , is due to the elastic forces generated by the deformation of the beams. It can be computed from the derivative of the overall elastic energy  $U_{elastic}$  with respect to the generalized coordinates as:

$$Q_{i,elastic} = - \frac{\partial U_{elastic}}{\partial q_i} \quad (10)$$

The second contribution to the generalized forces,  $Q_{i,external}$ , depends on the external forces and torques acting on the structure. The  $i$ th external force can be computed as the derivative of the corresponding virtual work  $W$  with respect to the  $i$ th generalized coordinate  $q_i$ :

$$Q_{i,external} = - \frac{\partial W_{external}}{\partial q_i} \quad (11)$$

## 2.2 Computation of elastic energy contributions

The overall elastic energy  $U_{elastic}$ , required to compute the elastic forces in (9) is composed of three terms. The first is due to bending, the second is due to stretching and the third to torsion.

The bending elastic energy is proportional to the square of the variation of the bending strain:

$$\begin{aligned} U_{bending} &= \frac{1}{2} \int_{spline} EI (\varepsilon_b - \varepsilon_b^0)^2 ds = \\ &= \frac{1}{2} \int_{u_{min}}^{u_{max}} EI (\varepsilon_b - \varepsilon_b^0)^2 \left\| \frac{dp(u)}{du} \right\| du \end{aligned} \quad (12)$$

where:

$\varepsilon_b^0$  is the bending strain of the free form spline;

$\varepsilon_b$  is the bending strain of the deformed spline;

$E$  is the Young Modulus of the material of the beam;

$I$  is the moment of Inertia of the cross section of the beam with respect to the bending axis (section is considered symmetrical).

The bending strain can be expressed with the scalar Frenet curvature  $\kappa(u)$ :

$$\varepsilon_b(u) = \kappa(u) \quad (13)$$

The generalized elastic forces due to bending can be evaluated as:

$$\begin{aligned} Q_{i,bending} &= - \frac{\partial U_{bending}}{\partial q_i} = - \frac{1}{2} \int_{spline} EI \frac{\partial (\varepsilon_b - \varepsilon_b^0)^2}{\partial q_i} ds = \\ &= - \frac{1}{2} \int_{u_{min}}^{u_{max}} EI \frac{\partial (\varepsilon_b - \varepsilon_b^0)^2}{\partial q_i} \left\| \frac{dp(u)}{du} \right\| du \end{aligned} \quad (14)$$

It is important to underline that Eq. (14) takes into account the influence of the actual beam length in the computation of the bending strain. Since the evaluation of the integral in (14) has to be performed over the entire length of the spline, the effects of beam lengthening can be taken into account, obtaining accurate results for large displacement analysis [23].

The stretching elastic energy is proportional to the square of the variation of the stretching strain:

$$\begin{aligned} U_{stretching} &= \frac{1}{2} \int_{spline} EA (\varepsilon_s(u) - \varepsilon_s^0(u))^2 ds = \\ &= \frac{1}{2} \int_{u_{min}}^{u_{max}} EA (\varepsilon_s(u) - \varepsilon_s^0(u))^2 \left\| \frac{dp(u)}{du} \right\| du \end{aligned} \quad (15)$$

where:

$A$  is the cross section area of the beam;

$\varepsilon_s^0$  is the stretching strain of the free form spline;

$\varepsilon_s$  is the stretching strain of the deformed spline;

The stretching strain can be evaluated as:

$$\varepsilon_s(u) = 1 - \left\| \frac{dp(u)}{du} \right\| \quad (16)$$

The generalized elastic forces due to stretching can be evaluated as:

$$\begin{aligned} Q_{i,stretching} &= - \frac{\partial U_{stretching}}{\partial q_i} = - \frac{1}{2} \int_{spline} EA \frac{\partial (\varepsilon_s - \varepsilon_s^0)^2}{\partial q_i} ds = \\ &= - \frac{1}{2} \int_{u_{min}}^{u_{max}} EA \frac{\partial (\varepsilon_s - \varepsilon_s^0)^2}{\partial q_i} \left\| \frac{dp(u)}{du} \right\| du \end{aligned} \quad (17)$$

The twisting energy has to be computed taking into account two terms. The first one is the contribution due to the geometrical torsion  $\tau$  of the spline, computed using Eq. (6). The second one is due to the roll of the cross section. Thus, the overall twisting strain  $\varepsilon_t$  can be computed as:

$$\varepsilon_t(u) = \tau(u) + \frac{d\theta(u)}{du} \quad (18)$$

Since geometric torsion is not defined for a straight line, we can assume it is zero in this case.

The twisting energy can be written as:

$$U_{twisting} = \frac{1}{2} \int_{spline} G I_0 (\varepsilon_t(u) - \varepsilon_t^0(u))^2 ds = \frac{1}{2} \int_{u_{min}}^{u_{max}} G I_0 (\varepsilon_t(u) - \varepsilon_t^0(u))^2 \left\| \frac{dp(u)}{du} \right\| du \quad (19)$$

where:

$G$  is the shear modulus of the material of the beam;

$\varepsilon_t^0$  is the twisting strain of the free form spline;

$\varepsilon_t$  is the twisting strain of the deformed spline.

The generalized elastic forces due to twisting can be evaluated as:

$$Q_{i,twisting} = -\frac{\partial U_{twisting}}{\partial q_i} = -\frac{1}{2} \int_{spline} G I_0 \frac{\partial (\varepsilon_t - \varepsilon_t^0)^2}{\partial q_i} ds = -\frac{1}{2} \int_{u_{min}}^{u_{max}} G I_0 \frac{\partial (\varepsilon_t - \varepsilon_t^0)^2}{\partial q_i} \left\| \frac{dp(u)}{du} \right\| du \quad (20)$$

### 2.3 Computation of inertial contributions

Since the spline is a continuous curve, the kinetic energy  $T$ , can be written as:

$$T = \frac{1}{2} \int_{spline} \{\dot{p}(u)\}^t [M] \{\dot{p}(u)\} ds \quad (21)$$

where:

$\dot{p}(u) = \frac{dp(u)}{dt}$  is the time derivative of the spline parametric expression;

$$[M] = \begin{bmatrix} \mu & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & I_0 \end{bmatrix} \text{ is the inertia matrix;}$$

$\mu$  is the linear density (mass per unit length) of the material of the beam;

$I_0$  is the polar moment of inertia of the cross section of the beam;

$L$  is the length of the spline;

$s$  is the geometrical arc length ( $ds = \left\| \frac{dp(u)}{du} \right\| du$ ).

The time derivative of the spline parametric expression can be computed as:

$$\dot{p}(u) = \sum_{i=0}^{m-1} b_i(u) \frac{dP_i}{dt} \quad (22)$$

The (22) relates the spline derivative with that of the generalized coordinates ( $\dot{q}_i$ ).

## 3 Augmented Reality Implementation

### 3.1 Hardware set-up

For the specific purpose of this investigation, the implemented AR system (depicted in Figure 2) is comprised of one input video device Microsoft LifeCam VX6000 USB 2.0 camera, one Head Mounted Display equipped with Oled displays, one Revware Microscribe GX2 mechatronic tracker and a personal computer.

### 3.2 Software implementation

All the supporting software has been implemented using C++ programming language and Microsoft Visual Studio 2003 developing suite. Routines for image processing have been developed using the open source library named ARToolkit which has been successfully used in other investigations. It can be freely downloaded from [http://sourceforge.net/project/showfiles.php?group\\_id=116280](http://sourceforge.net/project/showfiles.php?group_id=116280). It comprises a set of numerical procedures which are able to detect and recognize planar patterned marker in a video stream in real-time.

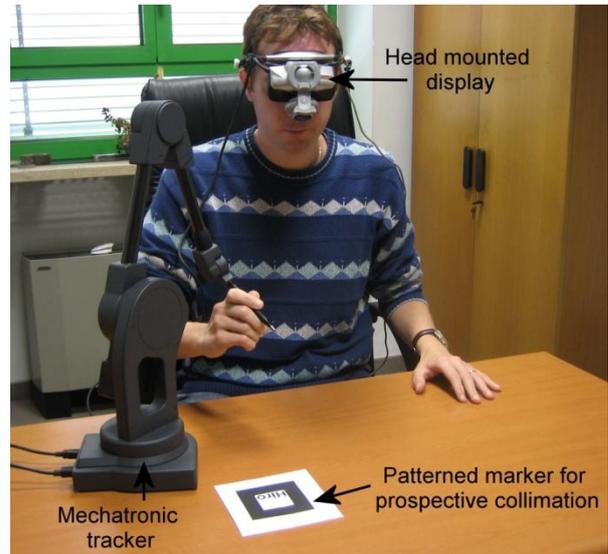


Fig. 2 Hardware setup for the Augmented Reality implementation

Using correlation techniques, the routines are also able to compute the relative position and attitude between markers and camera with good precision for visualization purposes. This computation is necessary for an accurate perspective collimation between virtual entities and real scene. The details about specific implementation and about the contents of the library go beyond the scope of this paper and the interested reader can find useful material freely downloadable from the internet site <http://www.hitl.washington.edu/artoolkit/>.

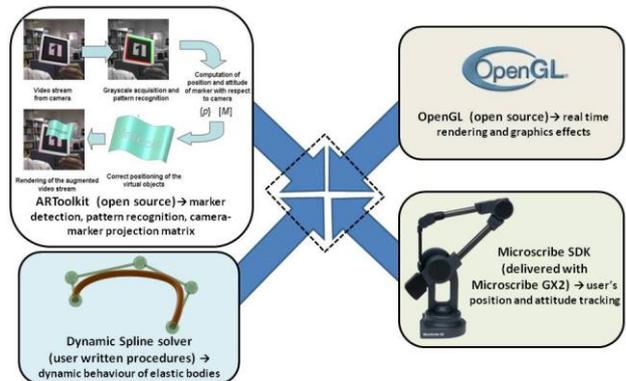


Fig. 3 Software integration strategy.

The Microscribe GX2 has been integrated using the Microscribe SDK library that allows the real-time access to position and attitude of each link of the instrumented arm.

All rendering tasks about virtual objects in the augmented scene have been performed using OpenGL

library. All these pieces of software have been integrated into a single simulation environment as shown in Figure 3.

### 3.3 Interaction between the user and the scene

The user communicates to the augmented scene by means of the Microscribe device. The tracking of the tip position and attitude is useful to interpret his intent to take part in the simulation.

The first step in the integration of the tracker in the augmented scene is the collimation between the information acquired by the instrumented device and that of the digital camera. The video stream acquired by the digital camera is elaborated by an image processing routine. It is able to recognize a patterned marker in the scene and to compute the corresponding transformation matrix between the camera and the real world. This matrix is used to project all the virtual contents in the augmented scene in the correct position and perspective.

The information acquired by the digitizer is concerned with the position and attitude of the end effector with respect to the reference frame fixed to the device itself.

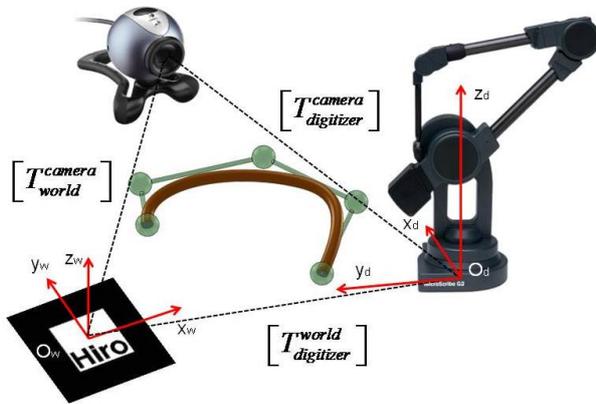


Fig. 4 Collimation between optical marker and mechanical tracking system (digitizer).

In order to ensure the collimation between the data stream coming from the camera and that from the tracker, it is important to compute the relative transformation matrix between the tracker and the world (the marker). This calibration has to be performed only at the beginning of the application and it has to be repeated only if the relative position between the world marker and the digitizer changes.

The calibration procedure can be performed by picking with the digitizer a set not-aligned points (four non-coplanar points at least) at known positions with respect to the relative frame associated to the marker.

For expressing the coordinate transformation between points, it is useful to deal with homogeneous transformation matrices which include information on both rotation and translation parameters. A generic homogeneous transformation matrix can be expressed in the form:

$$[T] = \begin{bmatrix} [\text{Orientation}]_{3 \times 3} & [\text{Position}]_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

In the same way, the generic point  $P$  can be expressed with the following coordinate vector:

$$\{P\} = \{x \ y \ z \ 1\}^T \quad (24)$$

The coordinate transformation of a generic point  $P$  from the local coordinate system fixed to the digitizer to the world coordinate system attached to the marker can be written as:

$$\{P\}_{world} = [T_{world}^{digitizer}] \{P\}_{digitizer} \quad (25)$$

where:

$\{P\}_{world}$  is the vector containing the coordinate of the point  $P$  expressed in the world reference frame;

$\{P\}_{digitizer}$  is the vector containing the coordinate of the point  $P$  expressed in the local (tracker) reference frame.

Considering a collection of points  $P_1 \ P_2 \ \dots \ P_n$ , we can build two matrices as:

$$[P]_{world} = \begin{bmatrix} \{P_1\}_{world} & \{P_2\}_{world} & \dots & \{P_n\}_{world} \end{bmatrix} \quad (26)$$

$$[P]_{digitizer} = \begin{bmatrix} \{P_1\}_{digitizer} & \{P_2\}_{digitizer} & \dots & \{P_n\}_{digitizer} \end{bmatrix} \quad (27)$$

In order to compute the matrix  $[T_{world}^{digitizer}]$  we have to solve the system of equations

$$[P]_{world} = [T_{world}^{digitizer}] [P]_{digitizer} \quad (28)$$

for the unknown elements of the matrix  $[T_{world}^{digitizer}]$ .

An homogeneous transformation matrix is defined by 6 independent parameters (three for the description of the rotation and three for the translation). For this reason, the system (6) has more equations than unknowns and the solution can be computed as:

$$[T_{world}^{digitizer}] = [P]_{world}^{-1+} [P]_{digitizer} \quad (29)$$

where the  $[P]_{world}^{-1+}$  is the pseudo-inverse matrix of the

$[P]_{world}$  matrix.

Due to numerical approximation or errors in acquisition, the orientation block of the computed matrix  $[T_{world}^{digitizer}]$  can result not exactly orthogonal. Since it represents a rigid spatial rotation, it is important to correct this imprecision. For this purpose, we can operate a QR decomposition of this orientation block:

$$[\text{Orientation}_{world}^{digitizer}]_{3 \times 3} = [R_1]_{3 \times 3} [U_1]_{3 \times 3} \quad (30)$$

where (due to the QR algorithm):

$[R_1]$  is an orthogonal matrix representing the corrected

rotation and  $[U_1]$  is a matrix whose upper band contains the errors of approximation and the lower band has only zero elements. In case of a pure rotation (orientation block without errors)  $[U_1] = [I]$ .

Finally, in order to compute the transformation matrix between the digitizer and the camera  $[T_{camera}^{digitizer}]$ , useful to collimate the acquired points to the visualized ones, a matrix multiplication has to be performed:

$$[T_{camera}^{digitizer}] = [T_{world}^{digitizer}] [T_{camera}^{world}] \quad (31)$$

Figure 5 shows some snapshots taken during a calibration procedure. The reference points are picked

using a reference cube of known dimensions (80 mm x 80 mm x 80 mm).

Once the position and the attitude of the digitizer are correctly recorded, we have to define the methods to interact with the simulation.

A possible solution is the use of a fictitious spring connected between the digitizer stylus tip and one of the control points (see Figure 6). By this way, if the tip is moved in the scene, it affects the dynamics of the spline that moves accordingly.

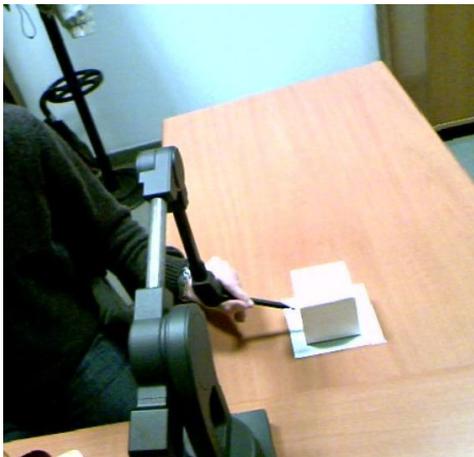


Fig. 5 Snapshots taken during calibration procedure

Mathematically, the use of a fictitious spring between the tracker stylus tip and a spline control point adds a term to the external force vector  $Q_{i,external}$  in Eqs. (8) and (9):

$$\{Q_{i,external}\} = \{Q_{i,external}\} - k \{d_{control\_point}^{digitizer\_tip}\} - c \{\dot{d}_{control\_point}^{digitizer\_tip}\}$$

where:

$\{d_{control\_point}^{digitizer\_tip}\}$  is the distance between the connected control point and the digitizer tip;

$\{\dot{d}_{control\_point}^{digitizer\_tip}\}$  is the derivative of  $\{d_{control\_point}^{digitizer\_tip}\}$  with respect to time;

$k$  is the stiffness coefficient;  
 $c$  is the damping coefficient.

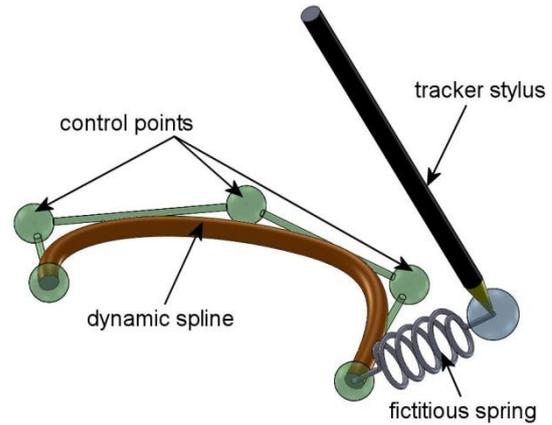


Fig. 6 A fictitious spring between the tracker stylus tip and a spline control point is used for the interaction

In order to prevent the dynamics of the spline to be affected by the presence of an external (and not physical) elastic component the value of the parameter  $k$  has to be chosen more than 5 times the bending stiffness of the beam. Moreover, the use of a damping coefficient improves the stability of the system and prevents jittering in simulation.

#### 4 Example and validation

In order to test the proposed methodology, in this section an example of implementation is presented and discussed.

The simulated scenario is a long, thin and very flexible beam that is pinned to a fixed frame. It moves under the effect of gravity and its motion can be altered by the user clipping its free end. In particular, the user can decide when the connection between the tracker and the spline has to be activated (simulating the clipping) or deactivated (simulating the release).

This scenario is also important to test the methodology with event based changes in the equations of motion. In order to achieve stable and correct results, the solution strategy has to be able to manage rapid changes in force definition.

The beam has been modelled as a B-spline with 5 control points. The gravity acts along -Y direction (downward in the vertical direction) and produces generalized forces acting on the coordinate  $y$  of each control point.

Geometrical, inertial and elastic parameters are summarized in Tab. 1.

The simulation has been performed with a fixed time step of 0.01 s. Each video frames, 4 integration steps are computed and the augmented scene is updated accordingly.

Parameter	Value	Note
Beam length	0.50 m	-
Beam section shape	circular $R = 5$ mm	constant
Beam mass	1.0 kg	homogeneous
Material Young Modulus	1.0 N/mm <sup>2</sup>	
Material Poisson coef.	0.45	

Tab. 1. Geometrical, inertial and elastic parameters of the example.

Figure 7 reports a series of five snapshots taken during the run of the simulation. The flexible beam is real-time rendered with colours (from red to green) that are related to the maximum Von Mises stress of each section,

computed from the strain evaluation according to Eqs. (13), (16) and (18).

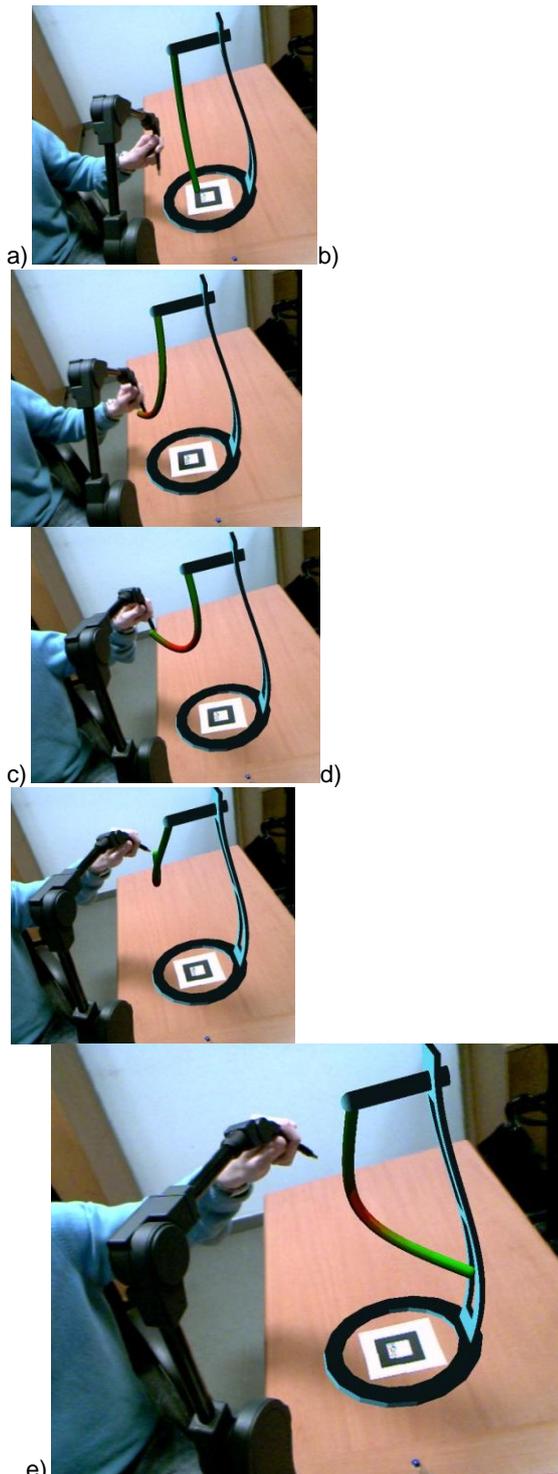


Fig. 7 Five snapshots from the discussed example

In the first part of the simulation (snapshot a of Figure 7) the beam is free to move and it is in rest position (aligned along the vertical direction).

Then, the user locates the tracker near the beam tip and activates the fictitious spring connection (snapshot b). From this moment, the tip of the beam moves subjected to this connection.

When the user moves the tracker, the tip of the beam follows it (snapshots c and d). It is important to notice that

the elastic strain and stresses are continuously evaluated, producing real-time contour plotting.

When the user decides to release the fictitious spring connection, the beam moves subjected to gravity force only and it oscillates around the equilibrium position (snapshot e).

The results of the simulation have been compared with a corresponding finite element analysis of a beam subjected to the same forces and the same imposed displacement. The structure has been modelled with 25 line (beam) elements.

Figure 8 reports a comparison between the tip trajectory of the beam simulated as a dynamic B-spline in the augmented reality environment and that computed from finite element modelling.

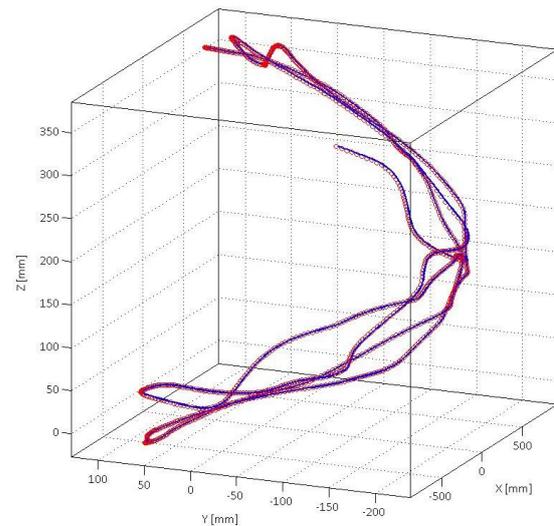


Fig. 8 Comparison between dynamic spline formulation and nonlinear finite element analysis. Continuous blue curve denotes the results coming from finite element model, the red markers denote the results coming from dynamic spline simulated in the augmented reality environment.

It can be noted a very good agreement between the two trends: after 10 seconds of simulation the differences between the tip position is lower that 2%.

The comparison between the elastic, potential and kinetic energies of the beam simulated as a dynamic spline in the augmented reality environment and those computed from finite element modeling show similar agreement. Differences are bounded below 4%.

## 5 Conclusion

The implementation of dynamic spline theory reveals to be a valid formulation for the simulation of the flexibility of one dimension elements like beams in an augmented reality environment. Such integration between flexible body simulation and augmented reality requires an appropriate representation on order to achieve real-time processing and accurate results.

Following the proposed approach, a beam can be described with a spline whose control points describe its global motion and deformation. By this way, the simulation can be performed with a smaller set of coordinates with respect to standard finite elements approaches and large displacements can be easily taken into account without approximations.

The dynamic spline formulation uses the geometrical control points to describe the displacement of the beam.

Control points aren't physical entities of the beam, but allow a better control of its deformation. Moreover, the control points have only translational degrees of freedom and this feature is an advantage for computation.

It is very important to underline that the dynamic spline formulation allows a more direct relationship between geometrical and simulation entities. In comparison with standard finite element approach, there is no need to mesh the geometry before simulation. The same mathematical representation can be used for both modelling and simulation.

The proposed implementation of an augmented reality environment makes use of a digital tracker able to increase the interaction between the user and the scene. The position of its tip can be acquired with a very good precision. Moreover, thanks to the use of optical markers and mechatronic digitizer, the visualization and tracking activities can be processed separately.

The proposed example of application confirms that the formulation is suitable for the integration in the augmented reality system, producing and accurate, realistic and real-time simulations.

## References

- [1] R.T. Azuma, Y. Baillot et al. *Recent advances in augmented reality*. IEEE Computer Graphics 21(6), (2001), pp. 34-47.
- [2] F. Liarokapis, An augmented reality interface for visualizing and interacting with virtual content, *Virtual Reality* vol. 11, (2007), pp. 23-43.
- [3] P.P. Valentini, E. Pezzuti, D. Gattamelata, *Virtual engineering in Augmented Reality*. In Computer Animation, Nova Science Publisher Inc., New York, (2010) pp. 57-83.
- [4] D. Gattamelata, E. Pezzuti, P.P. Valentini. *A CAD system in Augmented Reality application*, Proc. of the 20th European Modeling and Simulation Symposium, Briatico (CS), Italy, 2008.
- [5] P.P. Valentini, *Interactive virtual assembling in augmented reality*, International Journal on Interactive Design and Manufacturing, vol. 3, (2009), pp. 109-119.
- [6] Y. Pang, A.Y.C. Nee, S.K. Ong, M.L. Yuan, K. Youcef-Toumi. *Assembly Feature Design in an Augmented Reality Environment*, Assembly Automation, 26/1, (2006) pp. 34-43.
- [7] S. Wiedenmaier, O. Oehme, L. Schmidt, H. Luczak, *Augmented Reality (AR) for Assembly Processes Design and Experimental Evaluation*, International Journal of Human-Computer Interaction, vol. 16, (2003), pp. 497-514.
- [8] M.L. Yuan, S.K. Ong, A.Y.C. Nee, *Augmented Reality for Assembly Guidance Using a Virtual Interactive Tool*. International Journal of Production Research, vol. 46 (7), (2006), pp.1745-1767.
- [9] A.E. Uva, S. Cristiano, M. Fiorentino, G. Monno, *Distributed design review using tangible augmented technical drawings*, Computer-Aided Design, 42, (2010) pp. 364-372.
- [10] H. Aoyama, Y. Kimishima, *Mixed reality system for evaluating designability and operability of information appliances*, International Journal Interactive Design and Manufacturing, vol.3 (2009), pp.157-164.
- [11] P.P. Valentini, E. Pezzuti, *Interactive Multibody Simulation in Augmented Reality*, Journal of Theoretical and Applied Mechanics, vol 48, n.3, (2010), pp. 733-750.
- [12] A.A. Shabana. Flexible multibody dynamics: Review of past and recent developments, *Multibody System Dynamics* 1(2), (1997), pp.189-222.
- [13] R.E. Valembois, P. Fiset, and J.C. Samin. *Comparison of various techniques for modeling flexible beams in multibody dynamics*, *Nonlinear Dynamics* 12, (1997), pp. 367-397.
- [14] M. Grégoire, E. Schömer, *Interactive simulation of one-dimensional flexible parts*, *Computer-Aided Design*, vol. 39, (2007), pp. 694-707.
- [15] J.M. Ritchie, G. Robinson, P.N. Day, R.G. Dewar, R.C.W. Sung, J.E.L. Simmons, *Cable harness design, assembly and installation planning using immersive virtual reality*, *Virtual Reality* vol. 11, (2007), pp. 261-273.
- [16] P.P. Valentini, *Interactive cable harnessing in Augmented Reality*, *International Journal on Interactive Design and Manufacturing*, vol. 5, (2011), 45-53.
- [17] H. Qin, D. Terzopoulos. *D-NURBS: A physics-based framework for geometric design*. IEEE Transactions on visualization and computer graphics, vol. 2-1., (1996), pp. 85-96.
- [18] A. Theetten, L. Grisoni, C. Andriot, B. Barsky. *Geometrically exact dynamic splines*, *Computer-Aided Design* 40, (2008), pp. 35-48.
- [19] J. Lenoir, S. Cotin, C. Duriez, P. Neumann. *Interactive physically-based simulation of catheter and guidewire*. *Computer and Graphics*, 30(3), (2006), pp. 417-23.
- [20] J. Lenoir, L. Grisoni, P. Meseure, C. Chaillou. *Adaptive resolution of 1d mechanical B-spline*. In Graphite. Dunedin (New Zealand), (2005), pp. 395-403.
- [21] P.P. Valentini, E. Pennestri, *Modelling elastic beams using dynamic splines*, *Multibody System Dynamics*, (in press).
- [22] G. Farin. *Curves and surfaces for CAGD*. Morgan Kaufmann Publishers, S. Francisco, CA, fifth edition (2002).
- [23] J. Gerstmayr, H. Irschik. On the correct representation of bending and axial deformation in the absolute nodal coordinate formulation with an elastic line approach, *Journal of Sound and Vibrations*, 318, (2008), pp. 461-487.