

An ANN Approach for the Motion Planning of Redundant Manipulators

Rene V. Mayorga
Industrial Systems Engineering
University of Regina
Regina, Saskatchewan
Canada

Tanaporn Chayangkul Na Aduthaya
Industrial Systems Engineering
University of Regina
Regina, Saskatchewan
Canada

Abstract — In this article an Artificial Neural Network (ANN) approach for the motion planning of redundant robot manipulators is presented. The approach is based on formulating an inverse kinematics problem under an inexact context. This procedure permits to deal with the avoidance of obstacles with an appropriate and easy to compute null space vector; whereas the avoidance of singularities is attained by the proper pseudoinverse perturbation. Here the computation of the inverse kinematics problem is performed by a properly trained ANN and including a null space vector for obstacle avoidance which is also calculated by another properly trained ANN. The approach is tested on the simulation of a planar redundant manipulator performing some obstacle avoidance tasks. From the results obtained, the approach compares favorably with the numerical approach.

Keywords — ANNs; Motion Planning; Obstacle Avoidance; Redundant Robot Manipulators

I. INTRODUCTION

One of the most challenging problems that frequently arises in the field of robot manipulators is the trajectory/path planning problem. A general formulation of this problem usually includes the obstacle avoidance as a sub-problem. The trajectory/path planning of robot manipulators can be performed either in the joint space [9]; or in the Cartesian space [9]. Planning paths in the configuration space has the advantage that by establishing some via joint configurations it is possible to obtain smooth motions while achieving certain tasks. However, this approach has the disadvantage that it does not provide the means for the end effector to follow a desired trajectory. Alternatively, the planning in the operational space has the property that certain tasks requiring specific end-effector trajectories motions can be achieved. Nevertheless, this approach has the main disadvantages that due to the presence of singularities not all the trajectories in the operational space can be realized. Hence, the planning of paths and trajectories depends mainly on the assigned task. Therefore, is necessary to distinguish some different problems and preferably formulate (solve) them as an optimization problem [9], and also properly consider an inverse kinematics problem [7].

The inverse kinematics problem for redundant manipulators is usually studied under an *exact context*, where a precise solution is required, by global or local approaches. The global methods are limited to off-line programming and are not easily implementable in sensor based environments. Therefore, considerable efforts have been devoted to develop real-time local approaches [9]. Nevertheless, singularities-safe paths cannot always be ensured [9].

An alternative way to deal with the inverse kinematics problem, which can also be applied to non-redundant manipulators, is to consider the problem under an *inexact context* [7]. Under this approach the exactness to a solution is relaxed and either [7]: (a) the problem is reduced to one of priority subtasks; or (b) a damped least squares problem is considered. The damped least squares approach has the advantages of producing a subtle (temporary) degradation of the end effector near a singularity, and still be able to utilize the redundant degrees of freedom for the avoidance of obstacles by the proper use of the null space vector [7]. However, the approach in some cases may be computational expensive even for a few and simple obstacles. Therefore, there is still a need to improve this approach.

In this article a novel Artificial Neural Network (ANN) approach for the obstacle avoidance of redundant robot manipulators is presented. The approach is based (as in [8]) on formulating an inverse kinematics problem under a local *inexact context*, which permits to deal with the avoidance of obstacles with an appropriate and easy to compute null space vector; whereas the avoidance of singularities is attained by the proper pseudo-inverse perturbation. However, here the null space vector for obstacle avoidance is computed by a properly trained off-line ANN. Furthermore, once properly trained, the ANN can be used in real time to compute the null space vector and be included in the computation of the inverse kinematics which is also performed by another properly trained ANN, [6], [7]. Here, the proposed overall ANN approach is successfully tested on the obstacle avoidance of a planar redundant robot manipulator performing some benchmark tests. Here it is demonstrated that the proposed ANN approach compares favorably with its corresponding conventional numerical local version.

II. THE INVERSE KINEMATICS PROBLEM

One of the main problems that arise when studying the kinematics aspects of a robot manipulator is the so-called inverse kinematics problem. For a given manipulator, at any instant of time, a configuration of joints establishes a unique position and orientation of the end effectors in the Cartesian space. Formally, consider a manipulator with n degrees of freedom. At any instant of time, denote variables by $\theta_i \equiv \theta_i(t)$; $i = 1, 2, \dots, n$. Also, define the manipulation variables describing the robot tasks by a vector of m variables $X_j \equiv X_j(t)$; $j = 1, 2, \dots, m$. Furthermore, let $m \leq n$; both for the case $m = n$ (nonredundant manipulator) and $m < n$ (redundant manipulator). Finally, let $t_i \in [t_0, t_f]$ where t_0 and t_f are the initial and final time of the task interval, and let \mathfrak{R}^m and \mathfrak{R}^n be the m -dimensional and the n -dimensional Euclidean spaces respectively, such that $X \equiv X(t) = [X_1, X_2, \dots, X_m]^T \in \mathfrak{R}^m$ and $\theta \equiv \theta(t) = [\theta_1, \theta_2, \dots, \theta_n]^T \in \mathfrak{R}^n$ are related by

$$X(t) = f(\theta(t)). \quad (1)$$

In general, this relation is nonlinear and pointwise, hence an analytical inverse relationship cannot be obtained. By differentiating (1) with respect to time and defining

$$\dot{X} \equiv \dot{X}(t) = \frac{dX(t)}{dt}, \dot{\theta} \equiv \dot{\theta}(t) = \frac{d\theta(t)}{dt}, \quad (2)$$

$$J(\theta) \equiv J(\theta(t)) = \frac{\partial f(\theta(t))}{\partial \theta(t)}; \quad (3)$$

The following equation is obtained:

$$\dot{X}(t) = J(\theta(t))\dot{\theta}(t). \quad (4)$$

From this equation, it is possible to calculate a $\theta(t)$ path in terms of a prescribed trajectory $X(t)$. However, a non-singular $J(\theta(t))$ for all $t_i \in [t_0, t_f]$ cannot be fully guaranteed and cyclic behavior cannot be assured [9].

An alternative approach to deal with the problem of singularities, also applicable to nonredundant manipulators, is to consider the problem under an *inexact context*. Under this formulation the exactness to the solution of (4) is relaxed and a damped least squares problem [7] is considered. Following [6], [7], a solution in an inexact context is given by

$$\dot{\theta} = J_{wz\delta}^+(\theta) \dot{X} + [I - J_{wz\delta}^+(\theta)J(\theta)]v; \quad (5)$$

Where the weighted-perturbed pseudoinverse matrix $J_{wz\delta}^+(\theta)$ is given by:

$$J_{wz\delta}^+(\theta) = W^{-1}J^T(\theta)[J(\theta)W^{-1}J^T(\theta) + \delta Z^{-1}]^{-1} \quad (6)$$

$$= [J^T(\theta)ZJ(\theta) + \delta W]^{-1}J^T(\theta)Z; \quad (7)$$

where, the positive definite symmetric matrices W , Z , act as metrics to allow invariance to frame reference and scaling (homogenize dimensions). In some cases, W is usually a diagonal matrix to deal with joint limits and/or different joint units. Also, I is the identity matrix, $\rho_i \geq 0$ is a damping

factor, and $v(t_i)$ is an arbitrary vector intended for obstacle or singularities avoidance/prevention [7]. It can be easily shown that a suitable null space vector is given by:

$$v \equiv -\frac{1}{2} \left(\left[\frac{\partial r(\theta, \dot{\theta})}{\partial \dot{\theta}} \right] - \sum_{k=1}^N \lambda_k \left[\frac{\partial g_k(\theta, \dot{\theta})}{\partial \dot{\theta}} \right] \right). \quad (8)$$

where $r(\theta, \dot{\theta})$ is a side (scalar) criterion that is also desired to be optimized; $g_k(\theta, \dot{\theta}) - \epsilon_k \geq 0$, $k = 1, 2, \dots, N$, $\epsilon_k > 0$, are some constraints to be satisfied; and λ_k are scalars. Setting $\lambda_k > 0$ implicitly implies that a constraint is binding, i.e. $g_k(\theta, \dot{\theta}) - \epsilon_k = 0$.

Furthermore, suppose that it is desired to decrease monotonically a set of functions $s_j(\theta)$; $j = 1, 2, \dots, M$; over the time interval $[t_0, t_f]$. This can be achieved by selecting $r(\theta, \dot{\theta}) = \sum_{j=1}^M \beta_j s_j(\theta, \dot{\theta}) = \sum_{j=1}^M \beta_j \left[\frac{\partial s_j(\theta)}{\partial \theta} \right]^T \dot{\theta}$; where β_k is an appropriate scalar [3], [7], [8]. Therefore,

$$v \equiv -\frac{1}{2} \sum_{j=1}^M \beta_j \left[\frac{\partial s_j(\theta)}{\partial \theta} \right] + \sum_{k=1}^N \lambda_k \left[\frac{\partial g_k(\theta, \dot{\theta})}{\partial \dot{\theta}} \right] \quad (9)$$

III. A NULL SPACE VECTOR FOR OBSTACLE AVOIDANCE

In this article, the null space vector is used for the obstacle avoidance of redundant robot manipulators. The concept of explicitly maximizing some areas between the links and the obstacle is considered as in [8]. The notion of volumes and areas determined by a set of vectors in \mathfrak{R}^p are addressed as in [8] by properly defining a q -box in \mathfrak{R}^p ; for $q \leq p$. The following proposition is easily shown in [2].

Proposition 1 Let $q \leq p$. and $b_1, \dots, b_q \in \mathfrak{R}^p$, be a set of independent vectors. Then the volume of the q -box in \mathfrak{R}^p is given by

$$V = \{\det(B^T B)\}^{\frac{1}{2}} \quad (10)$$

where B is the $p \times q$ matrix containing the b_j ; $j = 1, 2, \dots, q$ as its j -th column vector.

For the particular, yet important, case that $q = p$ the following Corollary it is also easily shown in [2].

Corollary 1 Let $q = p$, and $b_1, \dots, b_q \in \mathfrak{R}^p$, be a set of independent vectors. Then the volume of the q -box in \mathfrak{R}^q is determined by

$$V = |\det B| \quad (11)$$

where B is the $p \times q$ matrix containing the b_j ; $j = 1, 2, \dots, q$ as its j -th column vector.

The Proposition 1 constitutes the basis of the proposed approach for obstacle avoidance, [8]. That is, it is intended to properly fit the vectors b_1, \dots, b_q ; $q = 2, 3$; between the

manipulator links and a specified obstacle. Hence, from the explicit volume expression a side criterion, to be maximized by (5), can be easily obtained. The resultant procedure has the property that while the end effector tracks a specified trajectory, the links tend to stay away from the obstacle. This approach is similar to the one developed in [7]. However, here instead of using the null space vector to avoid singularities; it is utilized for obstacle avoidance.

The approach can be explained better by considering the case in which $q = p = 2$. In this case, the volume given by (4) in fact constitutes the area of a parallelogram. Now, consider the Fig. 1; where a planar redundant ($m = 2, n = 3$) manipulator, and obstacle are depicted. Hence, according to Corollary 1, the area of the corresponding (parallelogram) k -th q -box is given by

$$V_k = |\det B_k|; \quad k=1, 2, 3; \quad (12)$$

where the, 2×2 , B_k matrix contains $b_1^{(k)}, b_2^{(k)}$ as column vectors. Notice that the vectors $b_2^{(1)}, b_2^{(2)}, b_2^{(3)}$ depend on $\theta_1, (\theta_1, \theta_2), (\theta_1, \theta_2, \theta_3)$ respectively. Hence, it becomes natural to consider half of the parallelogram area; that is, only the area of the triangle determined by the vectors $b_1^{(k)}$ and $b_2^{(k)}$. Then, the side function to be maximized is given by

$$s(\theta) = \frac{1}{2}[V_1 + V_2 + V_3]. \quad (13)$$

Then, according to (9), the i -th component of the vector $v(\theta)$ intended for obstacle avoidance, is given by

$$v_i(\theta) = \frac{1}{4} \left[\frac{\partial V_1}{\partial \theta_i} + \frac{\partial V_2}{\partial \theta_i} + \frac{\partial V_3}{\partial \theta_i} \right]; \quad i = 1, 2, \dots, n. \quad (14)$$

Notice that each $V_k |k=1,2,3$ is given by (12) and this equation above is not continuously differentiable at the point that $\det(B_k) = 0$. However, this problem can be dealt with by properly defining:

$$\frac{\partial V_k}{\partial \theta_i} \Big|_{\substack{k=1,\dots,3 \\ i=1,\dots,n}} \equiv \begin{cases} \frac{D_k}{|D_k|} \frac{\partial D_k}{\partial \theta_i} & \text{If } D_k \neq 0 \\ 0 & \text{Otherwise;} \end{cases} \quad (15)$$

where $D_k \equiv \det(B_k)$.

In the case in which $q = p = 2$, it is easy to express explicitly each $D_k \equiv \det(B_k); k = 1, 2, 3$; in terms of $\theta_1, \theta_2, \theta_3$. Hence, the computation of (14) is a simple and fast one.

From Fig. 1, it can be easily observed that $P^{(k-1)} + b_1^{(k)} = O^{(k)}$. Then, $b_1^{(k)} = O^{(k)} - P^{(k-1)}$. Therefore, $D_k = \det[O^{(k)} - P^{(k-1)}, b_2^{(k)}] = \det[O^{(k)} - P^{(k-1)}, P^{(k)} - P^{(k-1)}] = [(O_x^{(k)} - x^{(k-1)})(y^{(k)} - y^{(k-1)}) - (O_y^{(k)} - y^{(k-1)})(x^{(k)} - x^{(k-1)})]$

where $x^{(k)}, y^{(k)}$ are the vector $b_2^{(k)}$ components and $O_x^{(k)}, O_y^{(k)}$ are the coordinates of the obstacle point $O^{(k)}$; $k=1,2,3$. However, notice that each $x^{(k)}, y^{(k)}$ in turn can be explicitly expressed in terms of the joint coordinates.

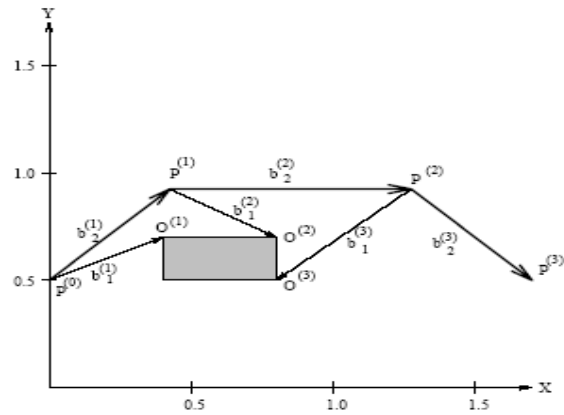


Figure 1: Example of an independent set of vectors b_i^k s.

Therefore from the equation (14) and (15), the components $V_i |i=1,2,\dots,n$; of the null space vector intended for obstacle avoidance can be easily evaluated in explicit form. The numerical approach, developed in [8] resolves redundancy at the velocity level using smooth null space vector components which are effective near and far from the obstacles; however here, the null space vector is effectively computed using an ANN for true real-time operation.

IV. A NUMERICAL PROCEDURE FOR THE COMPUTATION OF THE INVERSE KINEMATICS

First, it is important to mention here the following fact. In general most current industrial robot manipulators operate in a non-redundant fashion, that is, $m = n = 6$. However, by specifying $m < n$, say $m = 3$ ($n = 6$) for positional tasks, some degrees of freedom become redundant and the manipulator can be treated as such.

The attempt to solve the inverse kinematics problem by direct use of (5), including the damping factor, may be cumbersome due to the large amount of computation required for the evaluation of the pseudoinverse matrix.

Suppose that at iteration i the damping factor δ_i as given by a proper scheme (such as in [7]), and the vector v intended for obstacle avoidance as in (14), are available. Then solve for φ the following system

$$[J(\theta)J^T(\theta) + \delta_i I]\varphi = \dot{X} - J(\theta)v \quad (17)$$

by a Gaussian elimination process that takes into account the symmetry of the matrix as described in [7]. Finally, just compute $\dot{\theta}$ by

$$\dot{\theta} = J^T(\theta)\varphi + v \quad (18)$$

This simpler approach **may not** yet conduce to real time implementations. Hence, here a novel Artificial Neural Network approach is proposed for true real-time operation. This approach is still based on (17) and (18); but unlike other recent approaches, [1], [4], [5], [10], it is still quite simple and effective as it is shown in a subsequent section.

V. AN ARTIFICIAL NEURAL NETWORK APPROACH

Here, Artificial Neural Networks are used to “learn” the non-linear relationships given by Eq. (14) and (15); and (17) and (18). The approach is somewhat similar to the ANN approach presented in [7], where the null space vector is used for singularities avoidance; however, here the null space vector is used for obstacle avoidance.

Now, notice that the null space vector given by (14) and (15) can be easily computed using a properly trained ANN. In this case, the $\theta_1, \dots, \theta_n$ and the obstacle points O are the inputs to the ANN, and the v_1, \dots, v_n are the ANN outputs. The training is done off-line; however, once the ANN learns the null space vector for obstacle avoidance; it can be easily used in real-time.

Next, from the equation (17), let $m < n$ and solve for φ which is an $n \times m$ matrix. Then, just set $\dot{\theta}$ as in (18). In this case φ is computed off-line by a trained neural network (such as a multi-layered feedforward with backpropagation) with inputs θ , such that, [7]:

$$\varphi = J^T(\theta)A^{-1} \quad (19)$$

where $A = [J(\theta)J(\theta)^T + \delta I]$, (20)

Next set, $\dot{\theta}$ as

$$\dot{\theta} = \varphi[\dot{X} - J(\theta)v] + v \quad (21)$$

Finally, a step by step procedure is performed as in [7]:

- (a) Specify an initial set of plant variables $\theta(t_0)$;
- (b) Consider a specified task $\dot{X}(t_i)$;
- (c) Apply $\theta(t_i)$; to a trained neural network to obtain $\varphi(t_i)$;
- (d) Set $\dot{\theta}(t_i)$ as in (21);
- (e) Apply an efficient numerical integration (such as a fourth order Runge-Kutta) to get $\theta(t_{i+1})$;
- (f) Set $t_{i+1} = t_i + \Delta t$;
- (g) If $t_{i+1} = t_f$ stop; otherwise set $i = i + 1$ and return to step (b).

Here, the entire ANN approach is implemented for the obstacle avoidance of a redundant planar manipulator [7], [8]. In this case in order to achieve reasonable accurate results, the ANNs must be properly trained with the right number of training data set which is limited to fit the desired ranges. Once the ANNs are trained with the required range, they can be easily and effectively used to compute on-line in Eqs. (4), (19), and (21); resulting in a true real time obstacle avoidance approach.

VI. SIMULATION RESULTS

The proposed approach described in the previous section, is implemented in the simulation of a 3 DOF planar redundant manipulator performing some “benchmark tests” as in [8]. Notice that effective obstacle avoidance in 2D with only one redundant DOF for obstacle avoidance is in fact more difficult to achieve than in 3D with one or more redundant DOFs; due to the fact that in 2D with only one redundant DOF, the situation is quite restrictive. The links are specified in meters as

$L_0 = 0.5, L_1 = 0.6, L_2 = 0.85$, and $L_3 = 0.6$. Notice that in this case in the equation (20), $\delta = 0$. Here the following tasks are considered.

Task 1: Given an initial end-effector position

$$[X_1(t_0), X_2(t_0)] = [x^{(3)}(t_0), y^{(3)}(t_0)] = [1.555883, 1.014850]$$

with a corresponding initial configuration (rads.) specified by

$$[\theta_1(t_0), \theta_2(t_0), \theta_3(t_0)] = \left[\frac{\pi}{12}, \frac{\pi}{2}, -\frac{\pi}{6}\right]$$

Notice that in this task the initial manipulator configuration has the third link upwards presenting a difficulty for obstacle avoidance approaches, [8]. Also, notice that the considered tasks consist on moving the manipulator from an initial end effector position along a straight line with constant speed and $t_f = 10$ seconds, and trying to move away from the obstacle by the effect of the null space vector to a final end effector position $[X_1(t_f), X_2(t_f)] = [0.60, 0.20]$

The proposed ANN approach is implemented on a Silicon Graphics Octane digital computer. Here the Matlab Neural Networks Toolbox is used as a platform. In this particular case, in order to get more efficient results, the input and output data are first preprocessed by restricting between a learnable range. Then a feed-forward network is created by using the function NEWFF as the 3-layer-network of 35-14-2 for (19) and a 3-layer-network of 25-14-3 for the null space vector. In this proposed approach, the TANSIG and PURELIN transfer functions are used and also the TRAINLM training function is employed in the training process. Then the weights and biases are calculated for the Network. Once ANNs are trained, the propagation of an initial solution is obtained by (21). Then, the integration process is carried out (in the training as well as in the actual Task simulation) by a fourth order Runge-Kutta method with an integration step-size of 0.1.

First the task is simulated by the numerical approach [8] with no provisions for obstacle avoidance ($v(t) \equiv 0$). The results are shown on Fig. 2, where a collision is clearly seen. Next, the same task is performed with the vector $v(t)$ for obstacle avoidance as given by (14), with $\lambda = 4.0$. the results obtained using the numerical approach [8], are shown in Fig. 3, where it is clearly seen that the obstacle avoidance has been successful. Here the proposed ANN approach is performed for obstacle avoidance based on (14) and (19), three cases can be considered:

1. Computing φ by using a trained ANN and v by using a numerical method
2. Computing φ by using a numerical method and v by using a trained ANN
3. Determining both φ and v by using trained ANNs

Case 1: Here, φ is approximated by the proposed ANN approach while v is calculated by a numerical approach. The results are shown in the Fig. 4, where it is clearly seen that the

manipulator moves in a similar way as using a numerical method.

Case 2: The proposed ANN approach approximates v while φ is calculated by a numerical approach. Then, the obtained results using the proposed approach are shown in Fig. 5.

Case 3: In this case, both φ and v are approximated by the proposed ANN approach. Also in this case the task is successfully performed as the results shown in Fig. 6 .

Finally, one more task is also performed. This task is similar to the first task with the difference that now the given an initial end effector position is

$$[x_1(t_0), x_2(t_0)] = [1.495943, 0.559559]$$

with an initial configuration given by

$$[\theta_1(t_0), \theta_2(t_0), \theta_3(t_0)] = \left[\frac{\pi}{12}, \frac{\pi}{2}, \frac{\pi}{6} \right]$$

In this case the task is performed with no provisions for obstacle avoidance, and with $v(t)$ as in (14), using the numerical approach as shown in Fig. 7 and Fig. 8 respectively.

Also three cases are considered. This task is also effectively performed. The obtained results based on Case 1, Case 2, and Case 3, are respectively shown in Fig. 9, Fig. 10, and Fig. 11. From these figures; the success of the proposed ANN approach can be clearly observed.

VII. CONCLUSIONS

In this article an Artificial Neural Network approach for the obstacle avoidance of redundant robot manipulators has been presented. This approach permits the realization of fast and robust real-time algorithms for obstacle avoidance. Here a null space vector for obstacle avoidance is computed with a properly trained ANN. Then, this vector is properly used in the inverse kinematics computation which is also performed with another properly trained ANN. This article, demonstrates that the use of properly trained Artificial Neural Networks for the obstacle avoidance of redundant robot manipulators compares favorably with a conventional numerical approach. Here, the proposed ANN approach it is demonstrated by its successful application for the obstacle avoidance of a planar redundant manipulator performing some “benchmark tests”.

REFERENCES

[1] Abde A., “Trajectory Planning for Flexible Cartesian Robot Manipulator by Using ANN: Numerical Simulation and Experimental Verification”, *Robotica*, Vol 29, No. 5, December 2010.

[2] Fraleigh J. B., Beauregard R.A., “Linear Algebra,” Addison-Wesley Publishing Co., Second Edition, 1990.

[3] Carrera J., Mayorga R. V., “An Artificial Neural Network Approach for Motion Coordination of Hyper Redundant Articulated Systems”, *Proc. IEEE-RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, October 29 – November 2, 2007.

[4] Chen Y., Li B., “A Piecewise Acceleration-Optimal & Smooth-Jerk Trajectory Planning Method for Robot

Manipulator Along a Predefined Path”, *Intl. J. Adv. Robotic Systems*, Vol. 8, No. 4, 2011.

[5] Dubey A. D., Mishra R. B., Jha A. K., “Task Time Optimization of a Robot Manipulator using Artificial Neural Network and Genetic Algorithm”, *Intl. Journal of Computer Applications*, Vol. 51, No. 13, August 2012.

[6] Mayorga R. V., Carrera J., "A Radial Basis Function Network Approach For The Computation of Inverse Time Variant Functions", *International Journal of Neural Systems*, Vol. 17, No. 3, June 2007.

[7] Mayorga R. V., Sonongboon P., “An Artificial Neural Networks Approach for Inverse Kinematics and Singularities Prevention of Redundant Manipulators”, *Journal of Intelligent & Robotic Systems*, Vol. 44, December, 2005.

[8] Mayorga R. V., Janabi-Sharifi F., Wong A.K.C., “A Fast Approach for the Robust Trajectory Planning of Redundant Robot Manipulators”, *J. Robotic Systems*, Vol. 12, No. 2, pp. 147 – 161, 1995.

[9] Mayorga R.V., Wong A.K.C., “An Efficient Local Approach For The Optimal Path Generation of Robot Manipulators,” *Journal of Robotic Systems*, Vol. 7(1), Feb. 1990.

[10] Nil M., et al, “Fuzzy Neural Network Based Intelligent Controller for 3DOF Robot Manipulators”, *Proc. Intl. Symp. On Manufacturing Systems*, May 29-31, 2006.

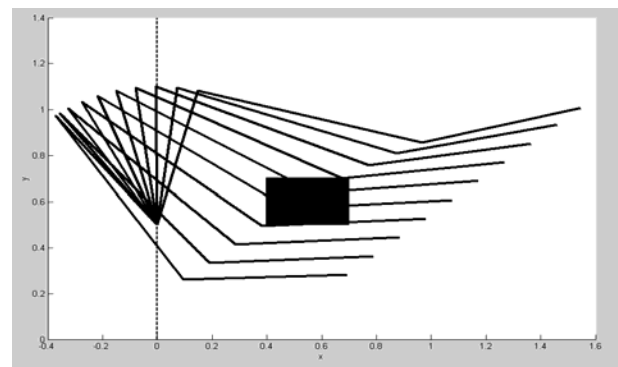


Fig. 2: Task 1: Entire numerically approach. No provision for obstacle avoidance.

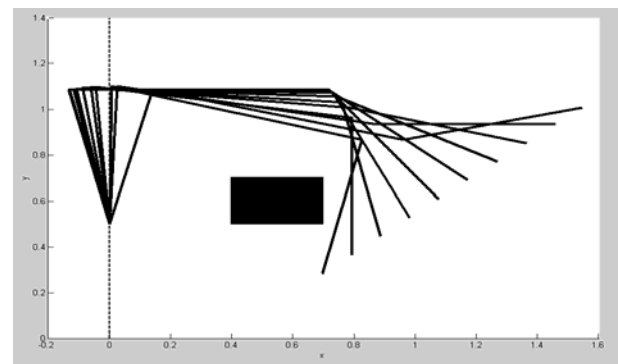


Fig. 3: Task 1: Numerical approach with v for obstacle avoidance

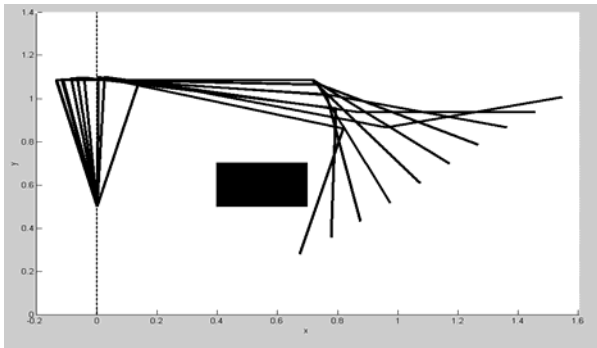


Fig. 4: Task 1: φ by the proposed ANN approach and v numerically for obstacle avoidance.

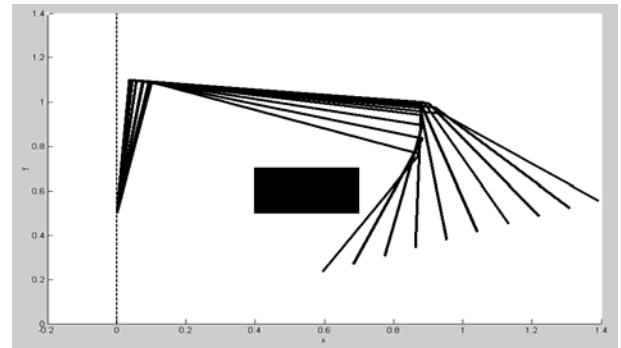


Fig. 8: Task 2: Numerical approach for obstacle avoidance.

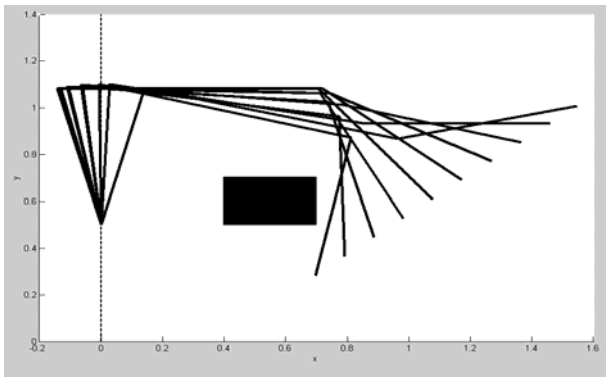


Fig. 5: Task 1: φ numerically, and v by the proposed ANN approach for obstacle avoidance.

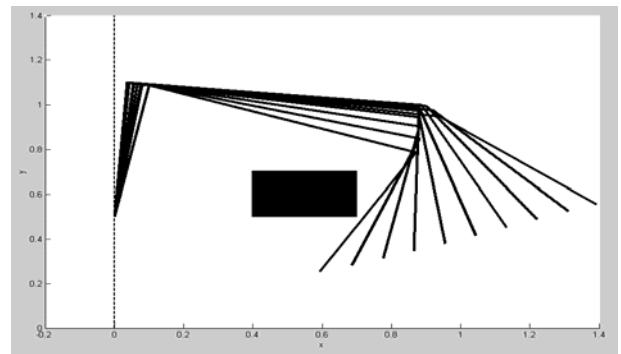


Fig. 9: Task 2: φ by the proposed ANN approach and v by the numerical approach for obstacle avoidance.

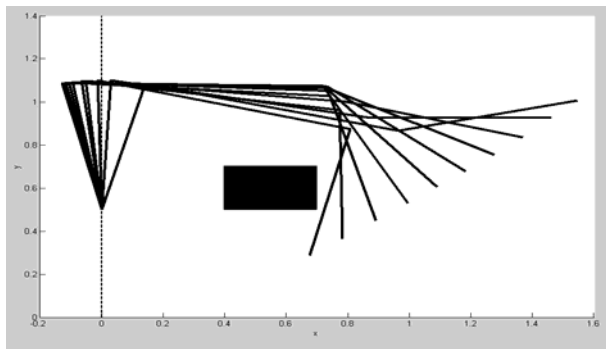


Fig. 6: Task 1: Both φ and v by the proposed ANN approach

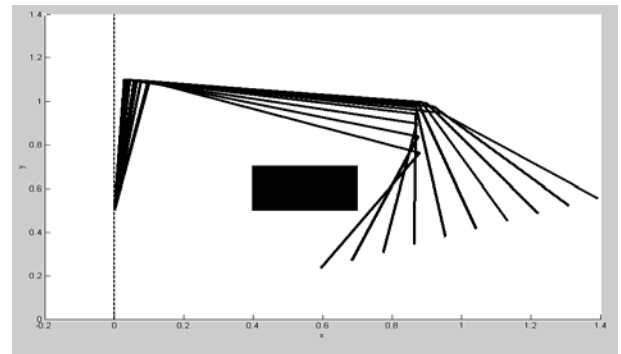


Fig. 10: Task 2: φ by the numerical approach and v by the proposed ANN approach for obstacle avoidance.

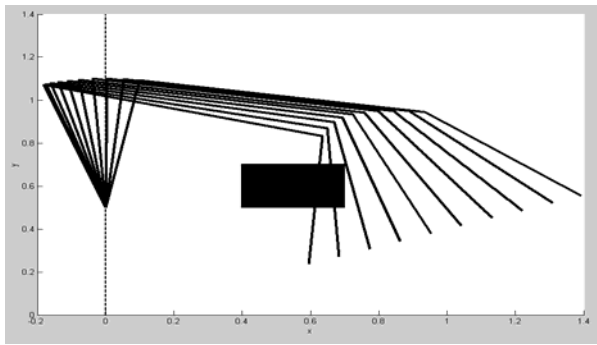


Fig. 7: Task 2: No provision for obstacle avoidance.

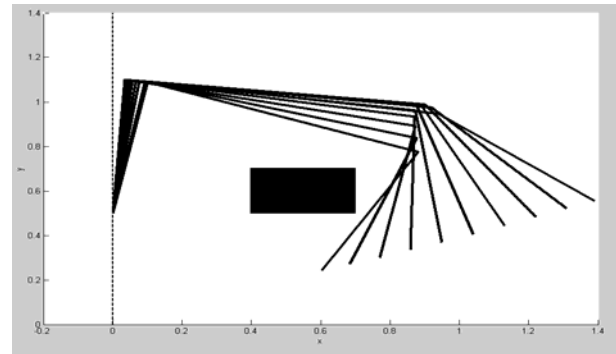


Fig. 11: Task 2: Both φ and v by proposed ANN approach.