Beam Shape Modeling and Shape Generation Method of Morphable Beam Device

Saburo Matunaga, Thomas Iljic and Yohei Tanaka (Tokyo Tech)

Key Words: Compact, Deployable, Morphable Beam, Inspection

Abstract

To conduct remote inspection missions, an experimental device using a flexible beam (bendable and with shape retention properties) without any articulated joints was developed and called Morphable Beam Device (MBD). A beam is bended as it is deployed and the end-probe is able to reach distant targets of interest. A camera is selected for the end-probe to realize much simpler system. In this paper, a beam shape modeling made by the MBD and a desired beam shape generation method are described.

1. Introduction

The great majority of the robotic arms used nowadays are rigid arms with articulated joints which introduce certain limitations: Spatial motion limitation due to its joint type, the number of joints, and the lengths between these joints. The space/weight occupied by the arm is non-negligible: presence of actuators on the arm itself, global rigidity etc... Our idea [1] is to introduce a new kind of totally flexible “arm” with no actuators on the beam itself. A shape is given to this flexible beam as it is deployed, enabling a wide range of shapes and lengths. The actuators are located on the spacecraft itself, where they push the beam out and give it his shape. We can divide the mechanism on the spacecraft in two distinct sub-mechanisms: the shaper part, which bends the beam to give it the actual shape, and the deployment part which pushes the stored beam from its storage compartment, through the shaper and to the outside of the spacecraft (Fig. 1). Contrarily to our device, an arm is by definition an articulated mechanism with joints on it. We thus decided to avoid the use of this terminology and named it “Morphable Beam Device” or MBD. The beam is free of any actuators: only the actuators located on the satellite will bend the beam while releasing it more and more.

To demonstrate this new concept, a first experimental device was developed, basic experiments had been done, and we found that the system has a retro-bending behavior[1]. In this paper, finding a behavior model, setting up a control framework for this device, including a control algorithm and an efficient control system, and the measurement of the curvature of the beam shape are conducted, and the relationship between bending displacement and the resulted beam curvature is obtained. A beam shape
modeling made by the MBD taking into the retro-bending and desired beam shape generation algorithms are described.

2. Definition and Mathematical Model

To understand and model the relationship between deployment and bending, the better way is to work with regard to the surface of deployment. Figure 2 shows the effective output surface and the resulting cones in which the deployment direction vector is contained.

Fig. 2 Output Surface and Zones

Fig. 3 shows the measurement of the curvature of the beam shape after lateral bending and deployment of the beam, which shows that at a fixed bending position, when deploying the beam, the curvature obtained is constant along the deployed part. The relationship between bending displacement and the resulted beam curvature is obtained as in Fig. 4, where the curvature drastically changes with the bending position: we call it the retro-bending. This behavior must be considered for generating beam shape algorithms. The relationship between the lateral displacement and the radius can be approximated well by combination of polynomial curves of order four and three as shown in Fig. 4(b).

Fig. 3 Beam Shape Measurement

---

(b) Relationship Between Lateral Displacement and Radius, and Approximated Curve Model

Fig. 4 Generated Beam Shapes

3. Beam Shape Generating Algorithms

In this section, a beam shape modeling made by the
MBD taking into the retro-bending and desired beam shape generation algorithms are described. The objective is to build a satisfying 3D curve passing by the points the user inputs under the condition that the curvature must be continuous along the beam to model reality. We focus on polynomial interpolation and chose the method of relaxed cubic spline curves: cubic bezier curves glued together so that the continuity of the first and second derivatives is verified. We then compute the characteristics (curvature, etc.) along the discrete beam model.

3.1 Constant Curvature Algorithm

3.1.1 Position Specified (CCA-I)

The user inputs are the followings as shown in Fig. 5:
1) Specify the direction of bending $\alpha$, namely normal plane $P$ including the beam.
2) Specify the position (one point) of the beam end.

When the output tangent is vertical, we can define a circle from two points and the tangent to one of the point, and we compute the corresponding curve, curvature and bending position[4].

Figure 6 shows an example of the algorithm CCA-I, and Fig. 7 shows an example of experimental verification, and the error is about 2-3 mm. Figure 8 shows a slice representation of the beam end positions accessible with the algorithm CCA-I and maximum beam length of 300mm.

3.1.2 Position and Direction Specified (CCA-II)

The user inputs are the followings as shown in Fig. 9:
1) Specify the direction of bending $\alpha$, namely normal plane $P$ including the beam.
2) Specify the position and direction (one point) of the beam end.

Figure 10 shows an example of the algorithm CCA-II.
For a given beam end direction, a given maximum beam length, let’s see what are the different targets accessible in a given vertical plane in a given direction by the MBD. We must note that there is also a restriction on the tangent angle at the exit point. To visualize the problem, Fig. 11 shows the maximum output cone for the tangents at point O, the exit of the deployer (this cone is shown in pink).

We can see that these conditions can be really restrictive. In fact, in the case, only one of the curves is acceptable regarding all these criterions (except the maximum beam length which isn’t specified.)

Fig. 12 shows an evolution with the direction at fixed maximum beam length and superposition for all directions in a given deployment plane P.

3.2 Variable Curvature Algorithm, VCA

The algorithm VCA implemented into the control station works the following way.

1) First, the user inputs: a series of points (n points) that define the curve, and the number of points (Nbp) to discrete the curve model into.
2) For each set of two points, a cubic Bezier curve is defined using the process earlier and such that the first and second derivatives of the curve will be continuous at the junctions.
3) A discrete representation of the curve is obtained,
then the vectors $t$, $b$, $n$ of the Frenet reference frame along the curve for each points, and the position of the center and the corresponding Radius and curvature of the osculation circle at each point are computed.

4) In order to compute the direction in which bend the beam at point $p$ as to produce the final expected shape, we compute for each point to different angle values, Theta and Beta. By processing all our points recursively starting from the deployer-end of the entire curve we want to have, we can for each step $j$ define:

a) The angle Theta$_j$ corresponding to the rotation of the vertical plane around the vertical axis such that the new plane contains $t_j$

b) The angle Beta$_j$ corresponding to the angle of the next tangent with the vertical axis.

Using this angle Beta$_j$ we then rotate the whole spline such that the next tangent becomes the vertical axis. Our beginning condition is the following: we start from the endpoint of the beam at the deployer end, with an initial vertical tangent. (the 3d cubic spline computed from the user points is rotated such that this is the case). The computation of these angle Theta$_j$ and Beta$_j$ can be summarized in Fig. 13.

Figure 14 shows an example of the algorithm VCA, and Fig. 15 is a variation of the curvature along the modeled beam.

![Representation of the angles along the discrete curve](image)

**Fig. 13** Recursive computing of the angles along the curve

![Variable Curvature Shape Generation Example](image)

**Fig. 14** Variable Curvature Shape Generation Example

![Curvature Distribution Along the Beam](image)

**Fig. 15** Curvature Distribution Along the Beam

4. An Application Example: Remote Inspecting

In this section, the case is considered where the satellite would approach a spacecraft’s orbit
sufficiently near to have a zero relative speed seen from the spacecraft. Let’s take for example the space-shuttle. Our satellite’s mission would be to inspect the surface of the spacecraft and particular targets. In a usual case, with no Morphable Beam Device (MBD), we would have to move the satellite around in order to inspect the surface all around the shuttle: front, back, top, etc. This would be very costly in propellant, which is very limited on small spacecrafts like satellites. Furthermore these maneuvers can be dangerous, with risks of collision, loss of control, etc. Such a mission would be very limited and would be very difficult to adapt and update the objectives during the mission itself.

Suppose now that our satellite is equipped with an MBD system and that we managed to position it at a relatively close, but not too much, distance from the spacecraft, as shown in Fig. 16. By deploying our beam, which if we found a good material could be tens of meters long, we can now go and inspect all around the spacecraft without moving the satellite and wasting our precious propellant. Since the spacecraft’s position, once stabilized, doesn’t move relatively to the shuttle, we can enter a rough three dimensional model of the shuttle’s position as seen from the MBD in our control software.

Fig. 16  Application Example: Remote Inspection Using MBD

5. Conclusions
In this paper, a beam shape modeling, beam shape generation algorithms by specifying the beam end position and/or direction, and also an application example of Morhable Beam Device (MBD) were discussed. In the future, we will develop a flight model of the MBD applicable to install on a satellite and will demonstrate it on orbit.

References