Optimizing Inverse Kinematic Solutions Using Fuzzy Logic

Mohammed Elsharif, Nabil Drawil, and Mohamed Samir Elbuni
Department of Computer Engineering, Faculty of Engineering
University of Tripoli, Tripoli, Libya
mohammed.elsharif@gmail.com, n.drawil@uot.edu.ly, mselbuni@gmail.com

***Abstract* – This paper presents a new and fast scheme for the provision of Inverse Kinematics Solution (IKS) for a 3-DOF arm system (LabVolt 5250 Robotic Arm). Different approaches of IK are investigated, and their pros and cons are presented. The objective was to develop an IKS that unifies the different approaches in order to strengthen the advantages and alleviate the weaknesses or recover from them efficiently. The main contribution of this research work is the development of IKS that is more accurate than analytical solutions and faster than numerical methods**.

Index Terms—**Inverse Kinematic, Fuzzy Inferencing, Sequential Quadratic Programming.**

I. INTRODUCTION

Inverse kinematics is the science of studying the position of an end effector by using the Cartesian system as method of defining a location in space. Alternatively, Inverse kinematics can be defined as a mapping of “locations” in 3-D Cartesian space to “locations” in the robot's internal joint space. This need naturally arises anytime a goal is specified in external Cartesian coordinates [1].

Because the kinematic equations are nonlinear and lack sufficient information, their solution is often challenging to be obtained in a closed form [2]. Also, questions about the existence of a solution and about multiple solutions arise. The existence or nonexistence of a kinematic solution defines the workspace of a given manipulator. The lack of a solution means that the manipulator cannot attain the desired position and orientation because it lies outside of the manipulator's workspace [1].

In this work, inverse kinematics solutions (IKS) are categorized into classes: analytical solutions and numerical solutions. Analytical solutions allow for fast and efficient calculation of the joint angles which give a desired end-effector configuration. However, it is often robot specific solution due to the number of degree of freedom (DoF) and joint angle representation. Most analytical solutions are obtained using geometric and algebraic identities to solve the set of nonlinear equations, coupled with algebraic equations which define the inverse kinematics problem [3]. On the other hand, numerical solutions can handle any DoF, but often rely on an iterative procedure to solve. This yields slow convergence time if it did not get stuck in local minima. The strength of numerical solutions lies in the high accuracy produced when converge to the right solution.

Furthermore, analytic form solutions are relatively faster and have a reasonable lower bound of accuracy than numerical solutions [3]. Inverse kinematics is computationally expensive and can result in significant control delays in real time for redundant Robots. However Fuzzy logic can reduce the amount of time and provide more suitable initial guess for the solution similar to how humans solve the inverse kinematics problem [4], [5]. Nevertheless, tuning the fuzzy logic is not always feasible due to the lack of expertise.

In this research work, we develop an IKS for LabVolt 5250 Robotic Arm (LRA), which is a prime example of a system that is designed for simple educational tasks. Such development allows the utilization of this arm in advanced applications. Difference approaches of IKS are investigated in this paper, and their pros and cons are presented. The aim in this work is to develop an IKS that unifies the different approaches so that advantages are strengthen and weakness are recovered efficiently.

The main contribution of this research work is the development of IKS that is more accurate than analytical solutions and faster than numerical methods. The remainder of this paper is organized as follows. Section II presents the principle methodologies to form an IKS. Section III introduced the proposed unified techniques to overcome the individual drawbacks. Section IV explains the experimental setup and shows the results. Finally, Section V provides concluding remarks.

II. PRINCIPLE METHODS OF IKS

The IKSs presented here deals with 3DoF robotic arm. These solutions can be implemented on LRA, which we have in Lab, despite its 6 DoF (θ_{base} , θ_{shldr} , θ_{elbow} , θ_{wrist} , $\theta_{gripper_{Open/Close}}$, θ_{rotate}). The possibility of such approach relies on the fact that the rest of joints can be excluded by knowing the two horizontal coordinates of a target object, in Cartesian system (regular 3D space). Therefore, the

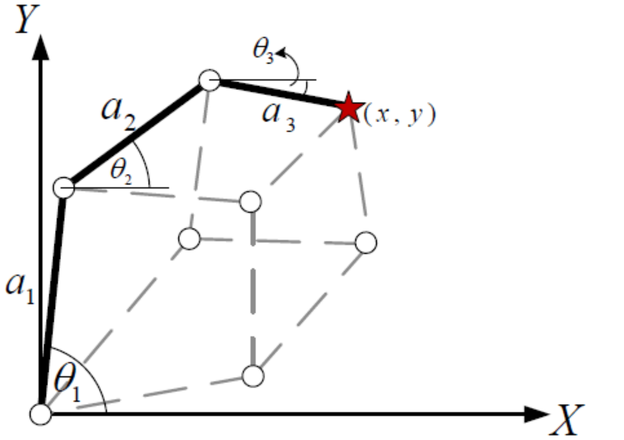

Figure 1 3DoF of robotic arm. Different possible solution to approach point (x, y)

focus here is on the 3 joints with vertical DoF: θ_{shldr}, θ_{elbow}, θ_{wrist}.

Fig. 1 shows a symbolic graph for the robotic arm in the vertical space (2D) and depicts the angles of interest, where θ_1, θ_2, θ_3 signify θ_{shldr}, θ_{elbow}, θ_{wrist} in the robotic arm respectively. The symbols a_1, a_2, a_3 signify the length of the arm segments. In the following, the formulation of two principle IKSs for LRA are explained.

A. *Analytic IKS*

Two special properties of the LRA make the presented solution herein different from others, like the ones in [6], [7]. The first property is the way the robot measures angles. Angles of LRA joints are considered independent and start their zeros horizontally, as depicted in Fig. 1. In other words, commanding LRA to change θ_1 does not result in any change in the upper θ's. The second special property is related to the customary reduction in the number of variables to deal with inverse kinematic problem by fixing θ_3 and solving for the rest of angles. Such assumption is valid in applications of drawing or cutting on horizontal surfaces [6], but not for many cases where robot is aimed to approach targets with different horizontal/vertical levels in the surroundings.

Consequently, when a 3DoF Robot Arm conducting movement in two dimensional space (vertical space) Fig. 1, the respective kinematic equations are as represented in equations (1) and (2). As seen in Fig. 1 many kinematic movements, depicted by the dashed line, converge to the solution of the Inverse Kinematics equations. However, solutions of these equations might not be feasible due to the physical limits of the arm joints or due to the length of the arm segments. Hence, the analytical solution must be searched for until a valid solution is found, iteratively.

$$x = a_1\cos(\theta_1) + a_2\cos(\theta_2) + a_3\cos(\theta_3) \tag{1}$$

$$y = a_1\sin(\theta_1) + a_2\sin(\theta_2) + a_3\sin(\theta_3) \tag{2}$$

Using equations (1) and (2) to solve for θ_1, θ_2, and θ_3, it is clearly observed that a very large set of solutions will be produced as a matter of having number of unknowns larger that number of knowns, i.e., x and y. The number of solutions depends on the precision of the robotic arm joints and gravitates towards infinity as the precision goes higher. Some of the solutions that converge are feasible for the robot's physical limits, but many of them are out of the robot's physical limit. For a Cartesian movement, if one of the joints, θ_i: $i = 1,2,3$, for example the wrist joint, can be estimated, then equations (1) and (2) can be reduced to

$$\tilde{x} = a_1\cos(\theta_1) + a_2\cos(\theta_2) \tag{3}$$

$$\tilde{y} = a_1\sin(\theta_1) + a_2\sin(\theta_2) \tag{4}$$

and now can be solved and now can be solved, where θ_3 is estimated and becomes part of the reduced vectors $\tilde{x}$ and $\tilde{y}$. The algorithm of IKS using the analytical approach is presented below.

Algorithm 1 : Iterative Analytic IKS

Input: x, and y;
Output: $\hat{\theta}_i$ $i = 1, 2, 3$
assume $\hat{\theta}_3 = -90$;
while physical limit not reached **do**
  solve for θ_1 and θ_2 simultaneously;
  if Solution is feasible **then**
    return $\hat{\theta}_i$ $i = 1, 2, 3$;
  else
    increment/decrement $\hat{\theta}_3$;
  end if
end while

This algorithm guarantees that the robot will always reach the target object as long as the solution is within the physical limits of the robot. The caveat of this algorithm is that solutions might not be accurate if the movement needed to be within millimeters' distance. In terms of execution time, worst case scenario to find the solution is when all possible degrees of θ_3 is tried out.

B. *Sequential Quadratic Programming (SQP)*

SQP is the numerical method used here to produce IKS for the LRA. SQP can handle any number of variable so that there is no need to reduce the θ's to obtain a solution. SQP also can handle smooth nonlinear problems, such as the one under disposal. It is a numerical optimization method that approximates solutions by minimizing a quadratic function until a designated minimum level of the objective function is reached. It reaches the solution after a number of iterations. Worth mentioning hear, the physical limitations of LRA movements can cause the SQP to stuck in local minima.

For the application of Inverse kinematics, like any convex nonlinear optimization methods, [8], equations (1) and (2) must be used to construct the SQP formulation, Equation (5). As seen, the joint angles, in the objective function, are bounded by the physical limits of LRA.

One drawback with the SQP method is the tendency for SQP to gravitate towards local minima that do not represent the right IKS. Usually this occurs when the initial value of θ's is not adequate or, more specifically, far from the right solution. In the following sections, this issue is tackled where we attempt to find adequate starting angles.

$$\arg\min_{\theta_1,\theta_2,\theta_3}\left(x-\sum_i a_i\cos(\theta_i)\right)^2+\left(y-\sum_i a_i\sin(\theta_i)\right)^2 \quad (5)$$

Subject to

$$-29^\circ \le \theta_1 \le 120^\circ$$
$$-120^\circ \le \theta_2-\theta_1 \le 118^\circ$$
$$-78^\circ \le \theta_3-\theta_2 \le 78^\circ$$

III. Unified Approaches

From the previous section, drawbacks are contrasted from one method to another, while these methods do not share same strength points. This observation motivates the potential improvement by fusing these methods and make them work in synergy. One common issue in the two methods is the initial estimate of θ(s).

The initial estimate of θ(s) can be a key solution if determined intelligently. Therefore, artificial intelligence is deployed in this research work to tackle this issue. Because neural networks and genetic algorithms are known for their complicated training procedures/sets, they were eliminated as a solution in this work. On the other hand, Fuzzy logic and Fuzzy Inference Systems (FIS) appears as an adequate tool to solve the mentioned issue. For instance, knowing the target location (*x*,*y*) in 2D, one can tell roughly how the arm should articulate. In other words, values for θ's can be estimated initially so that arm does not gravitate towards potential local minima. By definition, the framework work of the FIS can serve adequately this task as explained next.

An FIS is designed so that the knowledge about the arm articulation, based on the target location, is embedded in the FIS in a form of knowledge base. This knowledge base consists of number of rules, such as the following

$$IF\ I_1 \text{ is } A_i\ AND\ I_2 \text{ is } B_i\ THEN\ out \text{ is } C_i$$

where I_1 and I_2 signifies fuzzified inputs of interest, and *out* signifies the output variable; A_i, B_i, and C_i are fuzzy sets introduced by Zadeh in [9] that governs the i^{th} rule. The fuzzification of the measurements and evaluation of the rules are performed using *S-Norm* operator of the FIS. The aggregation step is performed using summation method to maintain the contribution of each membership to the sets of output variable. The decision crisp value is computed by defuzzifying the aggregated output using mean of maximum method [10], which selects the center of the global maxima volume in the aggregated shape. In the following we propose unifying techniques that aim to improve the performance of the basic techniques.

A. Hybrid Analytic IKS

In this technique, the estimate of θ_3 is produced by a FIS. The fuzzified input variables are the *x* and *y* coordinates of the target object. The output is the crisp value representing θ_3. The fuzzy membership function of the input/output fuzzy sets are depicted in Fig. 2. The knowledge base is comprised of the set of fuzzy rules presented in Table 1.

B. Hybrid SQP for IKS

One solution for the SQP to avoid gravitation towards local minima is through assigning specific initial values for θ's after examining the location of the target location of the end effector. Such examination can be conducted using FIS. The fuzzified inputs of the FIS are the polar coordinate system of the target (*x*,*y*): $R=\sqrt{x^2+y^2}$ and $\theta_t = \operatorname{atan}(y/x)$. The reason of constructing such inputs is that polar coordinate system can be interpreted into output $\theta_i : i \in \{1,2,3\}$ using logical rules. The FIS have three crisp values for each output in this case. Fig. 3 shows the membership functions of the input fuzzy sets,

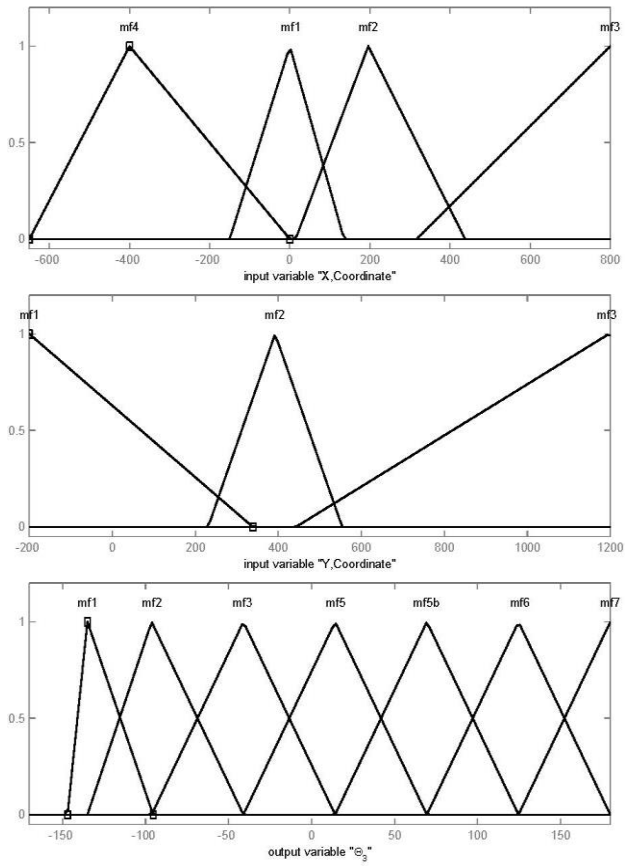

Figure 2 FIS membership functions for input variables: x and y, and output variable θ_3.

TABLE 1 FUZZY RULES TO ESTIMATE.

Rule number	Fuzzy rule
1	if (X is mf1) and (Y is mf1) then (θ_3 is mf1)
2	if (X is mf2) and (Y is mf1) then (θ_3 is mf2)
3	if (X is mf3) and (Y is mf1) then (θ_3 is mf3)
4	if (X is mf1) and (Y is mf2) then (θ_3 is mf1)
5	if (X is mf2) and (Y is mf2) then (θ_3 is mf3)
6	if (X is mf3) and (Y is mf2) then (θ_3 is mf5)
7	if (X is mf2) and (Y is mf3) then (θ_3 is mf6)
8	if (X is mf3) and (Y is mf3) then (θ_3 is mf5b)
9	if (X is mf4) and (Y is not mf1) then (θ_3 is mf7)

where the membership functions of the output fuzzy sets are shown in Fig. 4.

The FIS rules relates fuzzified inputs to the output fuzzy sets are depicted in Table 2. The entries of this table represent the fuzzy sets of the three outputs respectively, where 'd' means down, 'm' means mid fuzzy set, and 'u' means up fuzzy set.

TABLE 2 FIS KNOWLEDGE BASE RULES FOR HYBRID SQP.

θ_t \ R	Small	Mid	Large
Small	d,d,u	d,d,m	d,m,m
Mid	d,d,u	d,d,u	m,m,m
Lars	u,u,d	h, u, d	u,m,m

By using this FIS it is expected to improve the performance of SQP method, as shown in the next section.

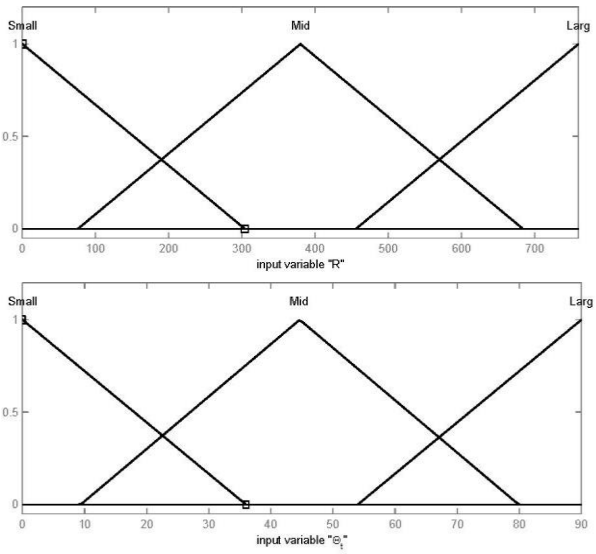

Figure 3 FIS membership functions for input variables: R and θ_t .

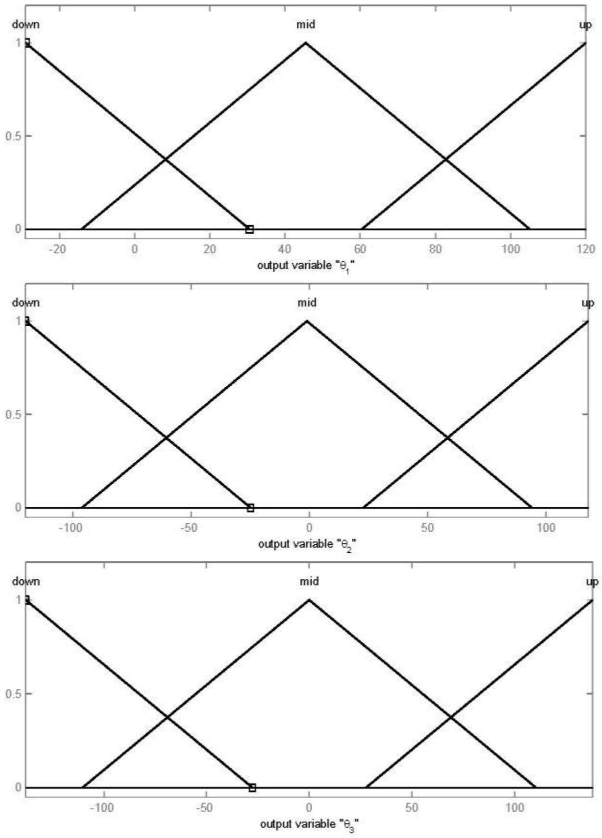

Figure 4 FIS membership functions for output variables: θ_i, $i \in \{1,2,3\}$

IV. EXPERIMENTAL WORK

Matlab was used for simulation with both the iterative analytical method and SQP, with and without the assistance of fuzzy logic. For the purpose of these experiments, a dataset was created randomly to generate target locations, which are scattered in the space accessed by LRA. Fig. 5 shows the dataset in 2D. These data points are bounded by the length of the LRA and body of the arm base, which is located at the origin.

The experiments are running by executing the different basic methods and unifying techniques. The performance measures are in terms of average executing time needed to obtain a solution and the average level of accuracy obtained by each solver. Standard deviation is calculated as well in most of the cases to bring further insight to the performance level.

A. Computational Time

Computation Time is important indicator, when it comes to inverse kinematics due to the high demand for instant results and real-time performance. Time is an indicator for the efficiency of a given method, and its ability to get implemented on low power processing systems, like cheaper micro controllers and other embedded systems.

As can be seen in Fig. 6 the first two epochs (2X500 executions) spent much less time compared with the rest of executions (from 1001 to 2000). That is because analytic method with and without FIS run first, then SQP and hybrid SQP run respectively.

B. *Displacement Error*

Both the Analytical Iterative Method and Sequential Quadratic Programming method were tested with and without the Fuzzy Prediction System. For the analytic methods, it is clear that they outperform SQP methods in the overall, see Fig. 7. However, SQP produces much less error in it converge to the right solution. The spikes in the

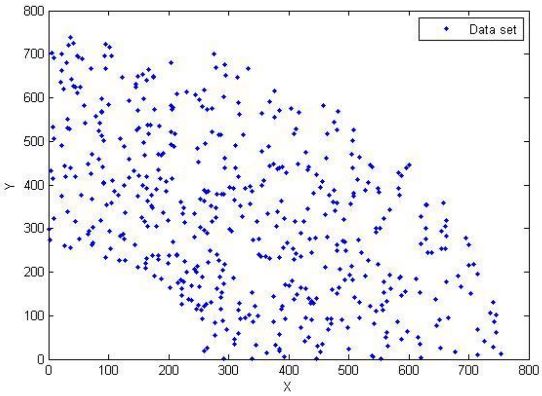

Figure 5 Randomly created dataset.

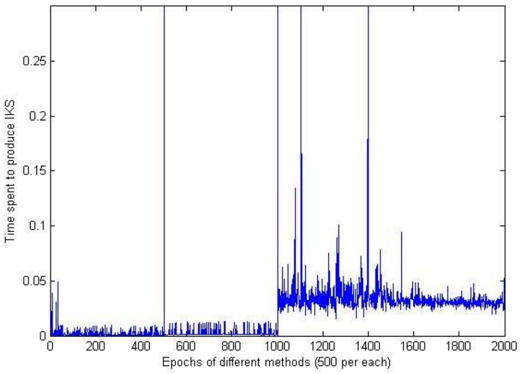

Figure 6 Comparison of execution time.

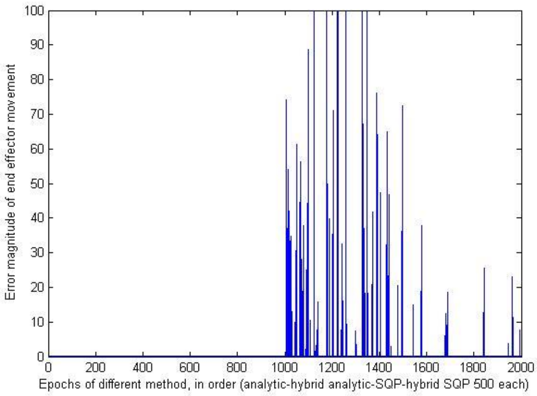

Figure 7 Comparison of error magnitude in end effector movement.

SQP methods result from gravitating the solution to local minima. It can be seen that the number of spikes extremely diminished when hybrid SQP is used.

C. *Remarked Improvement*

The results obtained above inspire an idea of mitigating the spikes in hybrid SQP either by switching to hybrid analytic method if a spike was discovered or use the solution of the hybrid analytic technique, i.e., θ_i : $i \in \{1, 2, 3\}$, as an input to the hybrid SQP. both fusing techniques are not expected to worsen the execution time cost, since analytic method does not spend much time compared to SQP method. Also, if hybrid analytic technique used as an initialization process for the hybrid SQP, then SQP can start from a very close initial point to the actual solution, which leads to very small number of iteration to converge to the target location with extremely precise estimate.

Fig. 8 shows the performance of the improved hybrid SQP in terms of error magnitude (in mm). First observation is the great reduction in the error magnitude when hybrid analytic used after hybrid SQP during the middle 500 epochs. In the last 500 epochs, hybrid SQP is executed after obtaining the initial θ_i's from the hybrid analytic. It is clear that the last technique outperforms all the previous ones in both performance measures, namely time and accuracy.

Table 3 presents the average and standard deviation of error magnitude statistics for every method and unifying

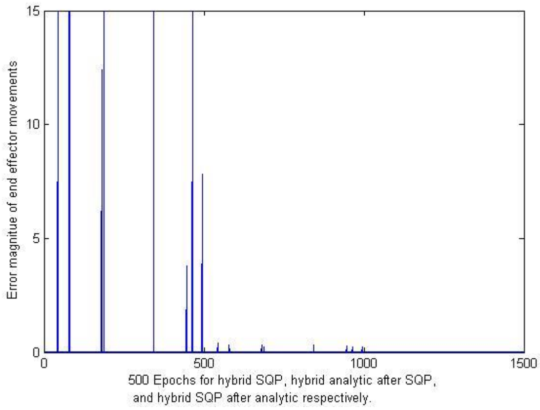

Figure 8 Comparison of error magnitude in end effector movement.

technique.

Table 4 presents a comparison among the methods and techniques in terms of time spent to produce IKS. It is noticeable that using hybrid SQL after hybrid analytic techniques improves the execution time compared with the SQP method. However, hybrid analytic technique outperforms the rest when execution time is considered.

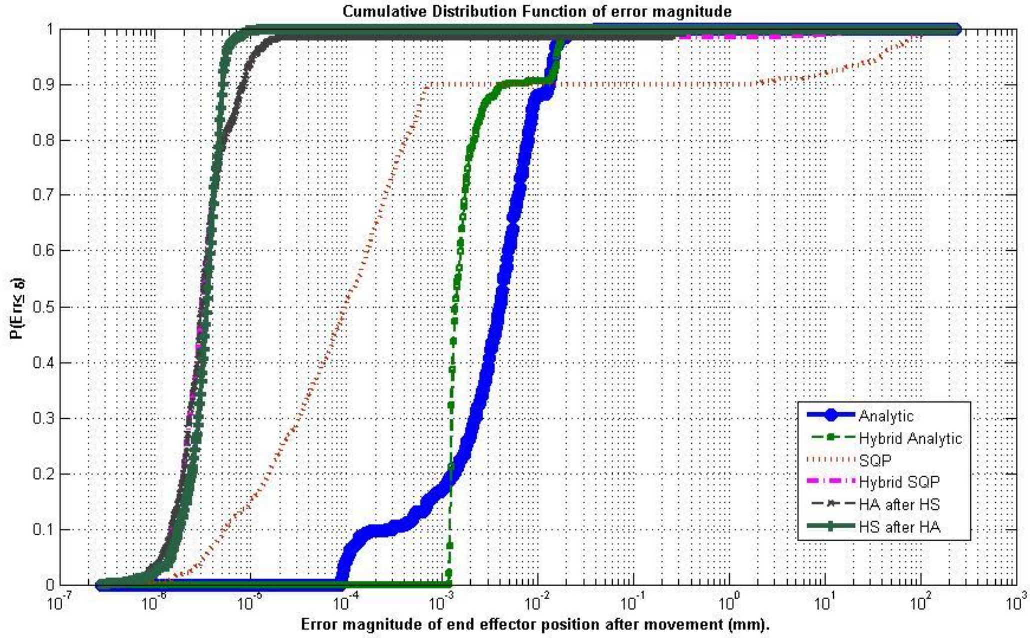

Figure 9 Comparison in terms of cumulative distribution function of error magnitude.

TABLE 3 PERFORMANCE COMPARISON IN TERMS OF ERROR MAGNITUDE OF THE END EFFECTOR MOVEMENT.

Method	Magnitude error (1E-3 mm)	
	Average	Standard Deviation
Analytic	5.79	24.80
Hybrid Analytic (HA)	3.86	22.2
SQP	4,833.38	20,262.56
Hybrid SQP (HS)	286.93	2,593.29
HA after HS	4.76	37.99
HS after HA	3.55 E-3	1.53 E-3

TABLE 4 PERFORMANCE COMPARISON IN TERMS OF EXECUTING TIME.

Method	Execution time (msec)	
	Average	Standard Deviation
Analytic	4.09	17.53
Hybrid Analytic (HA)	2.10	3.08
SQP	39.41	155.23
Hybrid SQP (HS)	34.03	45.13
HA after HS	31.47	3.59
HS after HA	25.14	5.63

Fig. 9 shows a comprehensive quantitative comparison among the different ways of determining IKS in this paper.

III. CONCLUSIONS

This paper presents a general strategy to obtain Inverse Kinematic Solution (IKS) of a multi-link robot arm utilizing a combination of different approaches namely: analytical solution, fuzzy prediction, nonlinear optimization (SQP) and numerical techniques. The simulation studies conducted on a 3-DOF LabVolt 5250 robotic arm shows the effectiveness of the proposed methods over conventional techniques to solve IK problems, resulting in outstanding performance and high accuracy. The hybrid method proves itself as a useful alternative in solving IK in robotic manipulators and can be combined with dynamic control strategies to fulfill high precision tracking requirements in robotic applications.

REFERENCES

[1] J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd ed. John Wiley & Sons, 2008.
[2] W. Kwan, “Closed-form and generalized inverse kinematic solutions for animating the human articulated structure,” 1996.
[3] L. Barinka and R. Berka, “Inverse kinematics-basic methods,” Web http://www. cescg. org/CESCG-2002/LBarinka/paper. pdf, 2002.
[4] A. H. Mary, T. Kara, and A. H. Miry, “Inverse kinematics solution for robotic manipulators based on fuzzy logic and pd control,” in Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Al-Sadeq International Conference on. IEEE, 2016, pp. 1–6.
[5] Y. Xu and M. Nechyba, “Fuzzy inverse kinematic mapping: Rule generation, efficiency, and implementation,” in Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on, vol. 2. IEEE, 1993, pp. 911–918.
[6] A. Hamori, J. Lengyel, and B. Resk´o, “3dof drawing robot using lego-nxt,” in Intelligent Engineering Systems (INES), 2011 15th IEEE International Conference on. IEEE, 2011, pp. 293–295.
[7] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot modeling and control. wiley New York, 2006, vol. 3.
[8] S. Wright and J. Nocedal, “Numerical optimization,” Springer Science, vol. 35, pp. 67–68, 1999.