Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A Learning Framework to Inverse Kinematics of Redundant Manipulators

G. JIOKOU K.A. * **A. Melingui** ** **O. Lakhal** *** **M. KOM** **
R. Merzouki ***

* *Department of Physic's, Faculty of Sciences, University of Yaounde I, Yaounde 8390, Cameroon (ginojiokou@gmail.com).*
** *Department of Electrical and Telecommunications Engineering, Ecole Nationale Supérieure Polytechnique, University of Yaounde I, Yaounde 8390, Cameroon (achillemelingui@gmail.com)*
*** *CRIStAL Laboratory, CNRS-UMR, Villeneuve d'Ascq 59655, France (e-mail: rochdi.merzouki@polytech-lille.fr).*

Abstract: This paper proposes a learning framework for solving the inverse kinematics (IK) problem of high DOF redundant manipulators. The latter possess more DOFs than those required to obtain the end effector (EE) pose. Therefore, for a given EE pose, several joint angle vectors can be associated. However, for a given EE pose, if a set of joint angles is parameterized, the IK problem of redundant manipulators can be reduced to that of non-redundant ones, such that the closed-form analytical methods developed for non-redundant manipulators can be applied to obtain the IK solution. In this paper, some redundant manipulator's joints are parameterized through workspace clustering and configuration space clustering of the redundant manipulator. The growing neural gas network (GNG) is used for workspace clustering while a neighborhood function (NF) is introduced in configuration space clustering. The results obtained by performing a series of simulations on a 7 DOFs redundant manipulator demonstrate the effectiveness of the proposed approach.

1. INTRODUCTION

Recently, kinematically redundant manipulators have been the subject of active research, mainly thanks to their high flexibility and versatility in the execution of certain complex tasks. Indeed, they offer the possibility of simultaneously performing secondary tasks other than the main one, such as joint limit avoidance and obstacle avoidance. These secondary tasks make the solving of inverse kinematics (IK) for this class of kinematic structures an integral part of their real practical application. Typical applications of such systems include collaborative robots, space robotic arms, dexterous hand, and so on.

The methods for solving the inverse kinematic problem of redundant manipulators can be classified into three groups: analytical or closed-form methods, numerical methods, and hybrid methods, i. e. those that combine the two previous ones. Regarding analytical methods, all the inverse kinematic solutions are expressed as functions in terms of the variables pose of the EE. They are computationally efficient and yield all IK solutions for a given EE pose. Peiper (1968) proposed a procedure to obtain IK solutions in closed-form for manipulator robots with three consecutive joints whose axes are parallel or intersect at a single point. How joint limits affect the feasibility of the inverse solution was also explored to develop an analytical method for computing feasible solutions under the joint limits. Other geometric methods have also been developed Wei et al. (2014); Singh and Claassens (2010). However, the above-mentioned contributions are highly configuration-dependent and can be very costly in terms of computation due to the increase in the number of DOFs.

As regards numerical methods, they generally work regardless of the number of degrees of freedom of the manipulator. The inverse of the Jacobian matrix Hollerbach (1985) is generally used to solve IK of non-redundant manipulators, while pseudo-inverse or extended Jacobian inverse Klein et al. (1995) are used for their redundant counterparts. However, Jacobian methods suffer from several shortcomings, including high computation costs and execution time, the existence of local minima and joint singularities.

Other researchers investigated the use of both previous approaches. Several closed-form solutions for inverse kinematic are derived by parameterizing or fixing a set of joint variables. Following that idea, an interesting analytical method based on workspace analysis has been proposed in Zaplana and Basanez (2018). The main idea is to reduce redundant manipulators to non-redundant ones by selecting a set of joints, denoted redundant joints, and parameterizing its joint variables. The inverse kinematics of the non-redundant manipulator obtained is then solved analytically using either Pieper, Paul or other geometric methods. However, this method is very dependent on the degree of redundancy of the manipulator and can be very expensive in terms of computation due to the increase in the number of potential redundant joints. A hybrid method of performing IK for general 2n + 1 (n is the number of joints) DOF manipulators with a spherical joint at the wrist has been proposed in Ananthanarayanan and

Ordóñez (2015). The analytical equations were used to determine the first two and last three joint angles and a numerical technique was used to solve the rest. However, this method can be very expensive in terms of computation time as the number of elbow joints increases.

Machine learning were widely used for solving IK of redundant robotic systems Raja et al. (2019); Kumar et al. (2010). Support Vector Regression, artificial neural networks, and fuzzy systems have been used to solve IKs of redundant manipulators. Kohonen Self-Organized Map (KSOM) networks have been used to solve the redundancy of a 7 DoF arm for tracking trajectories with low errors. The advantages of KSOM networks to maintain continuity of the IK solutions with the possibility to select desired Ik solutions from a set of possible IK solutions have been exploited. However, not only do Cartesian errors remain; several iterative loops may be necessary to improve the position accuracy of the selected IK solution.

In this paper, a learning framework that preserves the multiple IK solutions of redundant manipulators is proposed. The idea consists of dividing the redundant manipulator's workspace into clusters using clustering algorithms and eliminating some joint angle vectors that are too close to each other in each cluster using a neighborhood function (configuration space clustering). Thus, the remaining joint angle vectors in each cluster are potential inverse kinematic solutions for a given input vector that belongs to that cluster. Finally, criteria such as lazy arm movement and minimum angle norm can be applied to select a particular inverse kinematic solution among the redundancy manifolds. However, for a given EE pose; each selected IK solution leads to some pose errors. In this work, to completely avoid iterative loop in the derivation of IK solutions, depending on the redundancy resolution criterion, an IK solution is selected, and the configuration of some joints is maintained fix. Thus, the IK problem of redundant manipulators is reduced to non-redundant ones, and closed-form analytical methods developed for non-redundant manipulators (Pieper, Paul, etc.) can be applied to obtain the IK solution. Unlike self-organizing maps Kohonen (1990) and neural gas methods Martinetz et al. (1991), GNGs do not have parameters that change over time and can continue to learn, adding neurons and connections, until a performance criterion is achieved.

The remainder of this paper is organized as follows: Section II gives an overview of the forward kinematics of serial manipulators. Section III focuses on the development of the proposed learning framework for the IK problem of redundant manipulators. A series of simulations on a 7 DOF redundant manipulator is presented in Section IV to demonstrate the effectiveness of the proposed approach. Finally, some concluding remarks and future prospectives are drawn in Section V.

2. FORWARD KINEMATICS OF REDUNDANT MANIPULATORS

Among the methods developed to derive the forward kinematics of serial manipulators, the Denavit–Hartenberg (D–H) convention is generally adopted. This section describes the forward kinematics of a n-DOF serial manipulator, depicted in Fig.1. The frame assignments follow the Denavit–Hartenberg (D–H) convention, which enables to represent the location of every coordinate frame with respect to every other.

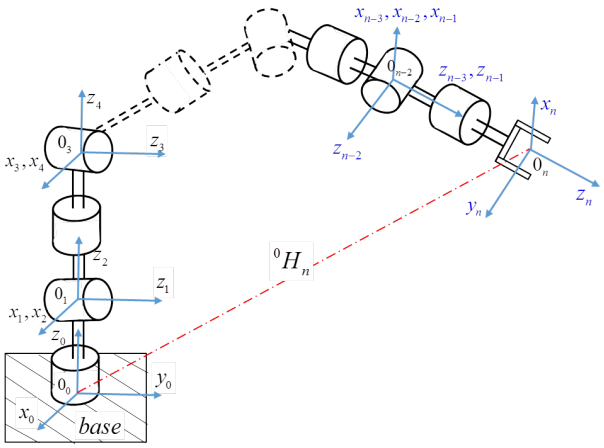


Fig. 1. n-DOF serial manipulator; ${}^{i-1}H_i(\theta_i)$ with $i = 1, 2, ..., n$, is the homogeneous matrix that represents the coordinate frame of the link i

Let us consider the robot's kinematic chain of Fig.1, the forward kinematic can be derived as follows:

$$\begin{aligned} {}^0H_n(\theta) &= {}^0H_1(\theta_1)\,{}^1H_2(\theta_2)\cdots{}^{n-1}H_n(\theta_n) \\ &= \prod_{i=1}^{n} {}^{i-1}H_i(\theta_i) \end{aligned} \tag{1}$$

where n and ${}^0H_n$ represent the total number of DOFs and the homogeneous matrix containing the position and orientation of the EE, respectively. q_i with $i = 1, 2, ..., n$ represents the angle position for each robot joint. The homogeneous matrix ${}^{i-1}H_i$ that transforms the frame attached to link $i-1$ into the frame attached to link i can be expressed as the product of four basic transformations

$${}^{i-1}H_i = T_z(\theta_i)\,T_z(d_i)\,T_x(a_i)\,T_x(\alpha_i), \tag{2}$$

The relation (2) can be rewritten as follows:

$${}^{i-1}H_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where C and S refer to the cosine and sine functions, respectively. Equation (1) that represents the final transformation from the EE frame to the base frame can be rewritten as follows:

$${}^0H_n = \prod_{i=1}^{n} {}^{i-1}H_i(q_i) = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ 0 & 1 \end{bmatrix} \tag{4}$$

where $P_{3\times1}$ is the EE position vector and $R_{3\times3}$ is the rotation matrix, which can be reduced to orientations around the three main axes using the Euler or ZYX notation.

3. PROPOSED LEARNING FRAMEWORK FOR INVERSE KINEMATICS OF REDUNDANT MANIPULATORS

Kinematically redundant manipulators admit an infinite number of inverse kinematic solutions. Thus, two prob-

lems generally emerge, namely, obtaining all the inverse kinematics solutions for a given EE pose and the redundancy resolution, which consists in selecting a particular inverse solution among a multitude of solutions. This section is devoted to the development of the proposed learning framework which combines Analytical and numerical methods to derive the IK solution of redundant manipulators. The section starts with the presentation of the learning architecture, followed by the clustering in the workspace and configuration space. Its end with the redundancy resolution process.

4. LEARNING FRAMEWORK ARCHITECTURE

The proposed learning framework is shown in Fig. 2. The learning database consists of samples identified by (x_k, q_k), $k = 1, \ldots, N$ with N the maximum size of the learning base. x_k and q_k are the EE pose and the configuration vector, respectively. The proposed learning framework is a four-step process. The first step called workspace clustering , The next step, called clustering in the configuration space, consists in forming the couple (x_k, q_k) while eliminating configuration vectors that are too close to the configuration vectors already existing in each cluster. A neighborhood function is used in this step to preserve the conservative property of the obtained inverse kinematic solutions. The third step is redundancy resolution and the last step consists in deducting the IK solution via Paul's method after parameterizing some joints of the redundant manipulator.

4.1 ***Clustering in workspace***

The discretization of the workspace allows to transform the infinite number of IK solutions into a finite number, thus reducing the computation time. However, since only a finite number of poses from the workspace are taken into account in the clustering process, the learning database must cover all regions of the manipulator's workspace. Therefore, the sampling period of the configuration space plays a crucial role as it must be chosen to ensure the presence of redundant solutions in the learning database. Another important parameter is the maximum number of clusters in the sense that it governs the number of redundant solutions. The ability of incremental neural networks such as GNG Qin and Suganthan (2004), Self-organizing incremental neural network (SOINN) to automatically insert new nodes into the hidden layers is exploited to eliminate the crucial parameter of the number of hidden neurons.

The clustering in the workspace is done using GNGs. For a given Cartesian vector x_k, only the winning prototype vector is updated w_p and its direct topological neighbors w_i with $i \in Q_{w_p}$ and $p \in F_{w_p}$. F_{w_p} and Q_{w_p} are the set of winning prototype vectors and the set of direct topological neighbours that are linked by an edge with w_p. The update rule is given as follows:

$$\Delta w_p = \varepsilon_b \left(x_k - w_p\right), \Delta w_i = \varepsilon_n \left(x_k - w_i\right), \forall i \in Q_{w_p} \quad (5)$$

We refer the interested reader to Fritzke (1995) for more details on the GNG algorithm.

4.2 ***Clustering in configuration space***

The clustering in the configuration space consists in associating each winning prototype vector w_p with its corresponding configuration vectors while eliminating configuration vectors that are too close to the configuration vectors already existing in the cluster. Suppose that the winning prototype vector w_p is associated with N_{w_p} configuration vectors referred as $q^j_{w_p}$, $j = 1, 2, ..., N_{w_p}$, as shown in Fig. 2. The input Cartesian vector q_k allows you to create a new configuration vector if it is not too close to those existing in the cluster or to update existing configuration vectors if it is close.

The configuration vector is updated using the following competitive rule

$$q^j_{w_p}(t+1) = q^j_{w_p}(t) + \eta h_j \left(q_k - q^j_{w_p}(t)\right) \quad (6)$$

where $h_j = \exp\left(-(\beta - j)/2\sigma_t^2\right)$ is a neighbourhood function Kumar et al. (2010). The latter generates a continuous and smooth path in the configuration space for a given continuous path in the workspace.

4.3 ***Redundancy resolution***

The clustering in the configuration space results in multiple configuration vectors for a given EE pose. The set $q^j_{w_p}$, $j = 1, 2, \ldots, N_{w_p}$ is the possible inverse kinematic solutions for a given EE pose x_k. A configuration vector can be selected among the redundant manifolds according to a given criteria such as:

- Minimum variation of the configuration space vector
- The criteria of the joint limit avoidance and the obstacle avoidance can also be applied via optimization functions, Zaplana and Basanez (2018).

4.4 ***Derivation of the IK solution***

In general, any selected IK solution results in Cartesian pose errors. In this work, to completely eliminate the remaining Cartesian errors while without include iterative loops, some joints of the redundant manipulator are fixed using the corresponding values of the selected IK solution, such that the redundant manipulator is reduced to non-redundant one. Finally, Paul's method is applied to derive the rest of joint variables. The choice of the fixed joints is straightforward because, as we operate within a particular cluster, small displacements of the EE are sufficient to reach the desired EE pose. For any n-DOF redundant manipulator with a spherical wrist, the following procedure can be implemented. The first joint that rotates the base of the manipulator will remain variable. Two of the following joints $(2, 3, ..., n-3)$ and the three joints of the spherical wrist must also be variable. The other joints can be configured via clustering in the workspace and in the configuration space.

For the implementation of Paul's method, if the last three joints of the manipulator form a spherical wrist as is usually the case, then the problem of inverse kinematics can be decoupled. We first determine the position of the spherical wrist from the base to the $n-3$-th joint; then we

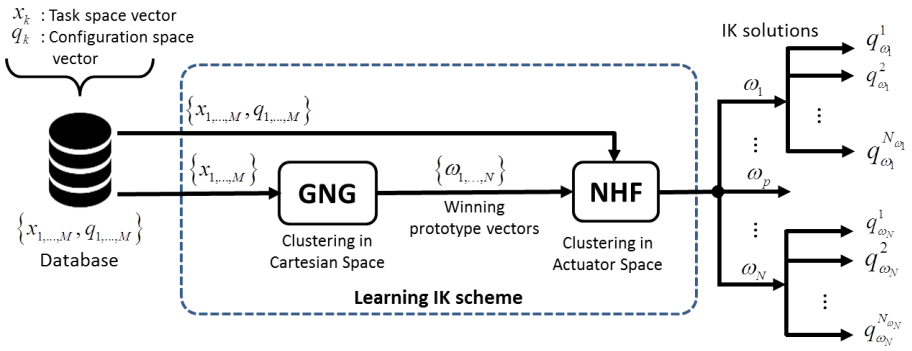


Fig. 2. Proposed IK learning architecture: for a task pose vector that belongs to w_p cluster, a set of N_{w_p} inverse kinematic solutions are associated.

use the last three joints to determine the orientation of the EE. This is done by moving the origin of the frame 0_n to the origin of the frame 0_{n-2}. The position of the spherical wrist will be defined by:

$$P_{n-2} = P_n - d_n \cdot {}^0R_n \cdot k \tag{7}$$

where d_n is the length of the last link, $k = [\,0\ 0\ 1\,]^T$, and ${}^0R_n$ is the rotation matrix of the frame n with respect to the frame 0. P_{n-2} and P_n are the position of the spherical wrist and the EE , respectively. Thus, for the EE position, we have

$$P_n = P_{n-2} + d_n \cdot {}^0R_n \cdot k. \tag{8}$$

As soon as the position of the wrist is known, the FKM can be used to determine the position of the EE.

5. SIMULATION ON KUKA LWR-4+ ROBOT

The proposed learning scheme is simulated on 7-DOF anthropomorphic arm where the last three joints conform a spherical wrist Fig.3. Due to the number of pages limit, this is the sole example case that is presented.

Fig. 3. KUKA LWR-4+ manipulator robot

The learning database is built from the forward kinematic model of the KUKA LWR-4+ and a database size of 50 000 samples is generated. The ranges of input and output spaces are given in Table I.

The IK solutions are obtained by performing the clustering in task and configuration spaces. The dimensions of the task space and configuration space are R^6 and R^7, respectively. GNG consists of several free parameters. However, preliminary tests have shown that only some parameters have a strong influence on the overcome of the training.

Table 1. Ranges of the input and output spaces

Range of joint angles	Range of Cartesian workspace
$-170° \leq q_1 \leq 170^0$	
$-120° \leq q_2 \leq 120^0$	
$-170° \leq q_3 \leq 170^0$	$-0.845m < x < 0.845m$
$-120° \leq q_4 \leq 120^0$	$-0.840m < y < 0.840m$
$-170° \leq q_5 \leq 170^0$	$-0.350m < z < 1.160m$
$-120° \leq q_6 \leq 120^0$	
$-170° \leq q_7 \leq 170^0$	

As a result, only a few parameters were varied within a predefined range based on a search grid to empirically select the best model. The adaptation step λ, the learning rate of best ε_n, the learning rate of neighbors ε_b, and the learning rate of output α were varied during the learning process.

The database is normalized in the range $[0.1, 0.9]$ and is randomly divided in the ratio $70 : 15 : 15$ for training, validation and test set, respectively. The following GNG's parameters $\varepsilon_b = 0.25$, $\varepsilon_n = 0.003$, $\alpha = 0.55$, $\alpha_{\max} = 50$, $d = 0.995$, and $\lambda = 100$ have achieved satisfactory performance. The learning process performed in MATLAB software using an Intel Core $i7 - 2670QMCPU$ at 2.20 GHz took approximately 16 hours. The clustering in configuration space took approximately 5 minutes.

5.1 Derivation of the IK solution of KUKA LWR robot

In the case of the 7-DOFs redundant manipulator, only the second joint (θ_2) is kept fixed. The other joints will be obtained by Paul's method. Other joints can be fixed, but the choice of the first one allows strategies such as obstacle avoidance to be easily implemented.

Let U_0 be the matrix defining the position and orientation of the effector such that

$$U_0 = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The centre of the kneecap located at the origin frame 0_5 and its position denotes ${}^0P_5$ will be defined as follows:

$$P_5 = P_7 - d_7 {}^0R_7 k \tag{9}$$

Equation (9) can be rewritten in the following form:

$$P_5 = \begin{bmatrix} P_{5x} \\ P_{5y} \\ P_{5z} \end{bmatrix} = \begin{bmatrix} p_x - d_7 a_x \\ p_y - d_7 a_y \\ p_z - d_7 a_z \end{bmatrix}. \quad (10)$$

In (10), the origin of the frame 0_7 has been translated to origin 0_5, and the position and orientation matrix at the point 0_5 can be expressed as follows:

$$U_w = \begin{bmatrix} s'_x & n'_x & a'_x & p_{5x} \\ s'_y & n'_y & a'_y & p_{5y} \\ s'_z & n'_z & a'_z & p_{5z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the above, the following equalities hold,

$$U_w \left[\, 0 \;\, 0 \;\, 0 \;\, 1 \,\right]^T = {}^0P_5 = {}^0H_5 \left[\, 0 \;\, 0 \;\, 0 \;\, 1 \,\right]^T. \quad (11)$$

By multiplying the both members of (11) by the inverse of ${}^0H_1$, we get:

$${}^1H_0 \times {}^0H_5 \left[\, 0 \;\, 0 \;\, 0 \;\, 1 \,\right]^T = {}^1H_0 \times U_w \left[\, 0 \;\, 0 \;\, 0 \;\, 1 \,\right]^T. \quad (12)$$

From this equality, we can draw the following system:

$$\begin{cases} c_1 p_{5x} + s_1 p_{5y} = -\left(c_2 c_3 s_4 + s_2 c_4\right) d_5 - s_2 d_3 \\ p_{5z} - d_1 = -\left(s_2 c_3 s_4 - c_2 c_4\right) d_5 + c_2 d_3 \\ s_1 p_{5x} - c_1 p_{5y} = s_3 s_4 d_5 \end{cases} \quad (13)$$

θ_2 being known from the clustering process, the system (13) can still be written as:

$$\begin{cases} c_1 p_{5x} + s_1 p_{5y} = -s_2\left(c_4 d_5 + d_3\right) - c_2 c_3 s_4 d_5 \\ s_1 p_{5x} - c_1 p_{5y} = s_3 s_4 d_5 \\ p_{5z} - d_1 = -s_2 c_3 s_4 d_5 + c_2 c_4 d_5 + c_2 d_3 \end{cases} \quad (14)$$

By squaring and adding each side of the equation system (14), we have :

$$p_{5x}^2 + p_{5y}^2 + \left(p_{5z} - d_1\right)^2 - d_3^2 - d_5^2 = 2 c_4 d_3 d_5 \quad (15)$$

$\rightarrow c_4 = \frac{p_{5x}^2 + p_{5y}^2 + (p_{5z} - d_1)^2 - d_3^2 - d_5^2}{2 d_3 d_5}$,

$\Rightarrow \theta_4 = a\tan 2\left(s_4, c_4\right)$, where $s_4 = \sqrt{1 - c_4^2}$

Knowing θ_4, we can easily calculate θ_3 since θ_2 is known. The third relationship of the system (14) gives:

$$d_1 + c_2 c_4 d_5 + c_2 d_3 - p_{5z} = s_2 c_3 s_4 d_5 \quad (16)$$

in that way, $c_3 = \frac{d_1 + c_2 c_4 d_5 + c_2 d_3 - p_{5z}}{s_2 s_4 d_5}$ and $s_3 = \sqrt{1 - c_3^2}$

$$\Rightarrow \theta_3 = a\tan 2\left(s_3, c_3\right) \quad (17)$$

Therefore, we can determine θ_1, the second relationship of the system (14) yields :

$$s_1 p_{5x} - c_1 p_{5y} = s_3 s_4 d_5 \quad (18)$$

Equation (18) is in the form $A_1 s_1 + A_2 c_1 = A_3$, where $A_1 = p_{5x}$, $A_2 = -p_{5y}$, and $A_3 = s_3 s_4 d_5$. Then, the following expressions can be derived

$s_1 = -\frac{A_1 A_3 \pm A_2 \sqrt{A_1^2 + A_2^2 - A_3^2}}{A_1^2 + A_2^2}$ and $c_1 = -\frac{A_2 A_3 \pm A_1 \sqrt{A_1^2 + A_2^2 - A_3^2}}{A_1^2 + A_2^2}$,

Finally, the first configuration variable can be obtain

$\theta_1 = a\tan 2\left(s_1, c_1\right)$.

Concerning the orientation angles, knowing that ${}^0R_7$is the rotation matrix of U_0, we have

$${}^0R_7 = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix}$$

By pre-multiplying the both sides of the above equation by ${}^4R_0$, we have:

$$\left[\, U \;\, V \;\, W \,\right] = {}^4R_7\left(\theta_5, \theta_6, \theta_7\right) \quad (19)$$

where $U = \left[\, U_x \;\, U_y \;\, U_z \,\right]^T$, $V = \left[\, V_x \;\, V_y \;\, V_z \,\right]^T$, and $W = \left[\, W_x \;\, W_y \;\, W_z \,\right]^T$.

By multiplying each member of Eq.(19) by ${}^5R_4$, we will get the following equality:

$$\begin{bmatrix} c_5 U_x + s_5 U_y & c_5 V_x + s_5 V_y & c_5 W_x + s_5 W_y \\ U_z & V_z & W_z \\ s_5 U_x - c_5 U_y & s_5 V_x - c_5 V_y & s_5 W_x - c_5 W_y \end{bmatrix} = \begin{bmatrix} c_6 c_7 & -c_6 s_7 & -s_6 \\ s_6 c_7 & -s_6 s_7 & c_6 \\ -s_7 & c_7 & 0 \end{bmatrix}.$$

It follows, after identifying both sides of this equality, that $s_5 W_x - c_5 W_y = 0$, hence,

$$\theta_5 = a\tan 2\left(-W_y, -W_x\right) \quad (20)$$

we also have the following equality

$$\begin{cases} c_6 = W_z \\ s_6 = -c_5 W_x - s_5 W_y \end{cases}, \text{ hence}$$

$$\theta_6 = a\tan 2\left(s_6, c_6\right) \quad (21)$$

Finally, $\begin{cases} c_7 = s_5 V_x - c_5 V_y \\ s_7 = -s_5 U_x + c_5 U_y \end{cases}$, hence

$$\theta_7 = a\tan 2\left(s_7, c_7\right) \quad (22)$$

5.2 simulation results

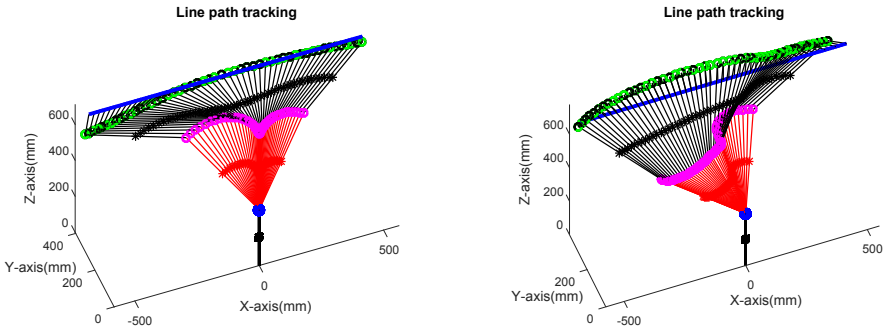
The simulations are performed in MATLAB software using an Intel Core $i7 - 2670QMCPU$ at 2.20 GHz.

Fig. 4 represents a linear path tracking where the spherical wrist is oriented down and upwards in Fig. 4-(a) and Fig. 4-(b), respectively. The circular path on the $X-Z$ plane is tracked in Fig. 5. The tracking with simple configuration vectors is represented in Fig. 5-(a) while the tracking with complex configuration vector is shown in Fig. 5-(b). These paths can be followed without error with several other configuration vectors. Others scenarios such that obstacles avoidance and joint limit avoidance can also be implemented.

The various simulations carried out to validate the proposed learning scheme have shown satisfactory results. It can be applied to any redundant manipulator and regardless of its number of DOFs.

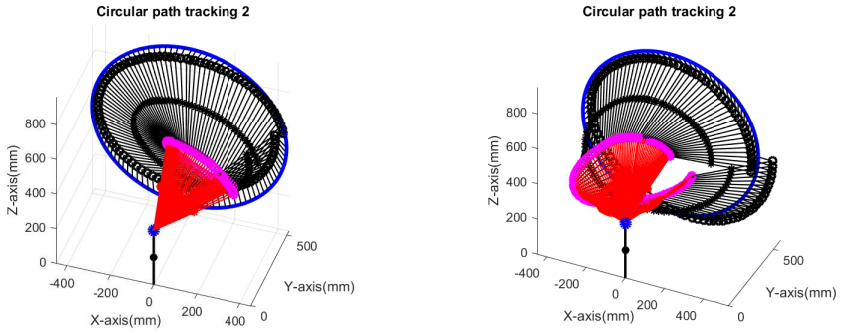
6. CONCLUSION

In this paper, a novel learning scheme that can learn redundant solutions of redundant manipulators has been proposed. Redundant solutions were maintained by performing clustering in the workspace and the configuration space of the redundant manipulator. The growing neural



a) b)

Fig. 4. Linear path tracking: a): a line path tracking with with the wrist pointing upwards, b): a line path tracking with with the wrist pointing down.



a) b)

Fig. 5. Circular path tracking 2: a): a circular path tracking with simple configuration vectors, b): a circular path tracking with complex configuration vectors.

gas network has been implemented in workspace clustering while a neighborhood function has been introduced in configuration space clustering. The simulations carried out have yielded satisfactory results in terms of IK solving, redundancy preservation and resolution. In a future work, the IK problem redundant mobile manipulator will be investigated.

REFERENCES

Ananthanarayanan, H. and Ordóñez, R. (2015). Real-time inverse kinematics of (2n+ 1) dof hyper-redundant manipulator arm via a combined numerical and analytical approach. *Mechanism and Machine Theory*, 91, 209–226.

Fritzke, B. (1995). A growing neural gas network learns topologies. In *Advances in neural information processing systems*, 625–632.

Hollerbach, J.M. (1985). Optimum kinematic design for a seven degree of freedom manipulator. In *Robotics research: The second international symposium*, 215–222. Cambridge, MIT Press.

Klein, C.A., Chu-Jenq, C., and Ahmed, S. (1995). A new formulation of the extended jacobian method and its use in mapping algorithmic singularities for kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, 11(1), 50–55.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9), 1464–1480.

Kumar, S., Premkumar, P., Dutta, A., and Behera, L. (2010). Visual motor control of a 7dof redundant manipulator using redundancy preserving learning network. *Robotica*, 28(6), 795–810.

Martinetz, T., Schulten, K., et al. (1991). A" neural-gas" network learns topologies.

Peiper, D.L. (1968). The kinematics of manipulators under computer control. Technical report, Stanford Univ Ca Dept of computer science.

Qin, A.K. and Suganthan, P.N. (2004). Robust growing neural gas algorithm with application in cluster analysis. *Neural networks*, 17(8-9), 1135–1148.

Raja, R., Dutta, A., and Dasgupta, B. (2019). Learning framework for inverse kinematics of a highly redundant mobile manipulator. *Robotics and Autonomous Systems*, 120, 103245.

Singh, G.K. and Claassens, J. (2010). An analytical solution for the inverse kinematics of a redundant 7dof manipulator with link offsets. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2976–2982. IEEE.

Wei, Y., Jian, S., He, S., and Wang, Z. (2014). General approach for inverse kinematics of nr robots. *Mechanism and Machine Theory*, 75, 97–106.

Zaplana, I. and Basanez, L. (2018). A novel closed-form solution for the inverse kinematics of redundant manipulators through workspace analysis. *Mechanism and Machine Theory*, 121, 829–843.