2011 4th International Conference on Mechatronics (ICOM), 17-19 May 2011, Kuala Lumpur, Malaysia

A New Geometrical Approach to Solve Inverse kinematics of Hyper Redundant Robots with variable link length

Annisa Jamali, Raisuddin Khan and Md. Mozasser Rahman
Department of Mechatronics Engineering, Faculty of Engineering
International Islamic University Malaysia, Jalan Gombak, 53100 Kuala Lumpur, Malaysia
raisuddin@iium.edu.my

***Abstract*— In this paper a new approach that generates a general algorithm for n-link hyper-redundant robot is presented. This method uses repetitively the basic inverse kinematics solution of a 2- link robot on some virtual links, where the virtual links are defined following some geometric proposition. Thus, it eliminates the mathematical complexity in computing inverse kinematics solution of n-link hyper redundant robot. Further, this approach can handle planar manipulator with variable links eliminating singularity. Numerical simulations for planar hyper redundant models are presented in order to illustrate the competency of the model.**

Keywords- Robot; Hyper-redundant; Variable length links; Inverse kinematics.

I. INTRODUCTION

Technology evolvements arise in parallel to the increasing number of robotics application. A robot having more degrees of freedom (DOF) than are necessary to perform a specific task is referred to as kinematically redundant or hyper-redundant robot [1]. Even though existence of redundancy complicates the solution process, redundant degree of freedom has been recognized as a mean to improve manipulator performance in complex and unstructured environments tremendously. Idea of hyper redundant manipulator is promising due to their ability and applicability in many complicated scenario such as avoidance of obstacles, easy access into cluttered and inaccessible workspace, reconfigurable in the case of failure of few joints, etc. Thus, generating algorithm for hyper-redundant manipulator to handle aforementioned complex scenario is one of the interests of the today's researchers.

In conventional robot control theory, the manipulator Jacobian is considered as one of the most important tools [2]. Earlier in 1980's, research on inverse kinematics problem of hyper redundant robot was focused on Jacobian pseudo inverse approach and it's optimization [2][3]. At present, there are numbers of algorithms for the redundant manipulator inverse kinematics. The algorithm formulation can be divided mainly into three categories [4] that is (i) the algebraic approach [5]. (ii) the iterative approach that include Artificial Neural Network [6], genetics algorithm , ANFIS [7], Fuzzy Logic [8] and (iii) the geometric approach [9][10][11][12][13].

In general, algebraic approach suffers from heavy computational burden and difficult symbolic expansion. Besides, although this approach offers several possible solutions with different arm configuration, but there is no proper indication to choose the best solution. Meanwhile, iterative approach involves high time-consuming computation where it takes considerable time to come up with useful inverse kinematics (IK) solution. Moreover, error minimizing always becomes a subject of investigation which also due to several possible solutions offered by this approach. The geometric approach means finding θ_i using geometrical heuristics to take advantage of the special structure of the manipulator. Thus, geometrical approach offers a direct solution.

This paper proposes a new geometrical approach to solve inverse kinematics of hyper redundant manipulator, where the length of the manipulator is adjustable. The method involves repetitive use of inverse kinematics solution of 2-link manipulator based on some geometric proposition, which depends on the selection of some reference point and virtual configuration. A coil shape is then formed out of the serially connected n-link which allows the system to avoid singularity. The paper is organized as follows; section 2 presents the details of algorithm formulation; section 3 verifies by computer simulation the general algorithm; and section 4 draws conclusions based on the performance of the algorithm.

II. DETAILS OF ALGORITHM FORMULATION

A. Defining Virtual Layers

At the beginning, a 2-link virtual robot is introduced by taking length of the virtual links equal to the sum of lengths of half of the total number of links, n. However, in the case of n an odd number, the virtual link closer to the origin will have even number of links and the other virtual link will have odd number of links. This 2-link virtual robot is then solved for inverse kinematics with either elbow-up or elbow-down configuration, so that the tip of the robotic manipulator reaches the desired position. This makes the 1st virtual layer SR_1. In Fig.1 OO_1P is the virtual layer SR_1formed out of an 8-link robotic manipulator, where P is the desired position and O is the origin. In the next step each of the virtual links of SR_1 is

The authors would like to thank Research Management Centre, International Islamic University Malaysia for funding this research.

978-1-61284-437-4/11/$26.00 ©2011 IEEE

divided in a similar fashion; that is virtual links OO_1 and O_1P are each divided into 2-link sub robots SR_2 and SR_3 respectively, where SR_{21} and SR_{22} are the two links of the sub robot SR_2, and SR_{31} and SR_{32} are the two links of the sub robot SR_3. Thus, number of links for any virtual sub robot, SR_{ij} are;

$$\left.\begin{aligned} n_i &= [n/2, n/2] \\ n_{i+1} &= [n_i/2, n_i/2] \\ &\vdots \\ n_{i+\infty} &= [n_{i+\infty-1}/2, n_{i+\infty-1}/2] \end{aligned}\right\} \quad \text{even link (1)}$$

And

$$\left.\begin{aligned} n_i &= [\lfloor n/2 \rfloor, \lceil n/2 \rceil] \\ n_{i+1} &= [\lfloor n_{ij}/2 \rfloor, \lceil n_{ij}/2 \rceil] \\ &\vdots \\ n_{i+\infty} &= [\lfloor n_{i+\infty-1}/2 \rfloor, \lceil n_{i+\infty-1}/2 \rceil] \end{aligned}\right\} \quad \text{odd link (2)}$$

The virtual sub robots that consist of n-links is continuously going through the same process until the sub robots left with 1 or 2 links respectively at each sub division. Each of the sub robots is generated by summing length of links in each division.

$$\left.\begin{aligned} L_i &= [\sum_{k=1}^{k=n_i} l_k, \sum_{k=n_i+1}^{k=n} l_k] \\ L_{i+1} &= [\sum_{k=1}^{k=n_{i+1}} l_k, \sum_{k=3}^{k=n_i} l_k, \sum_{k=5}^{k=n_i+n_{i+1}} l_k, \sum_{k=7}^{k=n} l_k] \\ &\vdots \\ L_{i+\infty} &= [\sum_{k=1}^{k=n_{i+\infty}} l_k, \sum_{k=1}^{k=n_{i+\infty-1}} l_k, \cdots\cdots\cdots\cdots] \end{aligned}\right\} \quad (3)$$

In one system, it is possible to have combination of (1) and (2) to obtain n-number of link in each subrobot. The sequence for number of links can be arranged according to subscript of sub robot (e.g: ij).

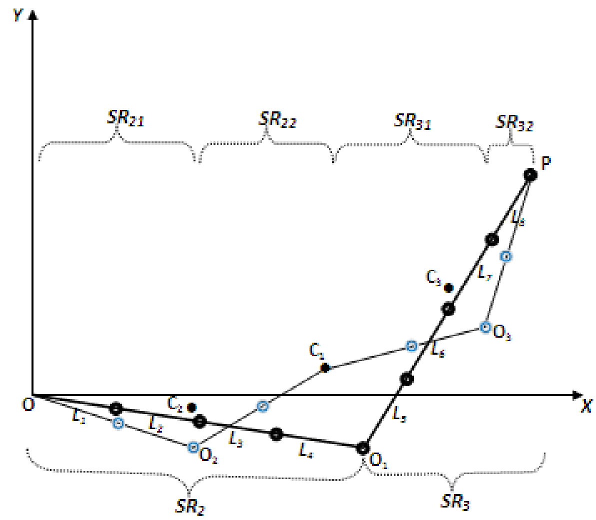

Figure 1. Initial virtual layers of 8-link robot.

B. Algorithm for Inverse Kinematics of Virtual sub robots

The first virtual layer SR_1 forms a virtual triangle OO_1P as shown in Fig.1. In the next step centroid of this triangle, C_1, is determined. Now applying the inverse kinematics solutions of 2-link model respectively on SR_2 and SR_3 considering C_1 respectively the desired position for SR_2 and anchor point for $SR_3$the configurations OO_2C_1 and C_1O_3P are achieved. Configurations OO_2C_1 and C_1O_3P again represents two triangles where C_2 and C_3 are their respective centroids. In general for a given triangle with three corners; (x_1, y_1), (x_2, y_2) and (x_3, y_3) centroid is given by the formula.

$$C_n = [x_n, y_n] \qquad (4)$$

$$x_n = \frac{(x_1 + x_2 + x_3)}{3}, y_n = \frac{(y_1 + y_2 + y_3)}{3} \qquad (5)$$

Repeating the two links inverse kinematics on the sub robots SR_{21} and SR_{22} respectively in a similar way using C_2 as the end point for SR_{21} and anchor point for SR_{22} we get the inverse kinematics for the 1st four links. The same process when applied to the sub robots SR_{31} and SR_{32} we get the inverse kinematics of the last four links. Thus the final virtual layer is achieved which is shown in Fig.2. This configuration may be accepted as the inverse kinematics solution of the multi-link robot; however, this is quite zigzag. Addition of few more steps on this algorithm helps eliminate the zigzag configuration. These steps are described in the next section.

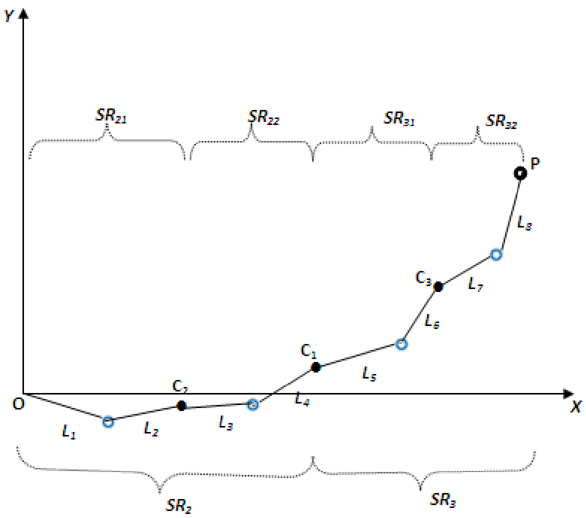

Figure 2. Final virtual layers of 8-link robot

C. Algorithm to form a coil shape out of the zigzag shape

It is evident from Fig.2 that the zigzag configuration is due to elbow down configurations of the links adjacent to the virtual centroids. One such centroid, C, surrounded by adjacent links is shown in Fig. 3. In this figure the virtual links CE (l_2) and CF (l_3) are elbow down. If now the point C is pushed diagonally opposite to the line EF, the new configuration will be EAF, i.e, links l_2 and l_3 become elbow up along with link l_1 and l_4. In order to achieve this transformation following calculations are done.

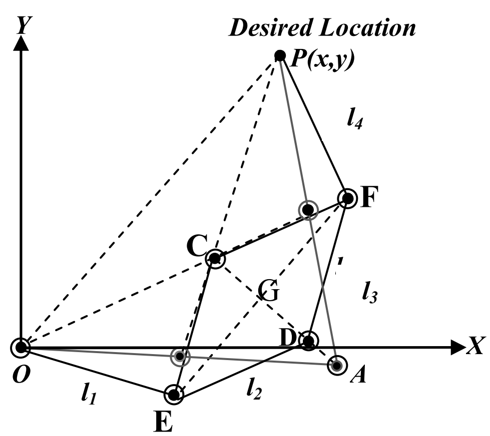

Figure 3. Geometry showing steps of converting 4 links zigzag virtual configuration to coil shape configuration.

Referring to Fig. 3, let length of EG is 'a' and length of EF is 'b', where G is the intersection of the perpendicular diagonals CD and EF of the parallelogram AECF. Thus lengths a and b can be expressed by the equations 6 and 7.

$$a = \sqrt{(x_C - x_E)^2 + (y_C - y_E)^2} \quad (6)$$

$$b = \sqrt{(x_F - x_E)^2 + (y_F - y_E)^2} \quad (7)$$

In the next step the angle, θ between CE and EG is calculated using equations 8-10.

$$\overline{EF} = (x_F - x_E)\hat{i} + (y_F - y_E)\hat{j} \quad (8)$$

$$\overline{EC} = (x_C - x_E)\hat{i} + (y_C - y_E)\hat{j} \quad (9)$$

$$\cos\theta = \overline{EF}.\overline{EC} / (|\overline{EF}||\overline{EC}|) \quad (10)$$

From there, c , the length of EG is derived as follows.

$$c = a\,(\cos\theta) \quad (11)$$

By using the ratio between 'c' and 'b', the point G, i.e., the intersection of *EF* and *CD* is calculated as follows:

$$\frac{c}{b} = \frac{x_G - x_E}{x_F - x_E}, \; \frac{c}{b} = \frac{y_G - y_E}{y_F - y_E} \quad (12)$$

$$x_G = \frac{c}{b}(x_F - x_E) + x_E, \; y_G = \frac{c}{b}(y_F - y_E) + y_E \quad (13)$$

Now the new position of C for elbow up configuration becomes D. The coordinates of D become:

$$x_D = 2x_G - x_C, \; y_D = 2y_G - y_C \quad (14)$$

In order to ensure links acquire a nice curve, checking of elbow up and elbow down is important. It can be determined by computing the slopes of the lines *EC* and *ED* in Fig. 3. The slopes of lines *EC* and *ED* are:

$$m_{EC} = \frac{y_C - y_E}{x_C - x_E}, \; m_{ED} = \frac{y_D - y_E}{x_D - x_E} \quad (15)$$

If $m_{EC} > m_{ED}$, the elbow up will take place while $m_{EC} < m_{ED}$ the elbow down will happen.

While applying the above algorithms for inverse kinematics solution it needs some adjustments in the case of subrobots consisting of 3 links, which consequently leads to a virtual layer of a two link robot, where one virtual link consists of two links and the other is a single link. However, for sub robots consisting of 4 links, it exactly follows the above algorithms. Example of these two cases are presented in the following simulation section.

III. SIMULATION

In this section, MatLab simulations are carried out on 2D hyper-redundant manipulators.

A. Case One (Equal Length Links and even numbers):

An 8-links hyper redundant manipulator is simulated as following. Let us assume desired position will be P= (0.4, 0.35) and length of the links are same say $l_1 = l_2 = l_3 = l_4 = l_5 = l_6 = l_7 = l_8 =$ 0.1m. By solving equation (1) and (3) we will have following results.

$$n_1 = [8/2, 8/2] = [4,4]$$
$$n_2 = [4/2, 4/2] = [2,2],[2,2]$$
$$L_1 = [\sum_{k=1}^{k=4} l_4],[\sum_{k=4}^{k=8} l_4] = [0.4, 0.4]$$
$$L_2 = [\sum_{k=1}^{k=2} l_k, \sum_{k=3}^{k=4} l_k, \sum_{k=5}^{k=6} l_k, \sum_{k=7}^{k=8} l_k] = [0.2, 0.2, 0.2, 0.2]$$

By solving the equations (4) until (14), one single value of θ for each link will be identified which can give a coil shape links. The end-effector final position is P= (0.4, 0.35) as per desired. For this particular case, the algorithms developed work without any adjustment. The results of the simulation for elbow up and elbow down configuration are presented in Figure 4 and Figure 5 respectively.

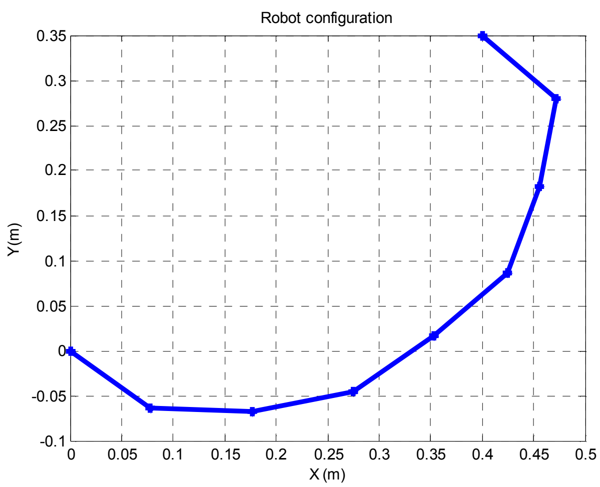

Figure 4. Inverse Kinematics of a 8-Link robot for elbow up configuration.

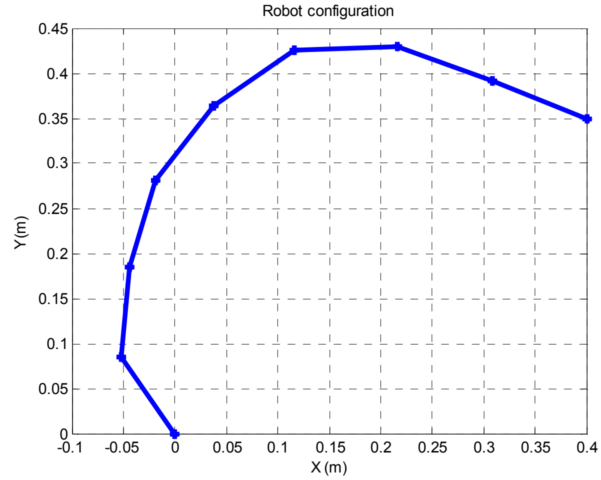

Figure 5. Inverse Kinematics of a 8-Link robot for elbow down configuration.

B. Case Two (Different Length Link with odd numbers):

A planar manipulator with seven links is selected as a second study case. Let l_1=0.12, l_2=0.09, l_3=0.08, l_4=0.10, l_5= 0.13 and l_6= 0.07 where the lengths are in meter. The target point is P=[0.25,0.3]. Equation (1), (2) and (3) are applied to obtain number of links and lengths of links for each division. In this case, both equation (1) and (2) are used because the system involves odd and even number of links at different division. The following results are obtained.

$$n_1 = [6/2, 6/2] = [3,3]$$
$$n_2 = [\lceil 3/2 \rceil, \lceil 3/2 \rceil] = [2,1],[2,1]$$
$$L_1 = [\sum_{k=1}^{k=3} l_k],[\sum_{k=4}^{k=6} l_k] = [0.29, 0.3]$$
$$L_2 = [\sum_{k=1}^{k=2} l_k, \sum_{k=3}^{k=3} l_k, \sum_{k=4}^{k=5} l_k, \sum_{k=6}^{k=6} l_k] = [0.21, 0.08, 0.23, 0.07]$$

Then, if equations (1) until (5) are applied, one single value of θ for each link will be identified for the configuration OO_2O_3P as shown in Figure 6. However, the layers SR_{21} and SR_{22} now consists of 2 and 1 link respectively, as such pushing O_2 to C_2 keeping C_1 as the anchor point will not work without some adjustment in the algorithm. Required adjustment is described below.

C. *Adjustment in the algorithm for odd number of links layer*

In Figure 6 the point C2 is adjusted extending C_1C_2 until C'_2, such that $C_1 C'_2 = L_3$. Then the point M is determined using the 2 links inverse kinematic solution for links 1 and 2. Similar approach is applied for links 4,5 and 6. The coil shaping algorithm is then applied as usual using equations 6 to 15. Final inverse kinematics of a six link robot is shown in Figure 7.

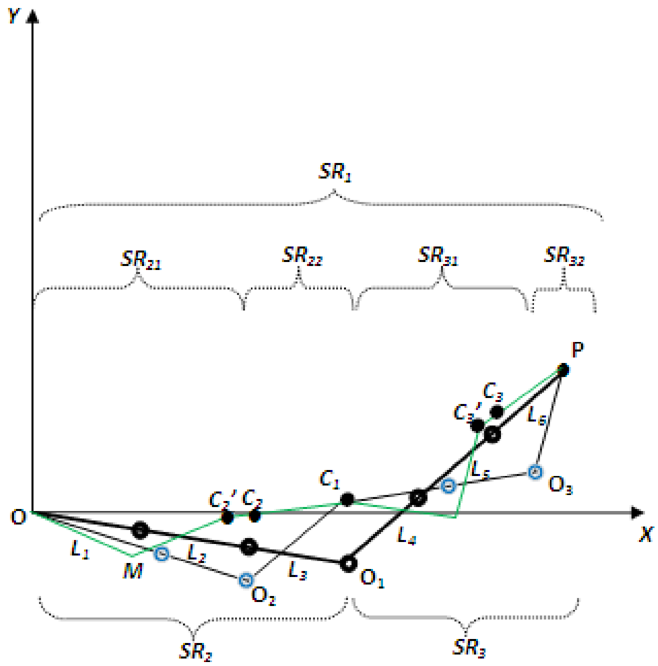

Figure 6. Adjustment scheme for odd number of links.

Variable length of link gives flexibility in picking the proper combination of hyper-redundant manipulator to accomplish desired task. Thus, in the case of a robot of many links few of the links could be locked to be considered as a single link. Such an action will make the robot a variable length link robot even though all the links are of same length originally. However, such an action will help reduce actuation of few actuators and thus make the operation faster as well as economic.

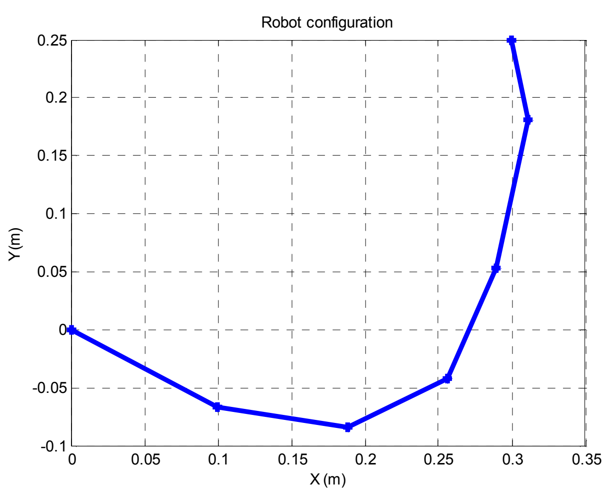

Figure 7. Inverse Kinematics of a 6-Link variable length robot for elbow up configuration.

IV. 5. CONCLUSION

In this paper, a new approach of calculating the inverse kinematics of *n* planar redundant manipulators is presented. The proposed method will generate a single solution with an option of elbow up and elbow down. Thus, the problem of finding one solution among the infinite solutions caused by redundancy is avoided. A simple mathematical computation is one of the advantages of this approach. Besides, the method introduces a sequence that will avoid singularity to occur in obtaining angles for each joint link. The ability of this approach to handle length link variability can increase efficiency in the real application. Simulations on the algorithm are carried out in this paper to check its' competency.

REFERENCES

[1] G. Antonelli, "Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems",. In the proceedings of IEEE International Conference on Robotics and Automation (ICRA 2008), pp: 1993-1998.

[2] H. Mochiyama, H. Kobayashi, The Shape Jacobian of a Manipulator with Hyper Degrees of Freedom, Proceedings of the 1999 IEEE International Conference on Robotics & Automation Detroit, Michigan May 1999

[3] J. Baillieul, "Avoiding Obstacles and Resolving Kinematic Redundancy", Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, 1986, pp. 1698-1704.

[4] L. Sheng, W. Yiqing, C. Qingwei and H. Weili, 2006. A new Geometrical Method for the Inverse Kinematics of the Hyper-Redundant Manipulators. In the Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO '06), Kunming, China, pp: 1356-135

[5] Chirikjian ,G. S. and Burdick, J. W. (1990). "An obstacle avoidance algorithm for hyper-redundant manipulator", Proc. of IEEE Conf. on Robotics and Automation (Cincinnati, OH), Vol.1, pp. 625-631.

[6] Mayorga, R. V., &, Aduthaya, T.C.N (2008), "An Artificial Neural Network approach for the obstacle avoidance of redundant robot manipulators", Journal of Intelligent Robots and System, pp. 4184 – 4184.

[7] S. Alavandar, M. J. Nigam (2008). Inverse Kinematic Solution of 3DOF Planar Robot using ANFIS. In: Int. Jour. of Computers, Communications & Control, Vol. III, pp. 150-155.

[8] D. W. Howard, & A. Zilouchian, "Application of Fuzzy Logic for the solution of Inverse Kinematics and Hierarchical Controls of Robotic Manipulators". In: Jour. of Intelligent and Robotic Systems, Vol.23, Issues.2-4, pp. 217-247, 1998.

[9] F. M. Ashrafiuon, & C. Nataraj, "A General Solution for the Position, Velocity and Acceleration of Hyperredundant Planer manipulator". In: Jour. of robotic Systems, Vol.1, Issues1, pp. 1-12, 2002.

[10] M. Ivanescu, N. Popescu, and D. Popescu, "A Variable length Hyperredundant Arm Control System", Proc. of the IEEE International Conference on Mechatronics & Automation, Niagara False, Vol.4, pp.1998-2003, 2005.

[11] W. J. Chung, W. K. Chug & Y. Youm, "Inverse Kinematics of Planar Redundant Manipulators using Virtual Link and Displacement Distribution Schemes". In: Proc. of the IEEE International Conference on Robotics and Automation. Sacramento, California. Vol.1, pp. 926-932, 1991.

[12] H. Ashrafiuon, K. Sanka, "Development of virtual link method for the solution of hyperredundant spatial robots". In: Jour. of robotic Systems, Vol.13, Issues.6, pp. 371–378, 1996.