

# Inverse Kinematics of a 10 DOF modular hyper-redundant robot resorting to exhaustive and error-optimization methods: A comparative study

Mario Sáenz Espinoza\*, José Gonçalves<sup>†\*</sup>, Paulo Leitão\*, José Luis González Sánchez<sup>‡</sup> and Alberto Herreros<sup>‡</sup>

\*Polytechnic Institute of Bragança, Portugal

<sup>†</sup>INESC TEC (formerly INESC Porto), Portugal

<sup>‡</sup>University of Valladolid, Spain

mario.rs.espinoza@alunos.ipb.pt, {goncalves,pleitao}@ipb.pt, {jossan,albher}@eis.uva.es

**Abstract**—This paper describes and compares several approaches applied to compute the inverse kinematics of a 10 degrees of freedom hyper-redundant robot. The proposed approaches are based on an exhaustive method and several error-optimization algorithms. The algorithms' performance was evaluated based on two criteria: computing time and final actuator positioning error. The mentioned hyper-redundant robot was projected to be used in biomedical applications.

## I. INTRODUCTION

Hyper-redundant robots are based on the design principle of more-than-necessary degrees of freedom (DOF) to perform a particular task. The number of “necessary” DOFs needed in a robot depends on the task itself. For example, in a three-dimensional environment, a 6DOF robot (Figure 1) is required to reach a given point with a certain orientation, while a 6DOF+ robot (Figure 2) would be able not only to reach the same point with the same orientation, but could also have the ability to do it redundantly, that is, having the possibility to do it in more than just one way.

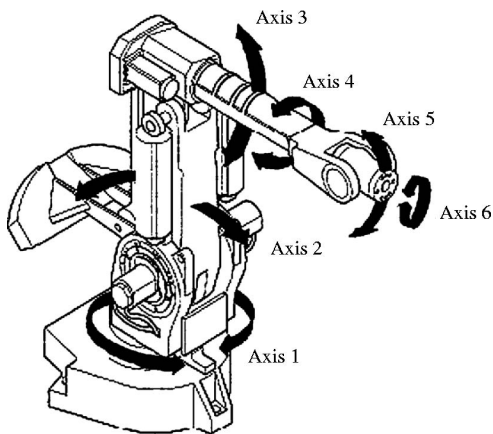


Fig. 1. Typical 6DOF robot manipulator

The hyper-redundant characteristic of this type of robots also provides an advantage for displacement and obsta-

cle avoidance on irregular environments over the common wheeled, tracked or legged robots [1].

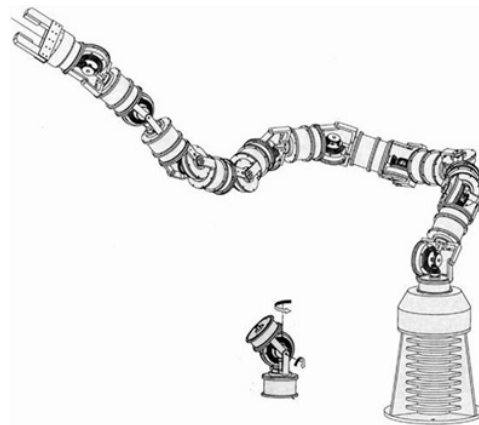


Fig. 2. Hyper-redundant manipulator robot by [2], Georgia Tech.

The hyper-redundant manipulators (*HRMs*) can be used for many tasks like servicing underground tanks [3] (since it is easier for the snake-like robots to inspect or repair any damaged components in this closed underground scenario) or nuclear core reactors [4] (again, due to its flexibility to operate in space-constrained environments). Other applications of *HRMs* include surgical snake-like robots [5] and aeronautical and space exploration [6] [7]. Therefore, it is of great interest to continue studying and developing new methods and applications for these type of robots, since the advantages of moving on irregular environments and extreme maneuverability are appealing, among others, for biomedical applications.

More specifically, one of the most featured applications of hyper-redundant manipulators in the medical field has been endoscopy. Endoscopy (from greek *-endo*: inside and *-scopia*: vision) is the diagnosing technique that uses an instrument to capture images from interiors of humans, animals or other inanimate objects. This allows the physicians to use minimally

invasive methods for screening, diagnosing and sometimes even treatment of certain health threats.

One of the first attempts to incorporate both snake-like robots and endoscopic applications was presented in [8]. A robotic colonoscope was proposed with somewhat success: an earthworm-like robot was built and validated on a lubricated and flexible urethane tube as a model for human intestines. However, the rigidity of this tube was far larger than the one shown on the pigs' intestines on the animal trial phase, making the colonoscope unable to move towards the digestive tube as desired.

More recently a sensor-based guidance control of a continuum robot for a semi-autonomous colonoscope was also proposed [9]; which, again, had somewhat success: it was only tested on tubes and as of 2012 is still lacking of animal/human validation. Nevertheless, these two cases are great examples of both the level of interest and how difficult it is to implement endoscopic functions into autonomous or semi-autonomous *HRMs*.

Regarding the present article, the projected hyper-redundant robot is described on section II. Several approaches to the inverse kinematics are proposed (section III) and also compared (section IV), ideally, as part of a larger project to develop a future hyper-redundant endoscope prototype that could deliver a new platform for health professionals to work with and, even more importantly, that could be translated into health benefits for the patients.

## II. HYPER-REDUNDANT ROBOT

According to [6]: "Proposed tasks for future robotic systems, ranging from space exploration to medical devices, will require robotic devices and components that are simple, robust, lightweight, inexpensive, and easy to control. Hyper-redundant binary systems have been proposed to meet this need".

Trying to meet the characteristics mentioned in the paragraph above, an electromagnetically-controlled 10DOF modular hyper-redundant robot was simulated (see Figure 3).

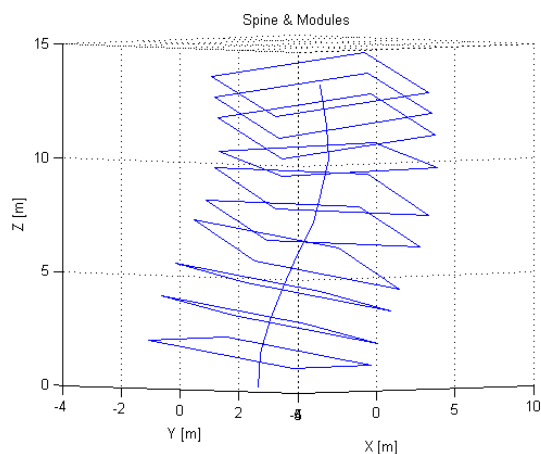


Fig. 3. A 10-module, 10DOF, hyper-redundant robot simulation

Another characteristic met by hyper-redundant robots is that they create an almost continuous workspace, as show in Figure 4. This feature becomes handy when dealing with obstacles, moving through constrained spaces or having to reach difficult points.

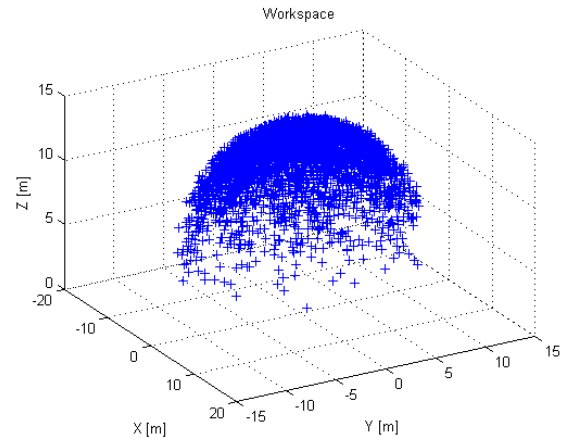


Fig. 4. Workspace created by 4000 random end-points of a 10-module hyper-redundant robot

This also means that hyper-redundant robots can reach points and/or sort obstacles in space in an easier manner, or at least in more than just one way, than DOF-constrained robots. Nevertheless, this also implies that forward kinematics, as well as inverse kinematics, are more complicated to calculate.

According to [4]: "To date, hyper-redundant manipulators have remained largely a laboratory curiosity. There are a number of reasons for this:

- 1) Standard kinematic techniques have not been particularly efficient or well suited to the needs of hyper-redundant robot task modeling.
- 2) The mechanical design and implementation of hyper-redundant robots has been perceived as unnecessarily complex.
- 3) Hyper-redundant robots are not anthropomorphic, and pose difficult programming problems".

Considering all these limitations and keeping in mind a near-future prototype, a *HRM* was designed based on a series of characteristics:

- 1) The *HRM* must be modular, so adding or subtracting modules would be a relatively easy task, allowing the robot to be scalable. It also provides the ability to achieve desired length.
- 2) Electromagnetic couplings were going to be used as actuators, since they are a cost-effective solution for the energy delivery and conservation problem.
- 3) Four electromagnets were to be used in order to allow ten possible positional combinations.

- 4) The simulated model should be able to have the flexibility to operate with different module sizes and minimum between-modules-distance parameters.

However, as mentioned before, these *HRM* characteristics would also made the inverse kinematic calculations quite costly. According to [3]: “A rigorous mathematical analysis of inverse kinematics for hyper-redundant manipulators was performed by Chirikjian. Chirikjian used techniques from differential geometry to describe robotic kinematics as fitting curves in space, and proposed a novel type of hyper-redundant robot known as a variable geometry truss (VGT) structure. The differential geometry approach describes a curve in space as a series of moving frames originating at the base of the manipulator, progressing along the length and ending at the end-effector. The curves used are taken from a basis set which tend to form S-shaped curves. Once a set of curves is determined to approximate a desired configuration, then the manipulator is fitted to the curve set”.

Furthermore, [3] continues: “Chirikjian et al. have attempted to use the Frenet-Serret formulation for curves in space, but have found it to be too computationally expensive to use in real-time for calculating inverse kinematics. Related advanced techniques by Gravagne use other formulations for these curves (such as wavelet functions), but require many simplifying assumptions about the manipulator design (such as equal segment lengths, or homogeneous bend of the backbone material), and these techniques have only been applied to 2-D (planar) robots. Before the work discussed herein, no general approach has been presented for finding the inverse kinematics of hyper-redundant manipulators in a three-dimensional setting”. The method mentioned and used by [3] was a genetic combinational algorithm, and a similar approach (error-optimization method) was also conducted by [10]. However, the present article tries to introduce new alternatives by comparing several global-optimization algorithms and an exhaustive method.

Although the latter paragraph is considerably old, some of the latest publications still state the same. According to [11]: “In the case of hyper redundant manipulators with high degrees of freedom, the computational burden of pseudoinverse Jacobian becomes prohibitive despite proposed improvements. Furthermore, most of the proposed schemes handle the inverse kinematic problem at the velocity level only. Therefore, among these many schemes, the geometrical method for path planning is preferred because of its simplicity, power saving and reduced computations compared to others”. Nevertheless, it fails to deliver a definitive and viable solution to the inverse kinematics issue.

This computational problem is still persistent for getting a valid solution for inverse kinematics. According to the results to be presented in this article, the processing time for inverse kinematics calculation undergoing an exhaustive method can go up to one year for a 10-module robot (see Figure 6, section III-A). This calculation time is obviously unacceptable for any real life application. Therefore, an optimization method was also studied as another solution to minimize both

processing time and position/orientation error.

### III. INVERSE KINEMATICS

In order to gather data for the comparative study, a statistically significant number of iterations (20 runs per method) had to be computed, with modules ranging from one through four for the exhaustive method (section III-A) and from one through ten for the error-optimization method (section III-B).

It is important to notice that the processing time for all methods was calculated only as the time the algorithm took to get to the solution, without taking into consideration other small time-consuming related calculations. This was done to provide more realistic and uniformed comparative times with all the different methods detailed in this section.

Since both approaches to be discussed (exhaustive and optimization algorithms) are based on both position and orientation error, the first step was to identify an expression capable of describing the error itself.

Equation (1) shows how the final plate position error  $\epsilon_P$  was calculated by a simple Euclidean distance, where  $P_u$  stands for the ‘user-defined’ final position vector and  $P_c$  is the ‘calculated’ final position vector.  $P_x$ ,  $P_y$  and  $P_z$  stand for the  $x$ ,  $y$  and  $z$  component value of both the user-defined and calculated position vectors.

$$\epsilon_P = \sqrt{(P_{u_x} - P_{c_x})^2 + (P_{u_y} - P_{c_y})^2 + (P_{u_z} - P_{c_z})^2} \quad (1)$$

This way of calculating the error was selected because it has a very light computational cost and is also a practical way of weighting the error in function of the distance to the target. The closer the calculated final point is to the user-defined position, the lower the  $\epsilon_P$  and, consequently, the final error expression.

The same methodology was used to compute the final orientation error  $\epsilon_A$ .  $A$  is a vector indicating the orientation of the *HRM* last plane’s normal vector. The only difference between the position error is that both vectors  $A_u$  (user-specified) and  $A_c$  (calculated) are unitarian, thus making  $\epsilon_A$  to vary between limited parameters, as shown in equation (3).

$$\epsilon_A = \sqrt{(A_{u_x} - A_{c_x})^2 + (A_{u_y} - A_{c_y})^2 + (A_{u_z} - A_{c_z})^2} \quad (2)$$

$$0 \leq \epsilon_A \leq 2 \quad (3)$$

Finally, as mentioned at the beginning of this chapter, a complete and final error equation was needed to compare all methods, which was achieved by applying equation (4).  $\lambda$  is chosen to emphasize the priority of the type of error the user wants to reduce. For the comparative purposes of this article,  $\lambda = 4$  was used, since it provides a good scalar parameter to relate position error with orientation error in equal manners.

$$\epsilon = \frac{\epsilon_P^2}{\lambda} + \epsilon_A^2 + \epsilon_P \epsilon_A \quad (4)$$

### A. Exhaustive Method

As its name suggests, this method works by examining every single combinational possibility of all robot's modules, and then selects the one with least error and displays it as a final answer.

In Figure 5 a typical error distribution over the different possible module combinations can be seen. Although it is not a complete graphic of all possible modules, it is a fairly good representation on how the error function should look like. This figure also provides more information: the error function which the 'error-optimization methods' (section III-B) will be trying to minimize is far from being smooth and easy to work with.

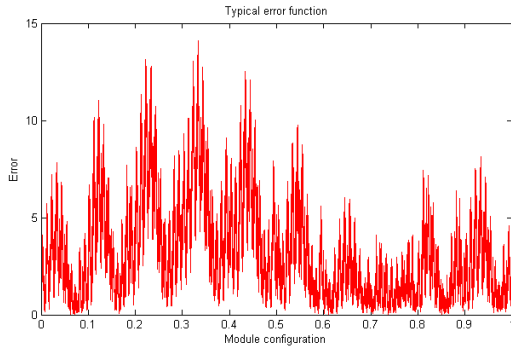


Fig. 5. Inverse kinematics: Position/Orientation error for a 4-module 10DOF hyper-redundant robot

Exhaustive method runs were only performed for up to 4 modules. Interpolation, with a 99.9% accuracy, was then used to calculate the time it would take for more modules to converge to the optimal solution.

Also, as shown in Figure 6, processing time is the main constraint for this particular method. Because of the exponential growth of possible combinations, something as little as adding one module to the system can make the processing time to increase around ten times.

Due to the high processing times, it is clear that although the exhaustive method guarantees the best possible solution, it is also restricted to be used with only a few modules. This same conclusion was reached by [10].

### B. Error-optimization methods

There have been proposed some inverse kinematics techniques for HRMs, and many authors have worked with them along the years, but a mere error-optimization approach to this problem is still yet to be published.

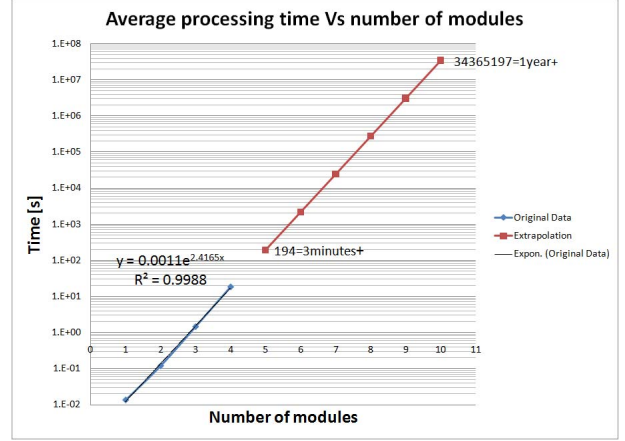


Fig. 6. Inverse kinematics: Processing times for exhaustive method

As such, a comparative study between the most common global-optimization algorithms: *Patternsearch*, *Genetic Algorithm*, *Globalsearch*, *Multistart: fminunc*, *Multistart: lsqnonlin* and *Simulannealbd* [12] (available by default in MatLab R2010a) was performed using equation (4) as the target function to minimize.

Unlike the exhaustive method, the optimization algorithms are not so time consuming, but the trade off for this propriety is the absence of the completely guaranteed optimal solution.

Since the error function to be minimized has an expected appearance as shown in Figure 5, it is important to give *Patternsearch*, *Multistart*, *Globalsearch* and *Simulannealbd* the best starting point as possible in order for them to, ideally, find the global minimum.

The processing time results for the error-optimization algorithms are presented on Figure 7. It can be observed that the times are dependent on the selected optimization algorithm, and while some are considerably faster (like *Patternsearch* and *Multistart*) when compared to others, it does not necessarily mean they have the least position/orientation error.

## IV. COMPARATIVE STUDY

As mentioned at the beginning of section III, 20 runs were performed to get statistically significant data ranging from one through ten modules for the error-optimizations methods. This simulation was done with 1 cm as the minimum distance between modules and 3 cm as the length of each one of them (modules are squared).

The error-optimization algorithms try to reach the user-input final position and orientation point by optimizing the error function (equation (4)), while the exhaustive method goes through every single possible combination. Since it is quite difficult to reach the exact point with the exact orientation for the error-optimization algorithms, a final position error (Figure 8) and a final orientation error (Figure 9) are obtained.

The reason that there is no position/orientation error for the exhaustive method is because it will always find the

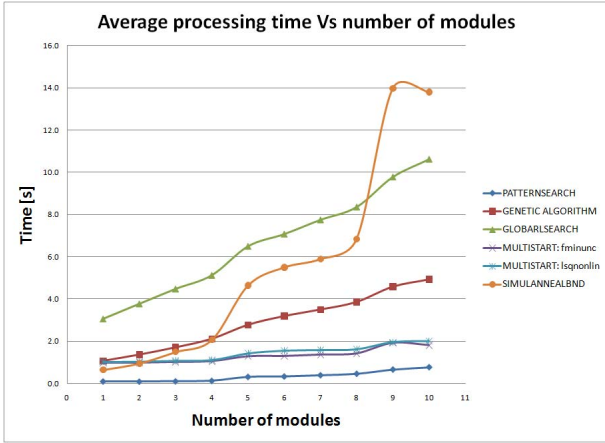


Fig. 7. Inverse kinematics: Processing times for various optimization algorithms

best possible answer, while not necessarily so for the error-optimization algorithms.

A good approximation of how the complete error function for 4+ modules looks like is shown on Figure 5. It is also notable that trying to find the global minimum by minimizing the error is a hard task: it is full of crests and valleys very close to each other, it is almost non-differentiable and is full of local minima with such a small error value that many algorithms could interpret these answers as final solutions, thus causing the position/orientation error.

By comparing the results obtained in sections III-A and III-B, it is noticeable that each method has its own particular field of operation: Exhaustive method is a very powerful tool for small number of modules (less or equal to 4), since it gives the optimal minimum position and orientation error in a fairly good time (less than 18 s). However, due to the small number of modules it can handle, exhaustive method has limited and practically none real-life applications.

Processing time for the exhaustive method could be speed-up to only 10% of its value if a first-module calculation was to be made by projecting the final point onto the  $XY$  plane and then orienting the first module towards that point. Nevertheless, doing this first calculation does not guarantee the optimal configuration, as it does the normal exhaustive way. Furthermore, the remaining 10% could still be a large problem because of the exponential nature of the possible module combinations.

This idea is not entirely useless, since the same principle was used for giving the error-optimization algorithms (*Patternsearch*, *Multistart*, *Globalsearch* and *Simulannealwnd*) their first starting module/point.

‘Long-lasting’ results optimization algorithms, such as *simulannealwnd*, *genetic algorithm* and *globalsearch* do not necessary have lower position and orientation errors. In other words, it is not possible to determine which algorithm to always use, since it is a problem-specific dependent situation. Despite of

this, it is noticeable why the error-optimization approaches proposed by the authors mentioned in section II were based on the genetic algorithm: it is the fastest among them three and has a very similar position and orientation errors.

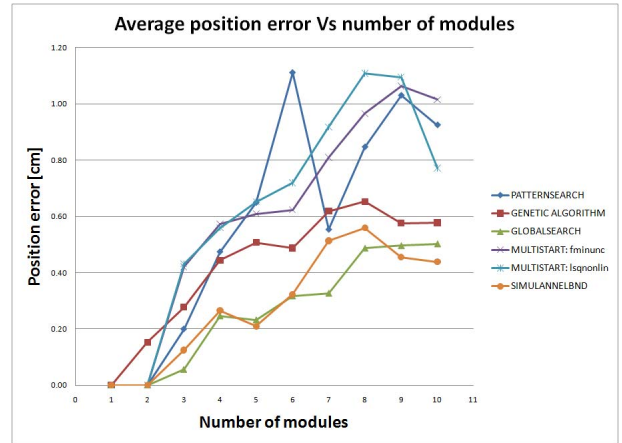


Fig. 8. Optimization method: Position errors

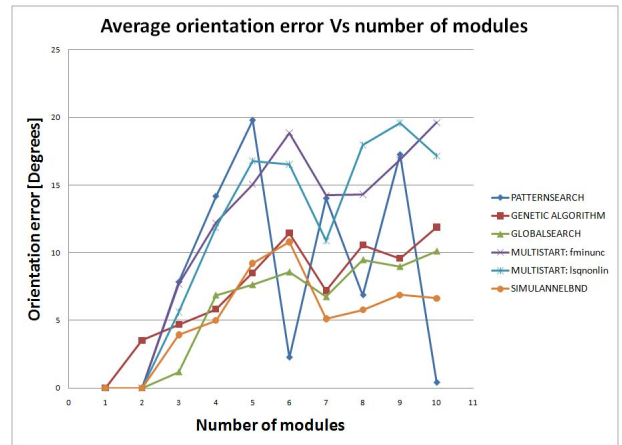


Fig. 9. Optimization method: Orientation errors

The most remarkable thing of the last presented graphs (Figure 8 and Figure 9) of the optimization method is that despite them having errors (amount depending on each algorithm), they are still very good approximations. The ratio of acceptable solution versus processing time is very small when compared to the exhaustive method; and perhaps the most important detail is that the error could be adjusted depending on the application that the *HRM* is intended for.

For example, if a lamp is going to be used for laparoscopic applications, the orientation error ( $\epsilon_A$ ) would be more important than the position one ( $\epsilon_P$ ), and  $\lambda$  could be adjusted as such. On the opposite, for applications where position is key,

the same parameter could also be modified to give priority to the position. The results shown in this section were obtained by trying to balance both errors as equally important.

All the information presented in this section IV sets a start point for continuing studying error-optimization approaches to the inverse kinematics of hyper-redundant robots and also opens a window of opportunity, since it has been demonstrated that this is a valid way of tackling this type of challenges.

## V. CONCLUSIONS AND FUTURE WORK

In this paper several approaches applied to compute the inverse kinematics of a 10 degrees of freedom hyper-redundant robot were presented and compared: It was observed that the exhaustive method is a very powerful tool for small number of modules (less or equal to 4), since it gives the optimal minimum position and orientation errors in a fairly good time (less than 18 s). However, due to the small number of modules, it is not suitable for any real-life applications. Unlike the exhaustive method, the error-optimization algorithms are not so time consuming, but the trade off for this propriety is the absence of the completely guaranteed optimal solution.

Moreover, it is of great interest to develop new methods, applications and prototypes of hyper-redundant manipulators (*HRMs*), since the advantages of moving on irregular environments and extreme maneuverability are appealing for biomedical applications.

Since the presented *HRM* is still under development and has not reached the prototype stage yet, better materials for miniaturization could be suggested or incorporated in the future. This would allow the *HRM* to access in an easier manner different body cavities and perform its tasks as an endoscope, ideally providing both better diagnosing/treatment capabilities to the health care personnel and less procedure injuries/discomfort to the patients. Keeping the same line of thought, artificial muscles could also be employed in order to replace the electromagnetic actuators. Even though these are still currently under study, the potential uses and applications of artificial muscles is almost infinite, and the medical field would definitely be very benefited by them.

Also as future work, better performing computers (that is, with better hardware characteristics) could also be used, and since they do not have to be built-in the *HRM*, they would allow both quicker solutions and better resources management. Parallel processing could also be implemented in order to speed up the processing time.

A problem-customized global optimization algorithm could deliver a big performance upgrade to the current project, and could be developed from the beginning to take into consideration restrictions dependent on the application the *HRM* would be used in.

## ACKNOWLEDGMENT

This work is part-funded by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Portuguese Foundation for Science

and Technology within project FCOMP - 01-0124-FEDER-022701.

## REFERENCES

- [1] P. Liljeback, K.Y. Pettersen, Stavdahl and J.T. Gravdahl, *A review on modelling, implementation, and control of snake robots*. Robotics and Autonomous Systems 60 (2012) 2940 Norway, August 2011.
- [2] H.Zhu, *Single Motor Driven Hyper-Redundant Manipulator*. Published in: <http://www.imdl.gatech.edu/haihong/Arm/Arm.html>
- [3] D. Sofge and G. Chiang, *Design, implementation and cooperative co-evolution of an autonomous/teleoperated control system for a serpentine robotic manipulator* Proceedings of the American Nuclear Society Ninth Topical Meeting on Robotics and Remote Systems Seattle Washington, March 2001.
- [4] G.S. Chirikjian and J.W. Burdick, *A Hyper-Redundant Manipulator*, IEEE Robotics and Automation Magazine. December, 1995.
- [5] K. Xu and N. Simaan, *Actuation Compensation for Flexible Surgical Snake-like Robots with Redundant Remote Actuation* Proceedings of the 2006 IEEE International Conference on Robotics and Automation. Orlando, Florida: May 2006.
- [6] A. Wingert and M. Lichter, S. Dubowsky, M. Hafez, *Hyper-Redundant Robot Manipulators Actuated by Optimized Binary Dielectric Polymers* Smart Structures and Materials Symposium 2002 San Diego, CA, USA, 2002.
- [7] V. Sujan and S. Dubowsky, *Design of a Lightweight Hyper-Redundant Deployable Binary Manipulator*. Journal of Mechanical Design. Volume 126, Issue 1. January, 2004.
- [8] A.B Stalkin and J.W. Burdick, *The development of a robotic endoscope*, IROS '95 Proceedings of the International Conference on Intelligent Robots and Systems-Volume 2. IEEE Computer Society Washington DC, USA 1995.
- [9] G. Chen, M.T. Phamb, T. Redarc, *Sensor-based guidance control of a continuum robot for a semi-autonomous colonoscopy*. International Journal of Robotics and Autonomous Systems 57, Elsevier, 2009.
- [10] Matthew D. Lichter, Vivek A. Sujan, and Steven Dubowsky, *Computational Issues in the Planning and Kinematics of Binary Robots*. Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, DC May 2002.
- [11] S. Yahya, M. Moghavvemi and H.A.F. Mohamed, *Redundant manipulators kinematics inversion*. Scientific Research and Essays Vol. 6(26), November, 2011.
- [12] Mathworks: *MatLab. Global Optimization Toolbox: User's guide*, Mathworks documentation. 2011.
- [13] .A. Gravagne, *On the kinematics of remotely-actuated continuum robots*. Proceedings of the 2000 IEEE International Conference on Robotics and Automation San Francisco, CA April 2000.
- [14] G.S. Chirikjian and J.W. Burdick, *A Modal Approach to Hyper-Redundant Manipulator Kinematics*, IEEE Trans. on Robotics and Automation, Vol. 10, No. 3. June, 1994.
- [15] A. Brett Stalkin and J.W. Burdick, *Kinematically optimal hyper-redundant manipulator configurations*, Proceedings of the 1992 IEEE International Conference on Robotics and Automation. Nice, France: May, 1992.
- [16] G. Chirikjian, *A Binary Paradigm for Robotic Manipulators* Proc. of IEEE International Conference on Robotics and Automation, Vol. 4, pp. 3063-3069. 1994.
- [17] D. Lees and G. Chirikjian, *A Combinatorial Approach to Trajectory Planning for Binary Manipulators*. Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN, 1996.