3D Kinematics of a Tentacle Robot

Giuseppe Boccolato, Florin Manta, Sorin Dumitru, Dorian Cojocaru

University of Craiova, Romania

Abstract — This paper deals with the control problem of a class of tentacle arms [7], [14], [15]. A tentacle robot produces changes of configuration using a continuous backbone made of sections which bend. The lack of discrete joints is a serious and difficult issue in the determination of the robot's shape. A tentacle arm has a variable length and theoretically it can achieve any position and orientation in 3D space. In order to get a better control in the constraint operator space, it is possible to increase the length of the tentacle [1]. A tentacle arm prototype was designed and the practical realization is now running.

Keywords — tentacle robot, motion control, modelling.

I. INTRODUCTION

Starting with 2008, the research group designed a new experimental platform for hyper redundant robots[1]. This new robot is actuated by stepper motors. The rotation of these motors rotates the cables which, by correlated screwing and unscrewing of their ends, determine their shortening or prolonging, and by consequence, the tentacle curvature. In the actual stage the manipulator is formed of three segments. For this presentation we consider that all segments are cylindrical. A new prototype based on truncated cone segments was designed and implemented. The backbone of the tentacle is an elastic cable 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 coordinate frame attached to the manipulator segment – i.e. allowing the movement in 3D.

Due to the mechanical design, it can be assumed that the individual cable torsion, respectively entire manipulator torsion can be neglected. The structure control will not be an open loop one, but a structure based on the information given by a robotic vision system which is able to offer the real 3D positions and orientations of the tentacle segments.

The tentacular arm is designed to be actuated by 3-phase stepper motors. The interfaces are pulse direction based without rotation monitoring. Set-point position of the stepper motor is preset as a pulse signal by a controller via signal interface. A pulse corresponds to one step of the motor. An electronic relay contact reports operating readiness. Three stepper motors are used for each segment of the tentacle.

4-Axis Stepper Motion Controller boards are used. It is a pulse train motion controller, which provides T/S curve control, on-the-fly speed change, non-symmetric acceleration and deceleration profile control, and simultaneous start/stop functions. This controller also offers card index settings for multiple cards in one IPC system. The boards offer powerful speed change functions that can be executed while the axis is moving. After motion begins, the target speed can be changed as needed according to the application. By using either a software function or external input signal, the controller can perform simultaneously starts and stops on multiple axes in a one-card configuration, or multiple axes in a multiple-card application (our case).

A tentacle arm prototype was designed and implemented. It is a cable-based mechanism having, in the first implementation, three segments (CAD images during the simulation in Fig. 1a and model implementation in Fig. 1b).

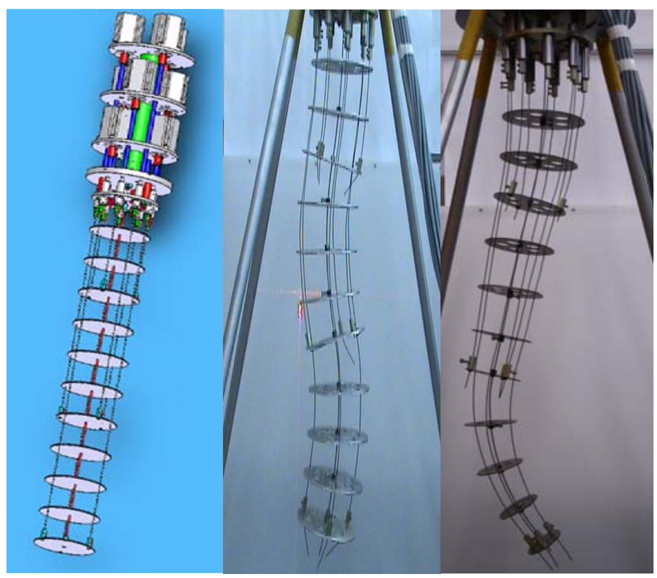

Fig. 1. A tentacle arm

Manuscript received November 30, 2009.

The research focused on hyper redundant robots to the Mechatronics Dept., Faculty of Automation, Computers and Electronics, University of Craiova, Romania, is supported by the Romanian National University Research Council CNCSIS under ID_93 IDEI grant.

Giuseppe Boccolato is an Early Stage Researcher supported by the European Commission under Marie Curie Program Network, FreeSubnet project, code MRTN-CT-2006-036186 gboccolato@robotics.ucv.ro

Florin Manta is Ph.D. student, florin_manta@robotics.ucv.ro

Sorin Dumitru is Ph.D. student, dumitru@robotics.ucv.ro

Dorian Cojocaru is full professor cojocaru@robotics.ucv.ro

II. GENERIC MODEL OF TENTACLE ARM

A. *Problem formulation*

In order to control a hyper-redundant robot we have to develop a method to compute the positions for each one of his segments [2]. [3]. By consequence, *given* a desired curvature $S^*(x, tf)$ as sequence of semi circles, *identify* how to move the structure, to obtain s(x, t) such that

$$\lim_{t \to t_f} s(x,t) = S^*(x,t_f) \qquad (1)$$

where *x* is the column vector of the shape description and t_f is the final time (see Fig. 2).

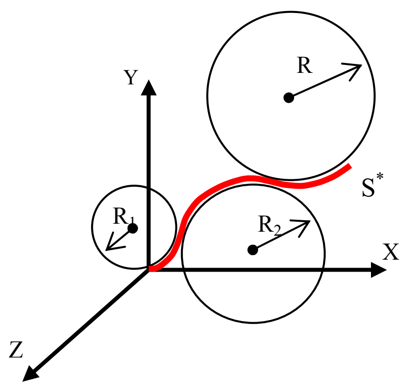

Fig. 2 The description of the desired shape

B. *Tentacle shape description*

To describe the tentacle's shape we will consider two angles *(α, θ)* for each segment, where *θ* is the rotation angle around Z-axis and *α* is the rotation angle around the Y-axis (see Fig. 2). To describe the movement we can use the roto-translation matrix considering $\theta = 2\beta$ as shown in Fig. 3.

The generic matrix in 2D that expresses the coordinate of the next segment related to the previous reference system can be written as follow:

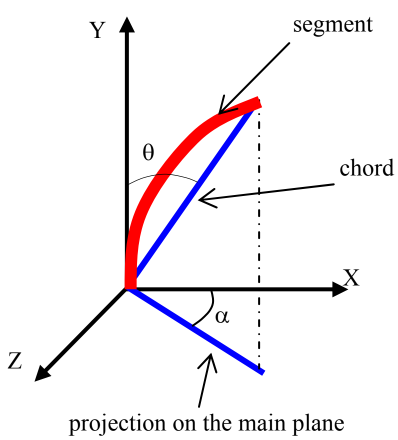

Fig. 3 The description of a 3D

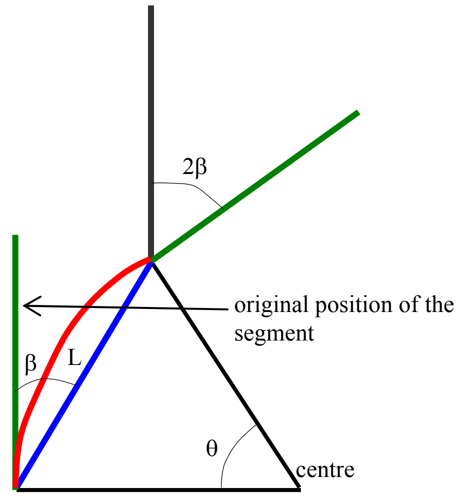

Fig. 3 Curvature and relation between θ and β

$$\begin{bmatrix} \cos(2\cdot\beta) & \sin(2\cdot\beta) & L\cdot\sin(\beta) \\ -\sin(2\cdot\beta) & \cos(2\cdot\beta) & L\cdot\cos(\beta) \\ 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

In 3D space we cannot write immediately the dependence that exists between two segments. This relation can be obtained through the pre-multiplication of generic roto-translation matrix. One of the possible combinations to express the coordinate of the next segment related to the frame coordinate of the previous segment is the following:

$$R^i{}_{generic} := R_z^i(\theta^i)\cdot Tr_y(V^i)\cdot R_y^i(\alpha^i)\cdot R_z^i(\theta^i) \qquad (3)$$

where $R_z^i(\theta^i)$ and $R_y^i(\alpha^i)$ are the fundamental roto-translation matrix having 4x4 elements in 3-D space, and $Tr_y(V)$ is a 4x4 elements matrix of pure translation in 3-D space and where V^i is the vector describing the translation between two segments expressed in coordinate of i-th reference system.

The main problem remains to obtain an imposed shape for the tentacle arm. In order to control the robot, we need to obtain the relation between the position of the wires and the position of the segment [8], [11].

III. CURVATURE OF ONE SEGMENT OF CILINDRICAL ROBOT

A. *Direct kinematic of the wires*

In the current stage of our research, a decoupled approach is used for the robot control scheme. Thus the three segments are controlled separately, without considering the interaction between them. Considering the segments of the tentacle separately, then $(\alpha, \theta)_i$ is the asigned coordinate of i-*th* segment. Having as purpose to command the robot to reach the position $(\alpha, \theta)_i$, the following relation is useful:

$$R = \frac{\overline{L}_{CB}}{\theta} \quad \forall \theta \neq 0 \tag{4}$$

where R represents the curvature's radius of the central bone and $\overline{L}_{CB}$ is a constant, equal to the length of the central bone.

Once we have θ and α together as parameters of the desired shape, and after we obtained *R,* we can compute the corresponding lengths of the wires. Depending on the types of wires and on the structure of the tentacle, we must choose the way to compute the length of each wire.

For the hard wire, made from the same material as the central bone, and by consequence having the same elasticity, referring to Fig. 4, we can write:

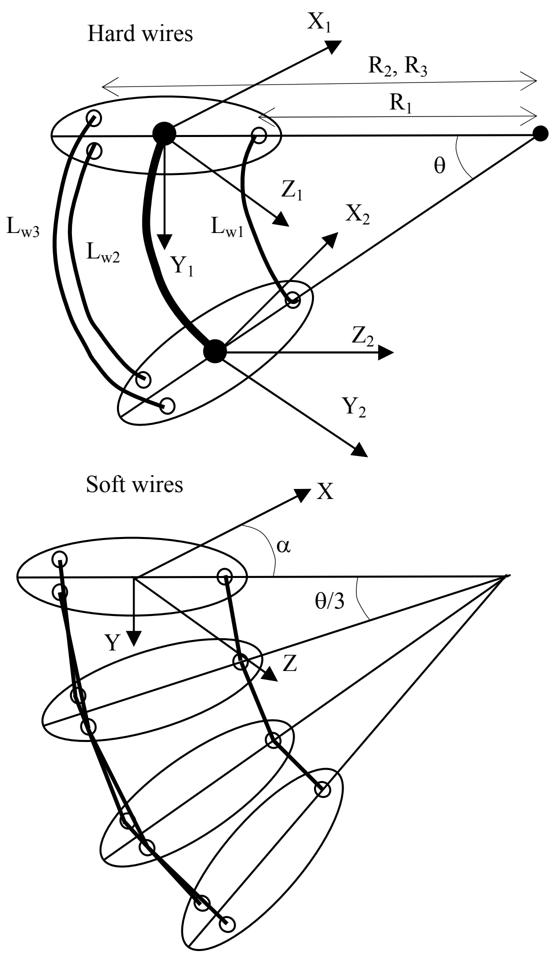

Fig. 4 Different types of wires.

$$\begin{cases} L_{w1} = R_1 \cdot \theta \\ L_{w2} = R_2 \cdot \theta \\ L_{w3} = R_3 \cdot \theta \end{cases} \tag{5}$$

For the soft wires, we can write:

$$\begin{cases} L_{w1} = [R_1 \cdot \theta] \cdot \dfrac{\sin(\theta / i)}{\theta / i} \\ L_{w2} = [R_2 \cdot \theta] \cdot \dfrac{\sin(\theta / i)}{\theta / i} \\ L_{w3} = [R_3 \cdot \theta] \cdot \dfrac{\sin(\theta / i)}{\theta / i} \end{cases} \tag{6}$$

where L_{wn} is the length of the n-th wire and R_i is the radius of the curvature of the real i-th wire.

Farther it can be written:

$$R_n = (R - \Delta R) \cdot \cos(\alpha_n) \tag{7}$$

where ΔR is constant equal to the distance between the center and the wires and α_n is:

$$\begin{cases} \alpha_1 = -\alpha \\ \alpha_2 = 120° - \alpha \\ \alpha_3 = 240° - \alpha \end{cases} \tag{8}$$

Obviously the equations (5) and (6), become the same for $i \to \infty$.

In order to reach the desired shape in a finite time t_f, we should choose the appropriate law for the time variation of the displacements and speed for the three wires, going from the home position to the final position.

For each instant, the wires must be moved in order to avoid elongation or compression of it self.

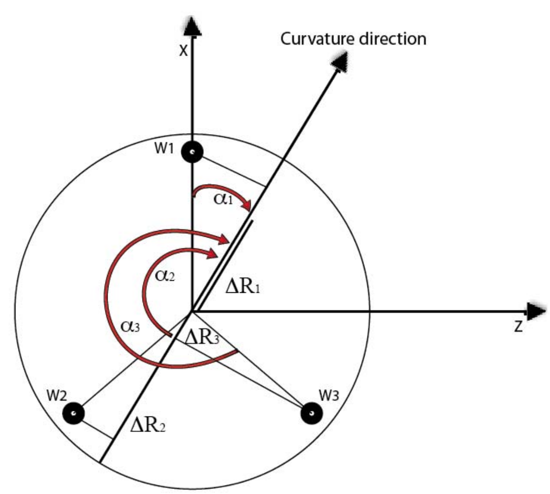

Fig. 5 Projection of the wire to get the α

The reference systems for each segment are oriented with the X-axes passing through the first wire. That means that the angles considered between the wires and the desired directions are as in the equation (8).

We can obtain the correlation between these angles and the bending direction of the segment. E.g. if the direction is

$\alpha = \frac{2}{3}\pi$, that means we intend to bend the tentacle in the direction of the second wire with the imposed value of θ degrees. In this case, if we will move the second wire of ΔL_{w2}, we should move the first and third wires with $\Delta L_{w2}/2$ and with the apropiate speed in order to maintain this relation during the movement.

Once we know the angle α, we can obtain the value $\Delta R_i = \Delta R \cdot \cos(\alpha_i)$, defining the displacements of the wires.

The algorithm that we are using, assigns the speed of the wires proportional to ΔR_i in order to go from the home position $(\theta=0,\ \alpha=0)$ to the position $(\alpha,\theta)_i$ with a constant speed of the motors.

In fact, given the final time t_f and the starting time t_i, after we obtained the displacement of the wires we impose the speed in order to reach the desired position in $(t_f\text{-}t_i)$ seconds.

So the speed is:

$$\dot{L}_{wi} = \frac{L_{wi}(t_f) - \overline{L}_{CB}}{(t_f - t_i)} \tag{9}$$

B. Reverse kinematic of the wire

Our structure does not have encoders. Counting the impulses given to the motors, we can evaluate the lengths $[L_{w1}, L_{w2}, L_{w3},]$. We use these values in order to obtain $(\alpha,\theta)_i$. The algorithm's steps are the following.

For the n-th rigid wire:

$$L_{wn} = \overline{L}_{CB} - \theta \cdot \Delta R \cdot \cos(\alpha_n) \tag{10}$$

Considering the equation (8) and (10), evaluating these for all the wires we can obtain:

$$\begin{cases} \sum_{i=1}^{3} \cos(\alpha_i) = 0 \\ \frac{1}{3}\sum_{i=1}^{3} R_i = R \\ \frac{1}{3}\sum_{i=1}^{3} Lw_i = L \end{cases} \tag{11}$$

Considering again the equation (10) for the first and second wires, we can write:

$$Lw_1 + \Delta R \cdot \theta \cdot \cos(\alpha_1) = Lw_2 + \Delta R \cdot \theta \cdot \cos(\alpha_2) \tag{12}$$

Replacing the (8) we obtain θ in function of α:

$$\theta = \frac{2}{\Delta R} \cdot \frac{Lw_1 - Lw_2}{3\cos(\alpha) - \sqrt{3}\sin(\alpha)} \tag{13}$$

And considering the eq. (10) for the third wire:

$$Lw_3 = Lw_1 + \frac{2 \cdot (Lw_1 - Lw_2) \cdot \left(3\cos(\alpha) - \sqrt{3}\sin(\alpha)\right)}{3\cos(\alpha) - \sqrt{3}\sin(\alpha)} \tag{14}$$

Finally the α angle can be obtained using the function atan2.

$$\alpha = \mathrm{atan2}\left(\sqrt{3}(Lw_2 - Lw_3), 2Lw_1 - Lw_2 - Lw_3\right) \tag{15}$$

where atan2 is an extension of arctan(y/x) on more quadrant having the following form:

$$\begin{cases} \mathrm{atan}(y/x) + \pi & if \quad x < 0, y \geq 0 \\ \mathrm{atan}(y/x) - \pi & if \quad x < 0, y < 0 \\ \mathrm{atan}(y/x) & if \quad x > 0 \\ \frac{\pi}{2} & if \quad x = 0, y > 0 \\ \frac{-\pi}{2} & if \quad x = 0, y < 0 \end{cases} \tag{16}$$

IV. Curvature Of One Segment of tronconical robot

The same methodology of the section III can be applied for a tronconical robot. The following paragraphs will show how the equations change.

A. Direct kinematic of the wires

The geometry of one segment for the 2D case is described in Fig. 6. The curvature's angle θ of the segment is considered as the input parameter, while the lengths L1 and L2 of the control wires are the outputs.

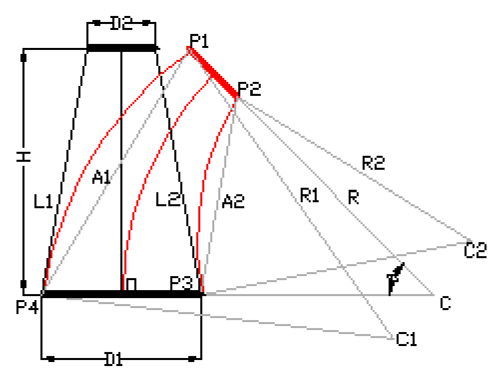

Fig. 6. The geometry of one segment.

The radius R of the segment curvature is obtained using equation (17):

$$R = \frac{H}{\theta} \tag{17}$$

where H is the height of the segment. The following lengths are obtained from Fig. 5, based on the segment curvature:

$$L_{11} = \overline{CP_4} = R + \frac{D_1}{2} \quad L_{12} = \overline{CP_1} = R + \frac{D_2}{2}$$
$$L_{21} = \overline{CP_3} = R - \frac{D_1}{2} \quad L_{22} = \overline{CP_2} = R - \frac{D_2}{2} \tag{18}$$

where D_1 and D_2 are the diameters of the segment end discs. Based on the Carnot theorem, the lengths *A1* and *A2* are then obtained:

$$A_1 = \sqrt{L_{11}^2 + L_{12}^2 - 2 \cdot L_{11}^2 \cdot L_{12}^2 \cdot \cos\theta}$$
$$A_2 = \sqrt{L_{21}^2 + L_{22}^2 - 2 \cdot L_{21}^2 \cdot L_{22}^2 \cdot \cos\theta} \tag{19}$$

The control wires curvature radius R_1 and R_2 are given by the relations (20):

$$R_1 = \frac{A_1}{2 \cdot \sin\frac{\theta}{2}} \quad R_2 = \frac{A_2}{2 \cdot \sin\frac{\theta}{2}} \tag{20}$$

Finally, the lengths of the control wires are obtained as in (21):

$$L_{w1} = R_1 \cdot \theta = \frac{A_1 \cdot \theta}{2 \cdot \sin\frac{\theta}{2}}$$
$$L_{w2} = R_2 \cdot \theta = \frac{A_2 \cdot \theta}{2 \cdot \sin\frac{\theta}{2}} \tag{21}$$

For the 3D case, a virtual wire is considered, which gives the α direction of the curvature. Considering one virtual wire in the direction of the desired curvature having length calculated as follows. Firstly the following lengths are computed:

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

where α_n is according to Fig. 5:

$$\begin{cases} \alpha_1 &= -\alpha \\ \alpha_2 &= 120° - \alpha \\ \alpha_3 &= 240° - \alpha \end{cases} \tag{23}$$

Based on (19) and (20) the curvature radiuses R1, R2 and R3 of the three control wires are then obtained. Finally the lengths of the control wires are computed with (24):

$$L_{w1} = R_1 \cdot \theta$$
$$L_{w2} = R_2 \cdot \theta \tag{24}$$
$$L_{w3} = R_3 \cdot \theta$$

Apart from the system presented we can obtain two useful relations:

$$\begin{cases} \sum_{i=1}^{3} \cos(\alpha_i) &= 0 \\ \frac{1}{3}\sum_{i=1}^{3} Lw_i &= L \end{cases} \tag{25}$$

The second equation of (25), can be utilized to estimate the virtual compression or the extension of the central bone. We call that virtual compression because before we compress the central bone, the robot will twist to find the shape to guaranty the wrong length of the wires.

V. Actuate the Tentacle Arm

A. Hardware architecture

The experimental mechanical structure consists in a three tentacle segment, resulting in a total length of more than one meter. Each segment's shape is controlled via three cables, actuated by three stepper motors, resulting in a 2 degree of freedom per element: 2 rotations around OX and OZ axis. The stepper motors are Berger Lahr, capable of 5000 steps per rotation. The rotation movement is converted in the translation needed for varying the length of the cables, by a roto-translation mechanism. The screw step is 1.25 mm, resulting in a 0,00025 mm displacement for each step made by motor. The step motors are computer controlled, and the controllers Berger Lahr SD3 generate the command.

B. Software architecture

The robot's control system consists in one computer running Windows XP and the Berger Lahr controllers interfaced by 3 modules Ad-link PCI8144. The main elements forming the software control architecture are the Visual C++ programming environment and the command functions included in the .DLL libraries provided by AD-Link. A program was developed in order to generate the displacement commands; this program computes the cable length variation in order to obtain a desired curvature and orientation, determines the equivalent number of motor steps, and transmits the command to the controllers, using the functions provided by Berger Lahr DLL's. This command scheme is represented in Fig. 6.

C. Moving the robot with constant angular speed

The problem. Once we know α, in order to obtain a constant speed $\dot{\theta}$, we can impose:

$$\dot{L}_{wn} = \frac{d}{dt}\left[(R - \Delta R_n) \cdot \theta\right] = \text{Constant} \tag{26}$$

Or:

$$\dot{L}_{wn} = \frac{d}{dt}\left\{\left[(R + \Delta R_n) \cdot \theta\right] \cdot \frac{\sin\left(\frac{\theta}{i}\right)}{\frac{\theta}{i}}\right\} = \text{const} \tag{27}$$

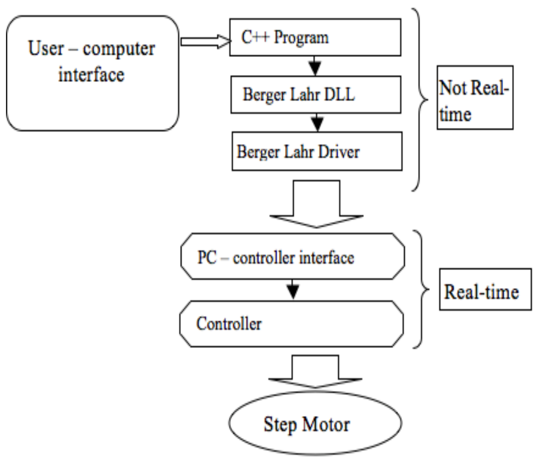

Fig. 6 Flow chart of the motion

Considering α fixed, the time dependence of θ and R, can be written:

$$R(t)=\frac{\bar{L}_{CB}}{\theta(t)} \tag{28}$$

$$\dot{L}_{wn}=\frac{d}{dt}\left[\left(\frac{\bar{L}_{CB}}{\theta(t)}-\Delta R\cdot cos(\alpha)\right)\cdot\theta(t)\right]=Constant \tag{29}$$

$$\dot{L}_{wn}=\frac{d}{dt}\left[\bar{L}_{CB}-\Delta R\cdot cos(\alpha)\cdot\theta(t)\right]=Constant \tag{30}$$

So, imposing $L_{wn}(t)$ constant also $\theta(t)$ will be constant.

Now considering θ fixed, we want to obtain a constant angular speed $\dot{\alpha}$.

$$\dot{L}_{wn}=\frac{d}{dt}\left[\bar{L}_{CB}-\Delta R\cdot cos(\alpha(t))\cdot\theta\right] \tag{31}$$

It implies that it is needed to change the speed of the wires continuously.

The most complex problem is to consider *(α(t), θ(t))* both variable in the time [4].

The solution: For a typical movement of a 3-links robot we can suppose to have two points in the state space *X(t)*:

- Considering the starting shape Xi and the final Xf (in radians), time t_i and t_f(in seconds):

$$\left[\theta_1^i,\alpha_1^i,\theta_2^i,\alpha_2^i,\theta_3^i,\alpha_3^i\right]=X_i \tag{32}$$

$$\left[\theta_1^f,\alpha_1^f,\theta_2^f,\alpha_2^f,\theta_3^f,\alpha_3^f\right]=X_f \tag{33}$$

When we want to move the tentacle from the starting shape to the final shape within $[t_f - t_i]$ seconds, we can divide the trajectory in $k-$steps such that $\left\|X_i - X_f\right\|^2 \le \overline{K_{steps}}^2$, with $\overline{K_{steps}}$ constant chosen.

Taking into account the position of the tentacle, the motors speeds are computed at every beginning of the step. The speeds will be constant for the entire step. This approximation introduces some errors, as twisting in the structure, because if α change continuously even the speed of the wires should change in the same way. Solving this problem it's impossible without a real-time control system.

D. Motion control test

The tests provide information about the repeatability and oscillation of the structure during the movement obtained by using the implemented algorithm. The first test is dedicated to the study of the repeatability and it is performed by imposing a fixed angle to the tentacle segments and moving each segment for several times. Finally an evaluation of the positioning errors is performed. The results obtained are plotted and listed in the following figure and table.

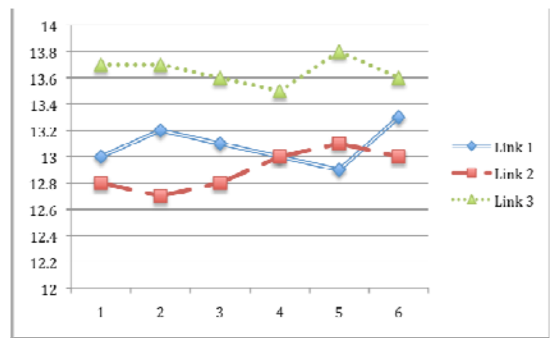

Fig. 8. Repeatability graphics for the 3 links.

Link Number	Error average	Error variance
1	0,08333 deg	0,02166 deg
2	0,1 deg	0,024 deg
3	0,65 deg	0,011 deg

Tab. 1. Average and variance of the errors on the links.

Going from link 1 to the link 3, the error grows because there is some interaction between the segments, in the same time the variance is compensated from the length of the wires.

The second test was designed to measure the structure's vibration introduced by our control algorithm. A high-speed video camera is placed in front of the robot and acquires images during the tentacle movements (125 fps, target object size 14 mm). Evaluating the images, for more than 500 frames, we obtained the following results plotted in Fig 8.

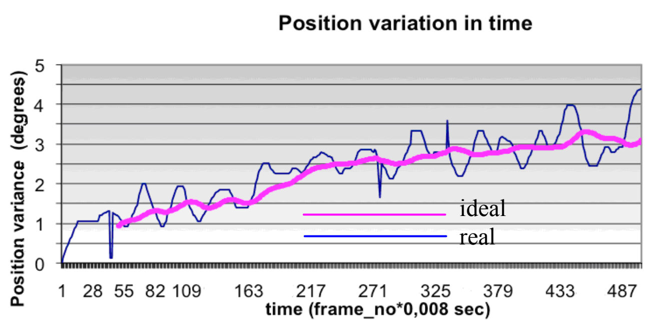

Fig. 8. Graphics of the vibration during the movement.

Taking out the average drift, which is given by the movement of the robot during the measurement we have the pure oscillation:

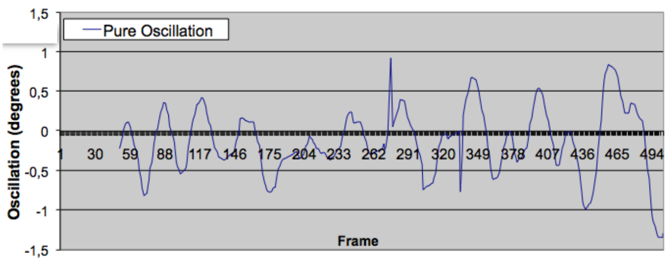

Fig. 9. Graphics of the pure oscillation of the structure.

Moving the robot on the shortest path in the state space X, we should change the speed continuously. Working in Windows, which is not a real-time system, this task is not a possible one. We imposed a constant speed for each step. This solution introduced an error because the robot's segments twisted.

One of the possible solutions for this problem, working under an operating system without a real real-time support, could be to find a trajectory in the state-space X which corresponds to a constant speed in the wires space.

E. Constant speed in the wire space for the conical model

We propose another class of motion control to solve the vibration problem showed in section D. It is not possible to follow every trajectory in the state space, maintaining the wires' constant speed. We can consider the trajectory in the state space divided in small steps. Between the steps, we need to avoid the structure's vibration without taking into consideration the trajectory. Considering the small steps, for a typical movement of a 3 links robot we consider two points in the joint space *X(t)* as follows:

- the starting shape of the little step (in radians) at time t_0 (in seconds):

$$\left[q_1^0, q_2^0, q_3^0\right] = X^0 \tag{34}$$

- the final shape at time t_f (in seconds):

$$\left[q_1^f, q_2^f, q_3^f,\right] = X^f \tag{35}$$

We also consider that $q_n^i = (\alpha_n^i, \theta_n^i)$ are the joint variable of the n-th segment (in radians) at the time i (in seconds).

Hypothesis 1: We impose the constant speed of the motors during the movement in the small steps.

The vibrations in the structure are given by the structure twisting; this twisting is given from the incompatible lengths of the three wires of the segment. In order to avoid the structure vibration we want to move the robot from the starting position to the final position maintaining constant speeds of the wires and maintaining valid (25). We can analyze the direct kinematic as time function writing:

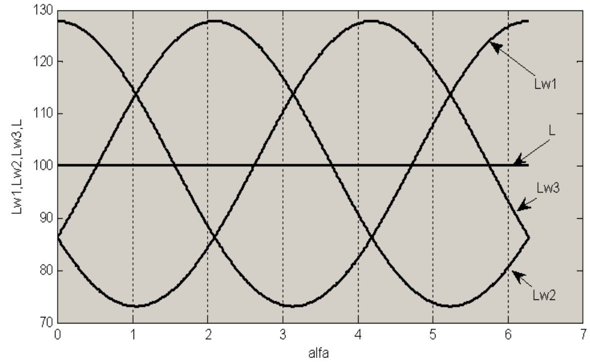

Fig. 10. Variation of the wires when α changes from 0° to 360°.

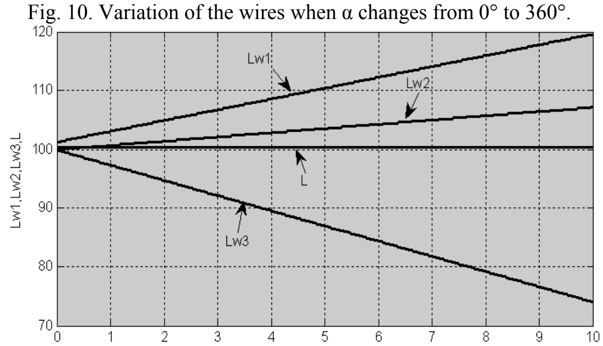

Fig. 11. Time evolution of the wire lengths without delays.

$$L_w(q(t)) = F(q(t)) \tag{36}$$

where *F(.)* represents the direct kinematic in (24).

In order to simplify the calculus in this paper, we can consider the following hypothesis.

Hypothesis 2: The structure is cylindrical D_1=D_2.

Under this hypothesis

$$R_n = R - \Delta R \cdot \cos(\alpha_n) \tag{37}$$

Considering $t_0=0$ and $t_f=\delta t$ given, we can calculate the initial and final wires positions.

$$L^0{}_w = F(q(t_0)) \tag{38}$$

$$L^f{}_w = F(q(\delta t)) \tag{39}$$

which is a column vector of three elements.

From this we can compute the constant speed vector *C*.

$$C = \frac{L^f{}_w - L^0{}_w}{\delta t} \tag{40}$$

We are going to consider a small movement during δt.

$$L_w(q(t+\delta t)) = F(q(t)) + J_F(q(t)) \cdot \dot{q}(t) \cdot \delta t \tag{41}$$

where J_F is the Jacobian of *F* and where:

$$J_F(q(t)) \cdot \dot{q}(t) = C \tag{42}$$

Considering (11) we can compute the average length of the wires, during the small step.

$$\sum_i L_{wi}(q(t+\delta t)) = \sum_i (L_{wi}q(t) + C_i\delta t) \qquad (43)$$

where C_i is the i-th component of the column vector representing the constant wire's speed.

$$\begin{aligned}\sum_i L_{wi}q(t) + C_i\delta t = \theta(t)\cdot\Delta R_i\cdot\cos\alpha_i(t) + L\\ + L - L - \theta(t+\delta t)\cdot\Delta R_i\cdot\cos\alpha_i(t+\delta t)\end{aligned} \qquad (44)$$

Regrouping θ(t+δt) and θ(t) and using (11) we obtain:

$$\sum_i L_{wi}(q(t+\delta t)) = L \qquad (45)$$

So during this movement there is no compression and no induced vibration. If the average of the wires is different from L in fact, the solution we have is not the wanted solution. A "wrong" solution might introduce some structure twisting and during the movement them can became a rotational vibration of the structure. The proof can be extended to the tronconic structure with $D_1 \neq D_2$. Besides, in (45) there is no more dependence of L to δt. This means that, if the starting and the final points are solution of (36), moving with constant speed through the two points, we don't compress the structure. In the state space, for example, choosing a constant speed of the wire means move (α,θ) of the first segment as showed in (46).

$$\begin{cases}\dot{\alpha} = \dfrac{C_1\cdot\sin(\alpha)\cdot\theta}{\Delta R\cdot(\sin^2(\alpha))\cdot\theta^2\cdot\cos^2(\alpha))}\\ \dot{\theta} = -\dfrac{C_1\cdot\cos(\alpha)}{\Delta R\cdot(\sin^2(\alpha))\cdot\theta^2\cdot\cos^2(\alpha))}\end{cases} \qquad (46)$$

This can be obtained by solving (42) for $q(t)$:

$$J_F^{-1}(q(t))\cdot C = \dot{q}(t) \qquad (47)$$

VI. Problems and Future Work

A machine vision system was designed and implemented [10], [13]. The algorithm provides all the information we need to describe the shape of a hyper-redundant system with only 2 cameras. This system will be used in order to determinate the real shape of the tentacle and his behavior. Our future goal is to assemble the visual system together with the control of the robot and close the loop studying several types of controls.

VII. Conclusion

In this paper we presented the experimental tentacle structure designed and implemented at the Department of Mechatronics, University of Craiova, Romania. We focused only to the actuation system of this robot. From this point of view, we proposed a direct and reverse kinematical model, and we developed a series of repeatability tests to validate the model and to determine the model error and the positioning error of the system. Acting to achieve this objective, we encountered some practical problems as shown upper in this paper. We proposed a solution for this problem and we performed a series of tests to validate the model and 2 different algorithms to move the structure [5]. [6].

As was considered from the beginning, the solution for determining the real shape and behavior of the tentacle is a machine vision system. This system is designed and implemented, but was not presented in this paper [9], [12].

References

[1] M. Blessing and I.D. Walker, Novel Continuum Robots with Variable-Length Sections, *Proceedings 3rd IFAC Symposium on Mechatronic Systems*, Sydney, Australia, September 2004, pp. 55-60.
[2] R. Buckingham and A. Graham. Reaching the unreachable – snake arm robots. *International Symposium of Robotics*. Available via: OCRobotics Ltd. www.ocrobotics.com.
[3] R. Cieslak and A. Morecki, Elephant Trunk Type Elastic Manipultor – A Tool for Bulk and Liquid Type Materials Transportation, *Robotica*, Vol. 17, pp. 11-16, 1999.
[4] D. Cojocaru, R.T Tanasie, D.B. Marghitu, D.B., 2008, A Complex Mathematical Expression Operations Tool, *Annals of The University of Craiova, Series: Automation, Computers, Electronics And Mechatronics*, ISSN: 1841-0626, Volume 5(32), no. 1, pp. 7-12.
[5] D. Cojocaru, R.T. Tanasie, A Software System for Virtual Servoing, *WSEAS Transactions on Information Science and Applications*, Issue 11, Volume 3, 2006, p2086-2093, ISSN 1790-0832.
[6] D. Cojocaru, R.T. Tanasie, M. Ivanescu, Graphic Simulator for the Visual Control of Hyperredundant Robots, *WSEAS Transactions on Systems and Control*, Issue 9, Volume 3, 2008, p759-768, September 2008, ISSN 1991-8763
[7] S. Hirose, *Biologically Inspired Robots*. Oxford University Press, 1993.
[8] G. Immega and K. Antonelli, The KSI Tentacle Manipulator. *Proc. 1995 IEEE Conf. on Robotics and Automation*, pp. 3149-3154.
[9] M. Ivănescu, M., D. Cojocaru, N., Popescu, D., Popescu, R.T. Tănasie, Hyperredundant Robot Control by Visual Servoing, *Studies in Informatics and Control Journal*, Volume 15, Number 1, p93-102, ISSN 1220-1766, 2006.
[10] M. Ivănescu, D. Cojocaru, N. Popescu, D. Popescu, R.T. Tănasie, Visual Based Control System for a Tentacle Manipulator, *WSEAS Transactions on Information Science and Applications*, Issue 3, Volume 3, 2006, p610-617, ISSN 1790-0832.
[11] K. Suzumori, S. Iikura, H. Tanaka, Development of Flexible Microactuator and its application to Robotic Mechanisms, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1622-1627, 1991.
[12] R.T., Tanasie, M. Ivănescu, D. Cojocaru, Camera Positioning and Orienting for Hyperredundant Robots Visual Servoing Applications, *Journal of Control Engineering and Applied Informatics*, Vol 11, No 1 (2009), p19-26, ISSN 1454-8658
[13] R.T. Tanasie, D. Cojocaru, Software Techniques for a Visual Servoing Architecture, *WSEAS Transactions on Computers*, Issue 11, Volume 5, 2006, p2682-2690, ISSN 1109-2750.
[14] I.D. Walkera, Darren M. Dawsona, a.o. *Continuum Robot Arms Inspired by Cephalopods*, DARPA Contr. N66001-C-8043, 2005.
[15] I.D. Walker and Carlos Carreras, Extension versus Bending for Continuum Robots, *International Journal of Advanced Robotic Systems*, Vol. 3, No.2, 2006, ISSN 1729-8806, pp. 171-178.