Simulation of a Hyper Redundant Arm

Giuseppe Boccolato*, Ionuț Dinulescu **,
Alice Predescu***, and Dorian Cojocaru******
University of Craiova
**gboccolato@gmail.com, **ionutdinulescu@gmail.com,*
****predescualice@gmail.com, ******cojocaru@robotics.ucv.ro*

Abstract — This paper deals with the subject of a hyper-redundant robot control simulation. This type of arm changes its configuration by bending a continuous backbone formed of sections connected in a serial configuration. Theoretically the arm can achieve any position and orientation in the working space. A tentacle arm prototype was designed and the practical realization is now running in open loop. The goal of this paper it's to establish the constructive way to build the visual algorithm to control the system. One limitation introduced about the visual serving system is that it does not answer in real time. The delay introduced by the image-processing task is taken into consideration and simulations are performed to design Closed-Loop control that stabilizes the system.

Keywords: hyper-redundant, arm, simulation, control, robot.

1. INTRODUCTION

The hyper-redundant robotic structures, also identified as tentacular robots or continuum robots, represent a robotic class of great interest and also of great complexity. The mathematical modeling of such a robot takes under consideration the kinematics approach. The direct kinematics models are based on the concept of curved segment and the concatenation of these segments to form a continuum multi-segment (Cieslak and Morecki, 1999; Hirose, 1993; Immega and Antonelli, 1995). The literature on this argument is wide between them, we found interesting: Ivanescu et al., 2006; Chirikjian 1995; Gravagne and Walker, 2000; Walker and Carreras, 2006. Various solutions have been tried in order to obtain a fast and accurate control algorithm.

The research group from the University of Craiova, Romania started working in the field of hyper redundant robots over 20 years ago. Since 2008, the research group designed a new experimental platform for tentacle manipulators. First, a cylindrical structured was designed and experimented. A new prototype based on truncated cone segments was designed and implemented (Boccolato et al. 2009, 2010; Cojocaru et al.,2010; Dinulescu et al., 2010).

The version built now, is actuated by stepper motors. The rotation of these motors generates the actuation for cables which, by correlated screwing and unscrewing of their ends, determines their shortening or prolonging. By consequence, the tentacle curvature changes with the cables length variation. The backbone of the tentacle is an elastic rod made out of steel, which sustains the entire structure and allows the bending. Depending on which cable shortens or prolongs, the tentacle bends in different planes, each one making different angles (rotations) respective to the initial coordinates frame attached to the manipulator segment – i.e. allowing the movement in 3D.

In this paper, the inverse kinematics problem is reduced to determining the time varying backbone curve behavior. The control of hyper-redundant robots is a complex problem, and known solutions are based on visual servoing control. A specialized module was developed in order to fulfill this task. In the control law the visual system it is rapresented as a delay. The acquisition system it is realized and the algorithm of acquisition is implemented. A clustering algorithm was developed and adapted for the particular case of hyperredundant robots. This algorithm identifies the bases of each robot segment as a cluster and computes their centers. The coordinates (obtained in image space) can then be transformed to virtual control space (a coordinate system used in the virtual simulation environment) in order to compare the real experimental values to the simulation ones.

The visual servoing control system is based on binocular vision. The continuous measurement of the arm's parameters derived from the real time computation of the binocular optic flow of the two images has been compared to the desired arm position (Tanasie et al., 2009). The error control function has been built in the 3D Cartesian space using the video information obtained from the two cameras after a fixed time. The 2D errors obtained in the planes of the two images have been determined by the differences between the existing values and the ones desired for the angles that define the arms projections. Having non real-time system, the measures will be taken almost in the time we want. To correct the possible errors the algorithm uses two different threads, one for the acquisition and one for the picture analysis. Using a circular buffer we are able to measure the position and the speed in the operation space.

2. TENTACLE ARM KINEMATICS

For the 3D case, a virtual wire is considered, which gives the direction of the curvature (α).

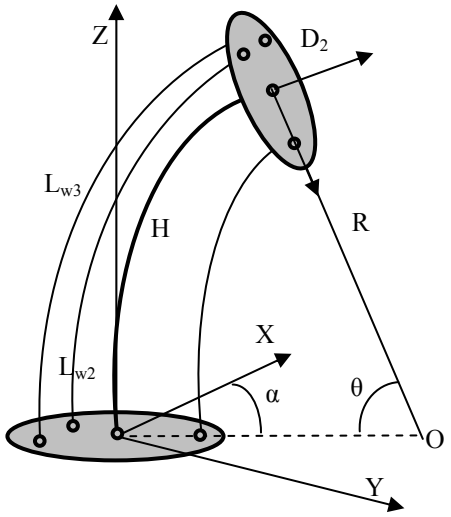

Fig. 1. The geometry of one segment.

The direction of the virtual wire must coincide with the direction of the desired curvature. The lengths in (1) are computed from Fig 1.:

$$\begin{aligned} L_{11} &= R+\frac{D_1}{2}\cdot\cos(\alpha_1) \qquad L_{12} = R+\frac{D_2}{2}\cdot\cos(\alpha_1) \\ L_{21} &= R+\frac{D_1}{2}\cdot\cos(\alpha_2) \qquad L_{22} = R+\frac{D_2}{2}\cdot\cos(\alpha_2) \\ L_{31} &= R+\frac{D_1}{2}\cdot\cos(\alpha_3) \qquad L_{32} = R+\frac{D_2}{2}\cdot\cos(\alpha_3) \end{aligned} \tag{1}$$

where α1 = -α, α2 = 120° - α, α3 = 240°-α, D1 and D2 are the diameters of the discs.

Based on the relation (1) the curvature radia R1, R2 and R3 of the three control wires are then obtained according to the relation (2).

$$\begin{aligned} R_1 &= \frac{\sqrt{L_{11}^2 + L_{12}^2 - 2\cdot L_{11}^2\cdot L_{12}^2\cdot\cos\theta}}{2\cdot\sin\frac{\theta}{2}} \\ R_2 &= \frac{\sqrt{L_{21}^2 + L_{22}^2 - 2\cdot L_{21}^2\cdot L_{22}^2\cdot\cos\theta}}{2\cdot\sin\frac{\theta}{2}} \\ R_3 &= \frac{\sqrt{L_{31}^2 + L_{32}^2 - 2\cdot L_{31}^2\cdot L_{32}^2\cdot\cos\theta}}{2\cdot\sin\frac{\theta}{2}} \end{aligned} \tag{2}$$

In depth information on how relations (2) are obtained can be found in Dinulescu et al., 2010.

Finally, the lengths of the control wires are computed with the relation (3):

$$\begin{aligned} L_{w1} &= R_1\cdot\theta \\ L_{w2} &= R_2\cdot\theta \\ L_{w3} &= R_3\cdot\theta \end{aligned} \tag{3}$$

The desired shape of the tentacle arm is obtained by controlling the θ and α angles of the three segments. For each of the segments we impose the interval time to go from the current shape to the desired shape of the individual segments. This approach allows the control wires to move with constant velocities, easily calculated with relation (4):

$$v_i = \frac{L_{wi}^f - L_{wi}^i}{T_f}, \tag{4}$$

where L_{wi}^f and L_{wi}^i are the final and initial lengths of the control wires for the i-th segment and Tf is the imposed interval time. The position of the segment end-point is calculated based on the values of the θ and α angles.

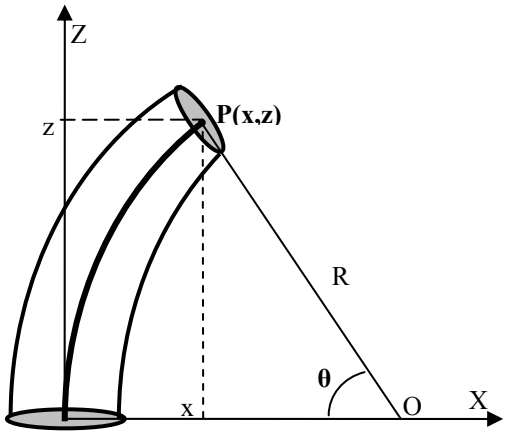

Fig. 2. The position of the segment's end-point.

For the 2D case, the coordinates of the terminal point are calculated with the following relations, obtained using the geometrical representation in Fig. 2:

$$\begin{aligned} x &= R - R\cos(\theta) \\ z &= R\sin(\theta) \end{aligned} \tag{5}$$

where R=H/θ and H is the height of the segment and θ is the curvature angle. The 3D position of the terminal point takes in consideration also the direction angle α. The position is calculated from the relation (5) by rotating the end point with angle α against the z axis. Relation (6) shows how the 3D position is obtained:

$$\begin{aligned} x &= [R - R\cos(\theta)]\cdot\cos(\alpha) \\ y &= [R - R\cos(\theta)]\cdot\sin(\alpha) \\ z &= R\sin(\theta) \end{aligned} \tag{6}$$

3. DIFFERENTIAL KINEMATICS

The position control system is used for controlling the (x,z) position of the arm's endpoint. For the 2D case, the relations (5) show that the x and z relations are dependent, therefore it is enough to specify only the coordinate x for the reference position. The z coordinate of the reference position will be calculated using the relation (7).

$$z = R\sin(\arccos\ (1-\frac{x}{R})) \tag{7}$$

From relation (7), the x coordinate is expressed as a function of the curvature angle θ, as in (8).

$$x = f(\theta) = R(1-\cos\ \theta) \tag{8}$$

The velocity of the x-cordinate is easily computed by the derivation of (8) in respect to the curvature angle θ. The differential model for one segment is then obtained with the relation (9).

$$\partial x = \frac{\partial f(\theta)}{\partial \theta} = J(\theta)\cdot\partial\theta \tag{9}$$

where x is the x-axis coordinate, θ is the curvature angle and J is the Jacobian, calculated as in (10) using relations (7) and (9):

$$J(\theta) = \frac{\partial f(\theta)}{\partial \theta} = -\frac{H\cdot(1-\cos(\theta))}{\theta} + \frac{H\cdot\sin(\theta)}{\theta} \tag{10}$$

In the 3D case, the position of the segment's end point depends both on the curvature angle θ and direction angle α. The relation between the position velocity and the internal parameters velocities is expressed as in the relation (11):

$$\partial\begin{bmatrix} x \\ y \\ z \end{bmatrix} = J\cdot\partial\begin{bmatrix} \theta \\ \alpha \end{bmatrix} \tag{11}$$

The Jacobian J is the matrix calculated in relation (12) below.

$$J = \begin{bmatrix} \frac{\partial x}{\partial\theta} & \frac{\partial x}{\partial\alpha} \\ \frac{\partial y}{\partial\theta} & \frac{\partial y}{\partial\alpha} \\ \frac{\partial z}{\partial\theta} & \frac{\partial z}{\partial\alpha} \end{bmatrix} \tag{12}$$

The elements of the 3D Jacobian are obtained by derivation of the relations (6) in respect to the appropiate angle.

4. IMAGE BASED SYSTEM CONTROL

Two video cameras provide two images of the whole robot workspace. The two image planes are parallel with XOY and ZOY planes from robot coordinate frame. The cameras provide the images of the scene that are stored in the frame grabber's video memory. Respective to the image planes are defined two dimensional coordinate frames, called screen coordinate frames or image coordinate systems. Denote $X_{S_1}Y_{S_1}$ and $Z_{S2}Y_{S_2}$, respectively, the axes of the two screen coordinate frames provided by the two cameras. The spatial centers for each camera are located at distances D1 and D2, with respect to the XOY and ZOY planes, respectively. The orientation of the cameras around the optical axes with respect to the robot coordinate frame, are noted by Φ and ψ, respectively (Fig. 3).

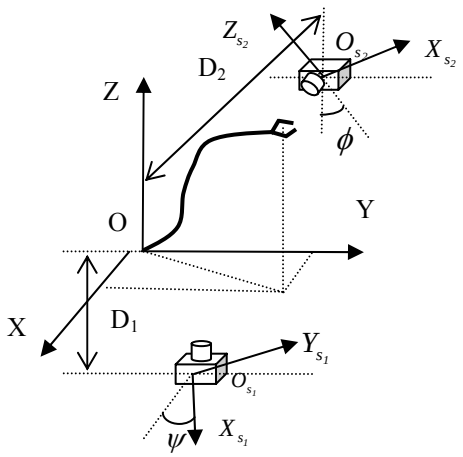

Fig. 3. The robot coordinate system.

The control system is an image – based visual servo control where the error control signal is defined directly in terms of image feature parameters. The desired position of the arm in the robot space is defined by the curve Cd, or, in the two image coordinate frames $Z_{S_1}O_{S_1}Y_{S_1}$ and $Z_{S_2}O_{S_2}Y_{S_2}$, by the projection of the curve C in the image coordinates frame (Fig. 3).

The control problem of this system is a direct visual servo-control, but the classical concept of the position control, in which the error between the robot end-effector and target is minimized, is not used. In this application the control of the shape of the curve in each point of the mechanical structure is used. The method is based on the particular structure of the system defined as a "backbone with two continuous angles θ and q". The control of the system is based on the control of the two angles θ(s) and q(s) These angles are measured directly or indirectly. The angle θ(s) is measured directly by the projection on the image plane $Z_{S_1}O_{S_1}Y_{S_1}$ and q(s) is computed from the projection on the image plane $Z_{S_2}O_{S_2}Y_{S_2}$.

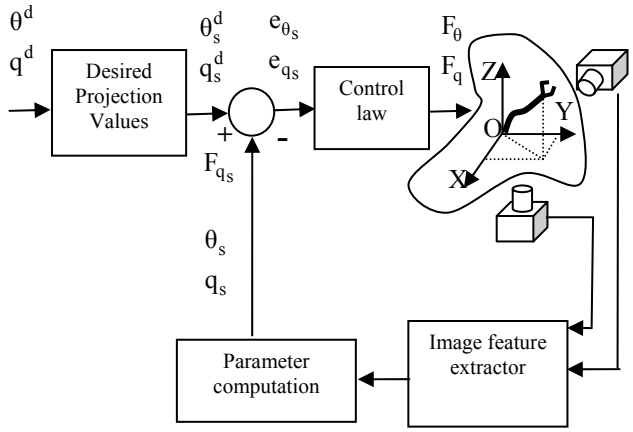

Fig. 4. The global control system.

In order to implement the visual-servoing system, a benchmark was organized based on two color camera with 0.05 lux low light sensitivity and the DT3162 frame grabber from Data Translation.

The cameras have motorized Pant/Tilt/Zoom (10x optical zoom) and are mounted in perpendicular planes offering the input for the frame grabber. The Pant/Tilt/Zoom precision is sufficient for this step of the application development (Pan: range +135, 10 50/sec; Tilt: range +90 45, 7 25 /sec; Zoom: 1x~10x optical zoom). Two white screens are placed in front of the cameras in order to increase the images contrast. The tentacle arm is placed between each camera and its background screen.

The image processing tasks are performed using Global LAB Image2 from Data Translation. The robot control algorithms are implemented in a C++ program running on a Pentium IV PC. In order to facilitate the image feature extraction, a set of markers are placed on joints along the backbone structure.

A very important task in developing this application is to control the camera position and orientation. From this point of view, the calibration operation assures that the two cameras' axes are orthogonal. The calibration for a pan/tilt/zoom camera orientation is achieved by means of an engineered environment and a simulation graphic module [4]. In the beginning, the tentacle manipulator receives the needed commands in order to stand in a test pose (imposed position and orientation). The term "camera calibration" in the context of this paper refers only to positioning and orienting the two cameras at imposed values.

5. CLOSED LOOP CONTROL

The control wires are actuated by stepper motors which allow open-loop control of the robot's shape. However, the stepper motors may loose steps, which results in large positioning errors. In order to compensate the errors, the control loop has been closed with an artificial vision system which measures the current curvature angles of the segments.

The differential kinematics obtained in the relation (10) is used by the control law in order to get the reference position. The closed-loop control system is shown in Fig.5.

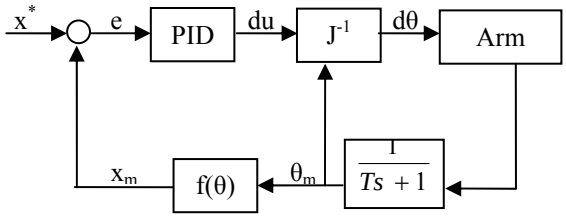

Fig. 5. The differential control system.

The video camera is modelled as a first-order system, as in relation (13), where T is the time constant of the camera system:

$$VC(s) = \frac{1}{Ts + 1} \tag{13}$$

The video acquisition system measures the curvature angle θm, based on which the measured value of the end point's x coordinate is then obtained, using relation (8). The error calculated as the difference between the reference position x* and the measured value xm is the input of the PID controller, which computes the command du. Using relation (13), the dθ command value is computed and transmitted to the arm's control system.

6. SIMULATION AND EXPERIMENTS

A MatLab simulation software has been created in order to facilitate experiments (Fig.6). The simulator allows full configuration of the mechanical parameters of the robotic arm, which include the diameters of the discs, and the distances between them. The initial and final shapes of the tentacle arm are specified by setting the corresponding values for the α and θ angles for each of the three segments. Then, the software simulates the shape transformation of the arm using the control approach proposed in the previous sections of this paper.

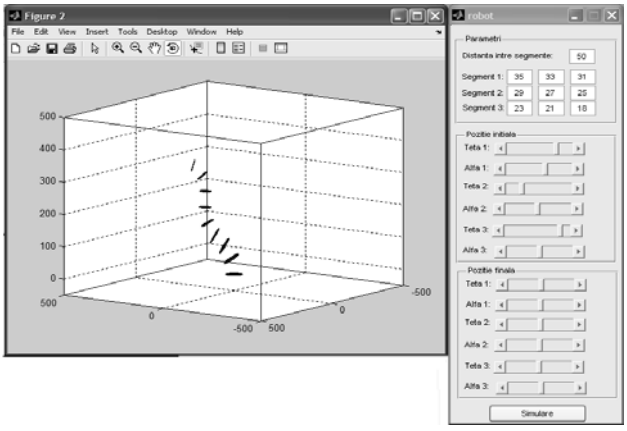

Fig. 6. The MatLab simulator of the tentacle arm.

Fig. 7 shows the evolution simulation of one arm segment when its corresponding curvature angle changes in time. The end point trajectory is also drawn by the simulator.

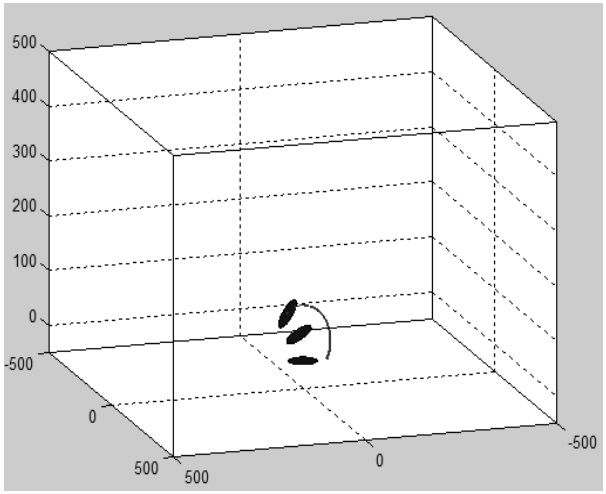

Figure 7. Trajectory simulation of the tentacle arm.

Fig. 8 shows the length variation of the control segments for a 2D motion ($\alpha = 0$), where θ goes from $-\pi/4$ to $\pi/4$. It can be remarked that the plots intersect when $\theta=0$, which corresponds to the position where the lengths of the three control segments are equal.

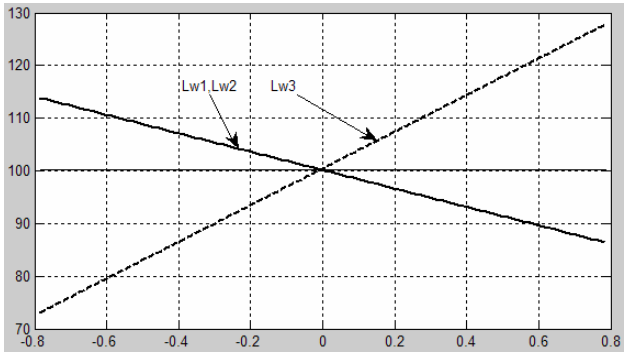

Figure 8. The evolution of the control wires lenghts for the 2D case.

The closed-loop control system has been implemented using the Simulink environment. The Simulink model is presented in Fig. 9.

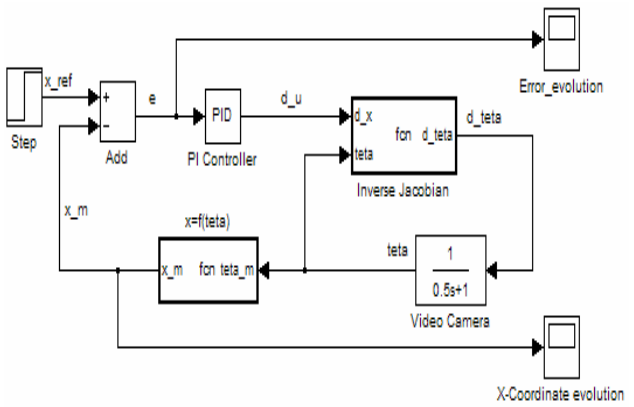

Fig. 9. The Simulink model of the control system.

The Simulink model allowed the tunning of the PID controller and evaluation of the control system performance. Fig. 10 shows the evolution of the curvature angle and of the system's error as a response to a one-unit step input.

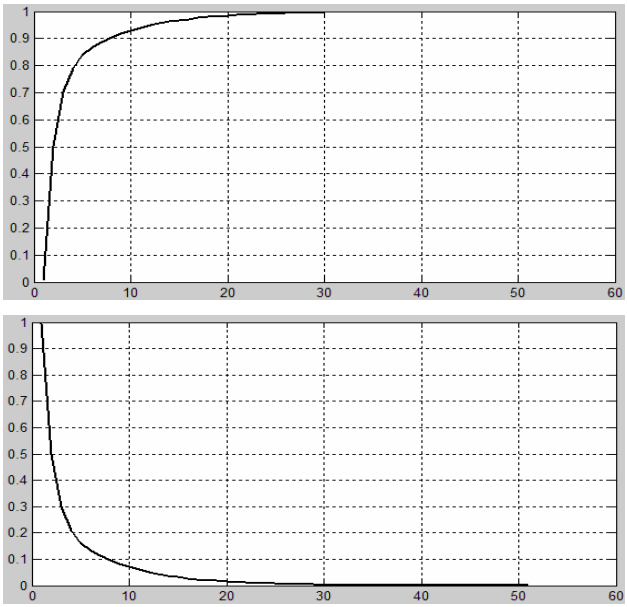

Fig. 10. Evolution of the a) system output, b) system error.

Fig. 11 shows the variations of the x and z coordinates when θ goes from $-\pi$ to π, while α is constant. From this plot results the circular shape of the segment's work envelope that gives its maximum reachability in the operation space.

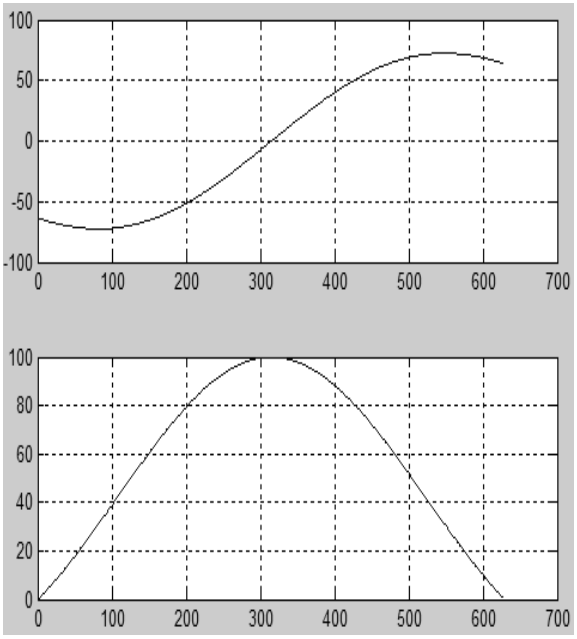

Fig. 11. Variation of the x and z coordinates for the end-point.

A VRML model of the tentacle segment has been developed, in order to achieve a better understanding of how the robot performs in real life (Fig. 12).

The VRML model uses the output data provided by the Matlab simulation described in this paper and renders the shape of the robot in a VRML enabled browser. The robot shape is obtained from MatLab simulation as an array of 3D points that is used by the VRML code for defining the spine field of the VRML extrusion transformation.

Based on a timer event, a coordinate interpolator object switches between multiple shapes of the robot calculated by the MatLab simulator, achieving a smooth animation of the robot movement.

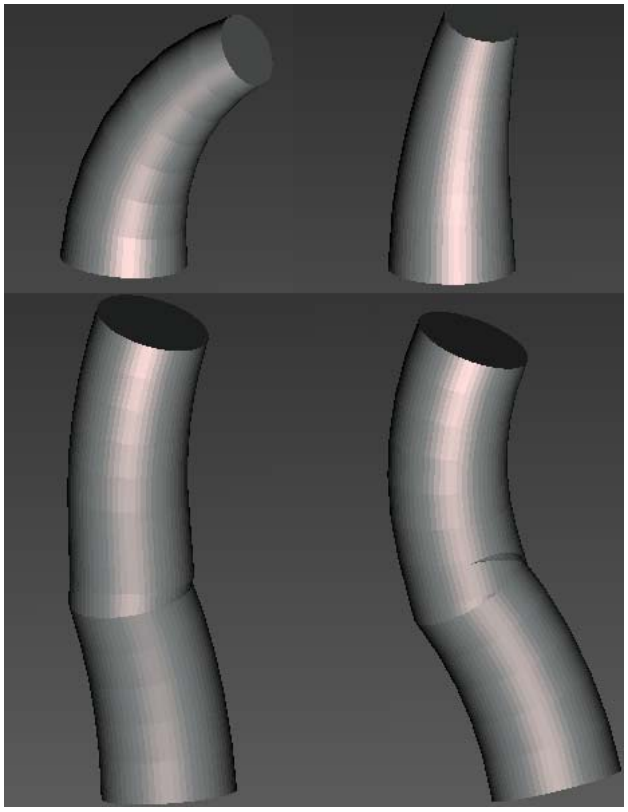

Fig.12. The VRML model of the tentacle robot.

7. CONCLUSIONS AND FUTURE WORK

The kinematics model of the hyper-redundant arm, along with simulation results have been presented in this paper. Also a vision based control system has been proposed. The control system uses two external cameras to capture the current shape of the robotic arm. The controller uses a traditional PID control law and the differential kinematics of the arm. A MatLab simulator has been designed in order to test the kinematics of the robot and closed-loop control system. The simulation output data is used by a VRML script in order to render a realistic animation of the robot.

The next step in our research involves the implementation of the controller on the real robot.

Acknowledgement: The research presented in this paper was supported by the Romanian National University Research Council CNCSIS through the IDEI Research Grant ID93.

REFERENCES

Blessing, M. and Walker, I.D. (2004). Novel Continuum Robots with Variable Length Sections, *Proc. 3rd IFAC Symp. on Mechatronic Syst.* Sydney, Australia, pp. 55-60.

Boccolato, G., Dinulescu I., Predescu A., Manta F., Dumitru S. and Cojocaru D.(2010). 3D Control for a Tronconic Tentacle, IEEE 12th International Conference on Computer Modelling and Simulation, pp. 380-385.

Boccolato, G., Manta, F., Dumitru, S. and Cojocaru, D. (2009). 3D Control for a Tentacle Robot, *3rd International Conference on Applied Mathematics, Simulation, Modelling (ASM'09)*, Athens Greece.

Chirikjian, G. S. and Burdick, J.W. (1995). Kinematically Optimal Hyper-redundant Manipulator Configurations, *IEEE Transaction on Rob. and Autom.*, vol. 11, no. 6, pp. 794-798.

Cieslak, R. and Morecki, A. (1999). Elephant Trunk Type Elastic Manipultor, *Robotica*, Vol. 17, pp. 11-16.

Cojocaru Dorian, Boccolato Giuseppe, Dinulescu Ionut, Predescu Alice, Manta Florin, Dumitru Sorin (2010). 3D Control for a Truncated Cone Tentacle Kinematics, Solid State Phenomena, ISSN: 1662-9779, Vols. 166-167 (2010), Robotics and Automation Systems, ISBN/ISBN-13: 3-908451-88-4/978-3-908451-88-4, pp 127-132, Trans Tech Publications, Switzerland.

Dinulescu, I., Predescu, A., Boccolato, G., Tanasie, R.T., Cojocaru, D.(2010). Control of a Hyper-redundant Robot, Proceedings of the RAAD 2010 19th International Workshop on Robotics in Alpe-Adria-Danube Region, ISBN 978-1-4244-6884-3, p435-440, June 23-25, 2010, Budapest, Hungary [IEEE Catalog Number CFP1075J-CDR].

Gravagne, I.A. and Walker, I.D.(2000). On the Kinematics of Remotely – Actuated Continuum Robots, *Proc. 2000 IEEE International Conference on Robotics and Automation*, San Francisco, pp. 2544-2550.

Hirose, S. (1993). Biologically Inspired Robots, *Oxford University Press*.

Immega, G. and Antonelli, K. (1995). The KSI Tentacle Manipulator, *Proc. 1995 IEEE Conference on Robotics and Automation*, pp. 3149-3154.

Ivănescu, M., Cojocaru, D., Popescu, N., Popescu, D. and Tănasie, I.D. (2006). Hyperredundant Robot Control by Visual Servoing, *Studies in Informatics and Control Journal*, Volume 15, Number 1, p93-102, ISSN 1220-1766.

Tanasie, R., Ivănescu, M., Cojocaru, D., 2009, Visual Servoing Camera Orienting and Positioning Algorithm for Tentacular Robot Control, RAAD 2009, 18th International Workshop on Robotics in Alpe-Adria-Danube Region, ISBN 978-606-521-315-9, paper ID 027, May 25-27, Brasov, Romania.

Walker, I.D. and Carreras, C. (2006). Extension versus Bending for Continuum Robots, *International Journal of Advanced Robotic Systems*, Vol. 3, No.2, 2006, ISSN 1729-8806, pp. 171-178.