$\mu_g(P)$ is the membership grade of P to belong to the green region and $\mu_b(P)$ is the membership grade of P to belong to the blue region. Values of $\mu_r(P)$, etc. may be determined the way described in Section II.

As far as we know fuzzy turbulence is a completely new concept, introduced only in this paper. To be more realistic this model should be three dimensional or if we also want to incorporate time into it, it will be a four dimensional space time model. In this four dimensional space the location of the center of a vortex will be denoted by $(x,\ y,\ z)$ at some instant t. The vortex will be of cylindrical shape, whose axis may make angles $\alpha,\ \beta$ and γ respectively with $x,\ y$ and z axes. Obviously, $\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$. To describe the vortex at $(x,\ y,\ z)$ a local cylindrical coordinate system $(r,\ \theta,\ h)$ is necessary. Also another parameter m as described in Section III is needed. r will be expressed in terms of m and θ. So to make a more realistic model of this type of turbulence we will need the following independent parameters: $x,\ y,\ z,\ t$, any two out of $\alpha,\ \beta$ and γ and m, θ and h, i.e., nine in all. Modeling of turbulence involving all these nine independent parameters (some of them including m are fuzzy numbers) will be the subject of future research.

The author has recently proposed a model of generation of intense tropical storms over the seas in terms of fuzzy dynamical systems [23]. It has long been suspected that atmospheric turbulence plays an important role behind generation of such storms. The model of turbulence proposed here and the model of generation of storms proposed in [23] have one very common aspect, namely the "fuzzy vortex." This, at least to some extent, supports the validity of the hypothesis of generation of storms from atmospheric turbulence. In near future the author has plans to work toward a modeling of generation of storms from atmospheric turbulence.

ACKNOWLEDGMENT

All the simulation works in this paper have been performed using MATLAB. Reference [7] has turned out to be very useful in this regard. I would also like to acknowledge Prof. H. P. Majumdar for his appreciation and encouragement. The role of the comments of three anonymous referees in improving the paper is most sincerely acknowledged.

REFERENCES

[1] M. F. Barnsley, *Fractals Everywhere*. New York: Academic, 1988.
[2] R. A. Brown, *Fluid Mechanics of the Atmosphere*. New York: Academic, 1991.
[3] C. A. Cabrelli, B. Forte, U. M. Molter, and E. R. Vrscay, "Iterated fuzzy sets systems: A new approach to the inverse problem for fractals and other sets," *J. Math. Anal. Applicat.*, vol. 171, pp. 79–100, 1992.
[4] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*. Reading, MA: Addison-Wesley, 1989.
[5] D. Dutta Majumder and K. K. Majumdar, "Complexity analysis, uncertainty management and fuzzy dynamical systems: A cybernetic approach with some case studies," Kybernetes, submitted for publication.
[6] G. A. Edgar, *Measure, Topology, and Fractal Geometry*. New York: Springer-Verlag, 1990.
[7] L. V. Fausett, *Applied Numerical Analysis Using MATLAB*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
[8] J. E. Hutchinson, "Fractals and self similarity," *Indiana Univ. Math. J.*, vol. 30, pp. 713–747, 1981.
[9] T. Kapitaniak and S. R. Bishop, *The Illustrated Dictionary of Nonlinear Dynamics and Chaos*. New York: Wiley, 1999.
[10] S. L. Loney, *An Elementary Treaties on the Dynamics of a Particle and of Rigid Bodies*. Madras, India: Macmillan India, 1972.
[11] K. K. Majumdar, "One dimensional fuzzy differential inclusions," *J. Intell. Fuzzy Syst.*, vol. 13, pp. 1–5, 2003.
[12] B. B. Mandelbrot, *The Fractal Geometry of Nature*. New York: Freeman, 1982.
[13] J. Milnor, "On the concept of attractor," *Comm. Math. Phys.*, vol. 99, pp. 177–195, 1985.
[14] H. T. Nguyen, "A note on the extension principle for fuzzy sets," *J. Math. Anal. Applicat.*, vol. 64, pp. 369–380, 1978.
[15] H.-O. Peitgen, H. Jurgens, and D. Saupe, *Chaos and Fractals: New Frontiers of Science*. New York: Springer-Verlag, 1992.
[16] S. B. Pope, *Turbulent Flows*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[17] C. Robinson, *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*. Boca Raton, FL: CRC, 1999.
[18] S. K. Venkatesan, "Some vortex algorithms for 2D Euler flows," in *Some Applied Problems in Fluid Mechanics*, H. P. Majumdar, Ed. Calcutta, India: Indian Statist. Inst., 1993.
[19] L. A. Zadeh, "Fuzzy sets as a basis for theory of possibility," *Fuzzy Sets Syst.*, vol. 1, pp. 3–28, 1978.
[20] [Online]. Available: http://www.isical.ac.in/~ecsu/kausik.htm
[21] T. Sudkamp, "On probability-possibility transformations," *Fuzzy Sets Syst.*, vol. 51, pp. 73–81, 1992.
[22] A. Kandel, *Fuzzy Techniques in Pattern Recognition*. New York: Wiley, 1980.
[23] K. K. Majumdar, "A mathematical model of the nascent cyclone," *IEEE Trans. Geosci. Remote Sensing*, to be published.

Obstacle Avoidance for Kinematically Redundant Manipulators Using a Dual Neural Network

Yunong Zhang and Jun Wang

***Abstract*—One important issue in the motion planning and control of kinematically redundant manipulators is the obstacle avoidance. In this paper, a recurrent neural network is developed and applied for kinematic control of redundant manipulators with obstacle avoidance capability. An improved problem formulation is proposed in the sense that the collision-avoidance requirement is represented by dynamically-updated inequality constraints. In addition, physical constraints such as joint physical limits are also incorporated directly into the formulation. Based on the improved problem formulation, a dual neural network is developed for the online solution to collision-free inverse kinematics problem. The neural network is simulated for motion control of the PA10 robot arm in the presence of point and window-shaped obstacle.**

***Index Terms*—Dual neural network, obstacle avoidance, quadratic programming, redundant manipulators.**

I. INTRODUCTION

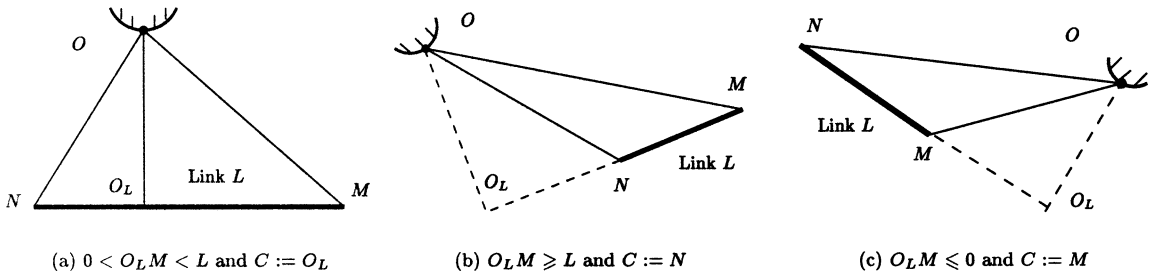
Robot manipulators have been applied in factory automation doing repetitive and dull work, such as carrying radioactive materials and working in hazardous or cluttered environments. It is very important for a robot manipulator to avoid collisions with obstacles; otherwise, the manipulators or the object being held may result in serious damage. Unlike nonredundant manipulators that may not have the ability to avoid obstacles while completing the specified end-effector motion, redundant manipulators with extra degrees of freedom (DOF) [4] may be utilized to improve their dexterity such that they can work effectively while avoiding obstacles [5]–[16].

Many studies have been reported on the obstacle avoidance issue. For example, the pseudoinverse method and its variants [1], [5]–[7] generally yield an minimum-norm particular solution plus a homogeneous solution. A goal of obstacle avoidance can then be specified to

Manuscript received September 25, 2002; revised February 19, 2003. This work was supported by the Hong Kong Research Grants Council under Grant CUHK4165/98E. This paper was recommended by Associate Editor M. S. de Quiroz.

The authors are with the Department of Automation and Computer-Aided Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

Digital Object Identifier 10.1109/TSMCB.2003.811519



(a) $0 < O_LM < L$ and $C := O_L$ (b) $O_LM \geqslant L$ and $C := N$ (c) $O_LM \leqslant 0$ and $C := M$

Fig. 1. Locations of the critical point C corresponding to three possible obstacle positions O relative to manipulator link L.

control the self motion of the manipulator in order not to collide with obstacles while completing the motion task. However, algorithmic singularities are likely to occur and cause infeasible solutions, even though the manipulator Jacobian is singularity-free [1], [9], [14]. The schemes with task priority strategy are proposed to remedy this problem [8], [9], but their convergence rates, extra computational burden, and serial-processing nature might hinder the higher DOF sensor-based applications, especially in the presence of multiple obstacles and many physical constraints such as joint limits.

Another technique for obstacle avoidance of manipulators and mobile robots is by means of artificial potential field [10]–[13]. In this method, obstacles are represented by repulsive surfaces while the target position is represented as an attractive pole so that the manipulator may reach the goal without colliding with obstacles. However, this approach may be more suited for mobile robot path planning than for redundant manipulator motion planning [12], [13], due to the existence of (structural) local minima, the inability to deal with arbitrarily shaped obstacles, and the intensive computation load for the complicated case with three-dimensional (3-D) multiple obstacles and multiple links.

To resolve manipulator redundancy under inequality constraints like joint limits and joint velocity limits, the quadratic programming (QP) methods have been recently established for the inverse kinematic control of redundant manipulators [14]–[20] based on the pseudoinverse methods. In particular, several methods [2], [15] are designed to maximize the distance between the robot links and obstacles. The heavy computational burden for the distance-function derivatives with respect to joint velocity may rule out real-time applications, in addition to the fact that it is unnecessary to always resolve manipulator redundancy for distance maximization when the manipulator is far away from obstacles. In contrast, other approaches [3], [14] treat the collision-free criterion as dynamic equality constraint.

Parallel and distributed computation methods such as recurrent neural networks are effective and efficient alternatives for real-time solutions to inverse kinematics problem [3], [18], [20]–[25]. In the paper, an improved problem formulation is proposed in the sense that the collision avoidance requirement is introduced by using dynamically updated inequality constraints instead of equality constraints, and that physical constraints such as joint limits are also included in the formulation. Then, we present a dual neural network for obstacle avoidance based on the proposed inequality-based QP formulation. Simulation results are discussed to control the 7-DOF PA10 robot arm equipped with a 1-DOF long tool to move along 3-D desired paths in the presence of point obstacles or the window-shaped obstacle.

II. Preliminaries

A. Inverse Kinematics

Consider a redundant manipulator of which the end-effector position/orientation vector $r_E(t) \in R^m$ in Cartesian space is related to the joint-space vector $\theta(t) \in R^n$ through the following forward kinematic equation:

$$r_E(t) = f(\theta(t)) \tag{1}$$

where $f(\cdot)$ is a continuous nonlinear function with a known structure and parameters for any given manipulator. The inverse kinematics problem is to find the joint variable $\theta(t)$ for any given $r_E(t)$ through the inverse mapping of (1). Unfortunately, it is usually impossible to find an analytic solution due to the nonlinearity of $f(\cdot)$. The inverse kinematics problem of manipulators is thus usually solved at the velocity level. Differentiating (1) with respect to time yields a linear relation between the Cartesian velocity $\dot{r}_E$ and the joint velocity $\dot{\theta}$:

$$J_E(\theta)\dot{\theta} = \dot{r}_E \tag{2}$$

where $J_E(\theta) \in R^{m\times n}$ is the end-effector Jacobian matrix defined as $J_E(\theta) = \partial f(\theta)/\partial\theta$. Thus, the following quadratic program is used for the online resolution of (2) [16], [20], [24], [25]:

$$\min \quad \|\dot{\theta}\|_2^2 \tag{3}$$

$$\text{s.t.} \quad J_E(\theta)\dot{\theta} = \dot{r}_E \tag{4}$$

where $\|\cdot\|_2$ denotes the Euclidean norm of a vector. However, like joint physical limits, various obstacles do exist in almost any manipulator workspace. For example, even if there are no external obstacles, collisions may still occur between manipulator links and the pedestal [26]. If a motion encounters a collision and then fails there, the desired end-effector trajectory $r(t)$ becomes impossible to follow, not to mention the physical damage possibly caused.

B. Critical Point Location

The initial step for obstacle avoidance is usually to locate the critical point C on the vulnerable link L of the manipulator by computing the distance $OC(t,\ \theta)$ between the obstacle point O and the link L at any time instant $t \in [t_0,\ t_f]$. As shown in Fig. 1, corresponding to different positions of the obstacle point O, there are three possible cases for locating the critical point C on the vulnerable link L. Based on the geometric property, we have

$$OO_L^2 = OM^2 - O_LM^2 = ON^2 - O_LN^2$$

$NO_L = NM - O_LM$, and the link length $NM := L > 0$, which join together to give $OM^2 - O_LM^2 = ON^2 - (NM - O_LM)^2$, and then

$$O_LM = (OM^2 + NM^2 - ON^2)/(2NM). \tag{5}$$

According to the sign and magnitude of O_LM in (5), there are three cases, as depicted in Fig. 1:

Case a) $0 < O_LM < L$: The critical point C is O_L, and the shortest distance OC is OO_L.

Case b) $O_LM \geqslant L$: The critical point C is N, and the shortest distance OC is ON.

Case c) $O_L M \leqslant 0$: The critical point C is M, and the shortest distance OC is OM.

Note that the above method of dynamically locating critical point C via obstacle point O is the common basic requirements for both model-based and sensor-based manipulator control. In model-based control, Cartesian world-map information regarding the environment is *a priori* available. The obstacle point O is thus determined via the online distance minimization between manipulator link (represented as zero-thickness stick) and obstacle object (usually represented as convex polyhedrons, spheres, or ellipsoids) [1], [2], [6], [12], [21]. In sensor-based control, the obstacle point O is determined by synthetic information of sensor fusion technologies, e.g., utilizing cameras, ultrasonic sensors, and infra-red sensitive skin [1], [6], [27]–[31].

C. Equality-Based QP Formulation

If the obstacle-link distance OC is less than the safety threshold d_1 [which is also called the inner space-of-influence (SOI) [26]], the critical-point Jacobian $J_O \in R^{3\times n}$ has to be calculated. Then, an escape velocity $\dot{r}_O \in R^{3\times 1}$ is desired and assigned at the critical point C, which directs the manipulator link L away from the obstacle O. The next step for obstacle avoidance is to treat the collision-free criterion as the dynamic equality constraint $J_O\dot{\theta} = \dot{r}_O$, and thus, the equivalent QP formulation is [1], [3], [7], [14]

$$\begin{aligned} \min \quad & \|\dot{\theta}\|_2^2 \\ \text{s.t.} \quad & J_E(\theta)\dot{\theta} = \dot{r}_E \\ & J_O(\theta)\dot{\theta} = \dot{r}_O. \end{aligned} \tag{6}$$

However, when adopting the above equality-based obstacle-avoidance QP formulation, it is always difficult (or sometimes impossible) to determine the suitable magnitude of escape velocity $\dot{r}_O$. Let us discuss two simple but heuristic inconsistent cases. In the first case, since the distances between an obstacle point O and two links are both less than the safety threshold d_1, two escape velocities $\dot{r}_{O1}$ and $\dot{r}_{O2}$ are needed. Clearly, the dimension of the joint velocity variables is n, whereas the total dimension of the equality constraints is $m + 3 + 3$. This is an overdetermined situation in mathematics instead of the original underdetermined situation (3) and (4), and thus, the QP solution procedure may fail, when $n < m + 3 + 3$ (for example, the end-effector positioning task of a 7-DOF manipulator $n = 7$, $m = 3$). In the second case, the distances between two obstacles and a link are both less than the safety threshold d_1, but the two escape velocities $\dot{r}_{O1}$ and $\dot{r}_{O2}$ may be of opposite directions. Unless magnitudes of the two escape velocities are both zero, the redundancy-resolution procedure may be in a dilemma and finally stop, due to solution discontinuity, vibration, and integration failures [1], [3], [7], [14]. In addition, it is hard and inefficient to identify online the specific inconsistent situation and then compensate, although there actually exist feasible solutions.

III. Improved Problem Formulation

To avoid unnecessarily reducing the solution space and to improve the formulation robustness, an inequality-based obstacle avoidance QP method is proposed in this section by using only the x, y, and z-axis directions of escape velocity $\dot{r}_O$. In detail, the direction of escape velocity $\dot{r}_O$ is geometrically selected as the vector $\overline{OC}$ pointing from the obstacle point O to the critical point C:

$$\bar{\dot{r}}_O = [\, x_C - x_O \quad y_C - y_O \quad z_C - z_O \,]^T \tag{7}$$

where $[x_C,\, y_C,\, z_C]$ and $[x_O,\, y_O,\, z_O]$ are, respectively, x, y, and z-axis coordinates of points C and O with regard to the base frame.

Referring to (6) and (7), the following dynamic inequality constraint as collision avoidance criterion can be derived:

$$J_N(\theta)\dot{\theta} \leqslant 0 \tag{8}$$

where after extending the scalar signum function $\mathrm{sgn}(\cdot)$ to the vector-valued case, we have

$$J_N(\theta) = -\mathrm{sgn}(\bar{\dot{r}}_O) \diamond J_O(\theta).$$

The vector–matrix multiplication operator $\diamond$ is defined as

$$u \diamond V = \begin{bmatrix} u_1 V_1 \\ u_2 V_2 \\ \cdots \\ u_p V_p \end{bmatrix}$$

where column vector $u := [u_1,\, u_2,\, \ldots,\, u_p]^T$ and the row vector V_i denotes the ith row of matrix V.

Proposition 1: The inequality-based obstacle-avoidance constraint (8) is a variable-magnitude escape velocity method.

Proof: See the Appendix.

Thus, the collision-free redundancy-resolution scheme is formulated as (8): When (8) becomes an equality, it requires the vulnerable link to keep still, and then, the distance OC will not decrease again; when (8) becomes a strict inequality, it requires that the vulnerable link is forced to move away from the obstacle, and then, the distance OC will increase. However, a possible discontinuity in $\dot{\theta}$ may occur when suddenly imposing (8) to the manipulator link at the time instant $t = \{t \in [t_0,\, t_f] | OC(t,\, \theta) = d_1\}$. This discontinuity can be removed by means of the smoothing consideration below:

$$J_N\dot{\theta} \leqslant b \tag{9}$$

where (9) is imposed only when the minimum link-obstacle distance OC is less than the outer safety threshold d_2, $b := s(OC)\max(J_N\dot{\theta}|_{d=d_2},\, 0)$, and the operator $\max(u,\, 0)$ is the vector-valued function for the maximal values between components u_i and 0 [2], [3], [20]. The distance-based smoothing function $s(d)$ is defined as

$$s(d) = \begin{cases} 1, & \text{if } d \geqslant d_2 \\ \sin^2\left(\dfrac{\pi}{2} \cdot \dfrac{d - d_1}{d_2 - d_1}\right), & \text{if } d_1 < d < d_2 \\ 0, & \text{if } d \leqslant d_1. \end{cases}$$

Physical interpretation of the smoothing consideration (9) is that when the robot link L enters the buffer zone $[d_1,\, d_2]$ of an obstacle point, an even and gradual deceleration of $J_N(\theta)\dot{\theta}$ will be imposed by using $s(d)$. Starting from its initial value of $\max(J_N\dot{\theta}|_{d=d_2},\, 0)$, the constraint term $J_N(\theta)\dot{\theta}$ in (9) finally decreases to 0 [i.e., (8)] to repel the link if the minimum link-obstacle distance OC tries to approach the inner safety threshold d_1.

Compared to previous approaches, additional physical constraints can be also included such as joint limits and joint velocity limits:

$$\theta^- \leqslant \theta \leqslant \theta^+, \qquad \dot{\theta}^- \leqslant \dot{\theta} \leqslant \dot{\theta}^+$$

where $\theta^{\pm}$ and $\dot{\theta}^{\pm}$ hereafter denote the upper and lower limits of joints and joint velocities, respectively. The limited joint range $[\theta^-,\, \theta^+]$ has to be converted into the $\dot{\theta}$-based constraint [16], [25], e.g.,

$$\beta(\theta^- - \theta) \leqslant \dot{\theta} \leqslant \beta(\theta^+ - \theta)$$

where $\beta > 0$ is used to scale the feasible region of $\dot{\theta}$. Joint physical limits $\theta^{\pm}$ and $\dot{\theta}^{\pm}$ are thus combined as $\eta^- \leqslant \dot{\theta} \leqslant \eta^+$ with the ith entries of $\eta^{\pm}$ defined as

$$\eta_i^- = \max\{\dot{\theta}_i^-,\, \beta(\theta_i^- - \theta_i)\}, \qquad \eta_i^+ = \min\{\dot{\theta}_i^+,\, \beta(\theta_i^+ - \theta_i)\}.$$

Therefore, the joint-limit-avoidance collision-free inverse kinematic problem is finally improved as the following constrained quadratic program:

$$\min \quad \|\dot{\theta}\|_2^2 \tag{10}$$

$$\text{s.t.} \quad J_E(\theta)\dot{\theta} = \dot{r}_E \tag{11}$$

$$\eta^- \leqslant \dot{\theta} \leqslant \eta^+ \tag{12}$$

$$J_N\dot{\theta} \leqslant b. \tag{13}$$

IV. Dual Neural Network

In this section, a one-layer dual neural network is developed to compute the optimal solution to limited-joint-range collision-free inverse kinematic problem (10)–(13). The dual neural network model [24] has been finally established for online quadratic optimization subject to equality, inequality, and bound constraints. Unlike other recurrent neural networks based on penalty parameters, the dual neural network is able to converge to exact optimal solutions. With the advantages of exponential convergence, small number of neurons, and simple piecewise-linear architecture, the dual network was applied to the bi-criteria inverse kinematic control of redundant manipulators [25].

The dynamic equation and output equation of the dual neural network model for solving (10)–(13) can be derived as

$$\dot{u} = \mu\left\{g(JJ^Tu - u) - JJ^Tu\right\} \tag{14}$$

$$\dot{\theta} = J^Tu \tag{15}$$

where μ is a positive design parameter to scale the convergence rate of the dual network, and the vector-valued piecewise linear function $g(u) = [g_1(u_1), \ldots, g_{m+n+3p}(u_{m+n+3p})]^T$ is defined such that $\forall i \in \{1, \ldots, m+n+3p\}$

$$g_i(u_i) = \begin{cases} \xi_i^-, & \text{if } u_i < \xi_i^- \\ u_i, & \text{if } \xi_i^- \leqslant u_i \leqslant \xi_i^+ \\ \xi_i^+, & \text{if } u_i > \xi_i^+. \end{cases} \tag{16}$$

The augmented vectors $\xi^{\pm}$ and matrix J are defined as

$$\xi^- := \begin{bmatrix} \dot{r}_E \\ \eta^- \\ b^- \end{bmatrix}, \quad \xi^+ := \begin{bmatrix} \dot{r}_E \\ \eta^+ \\ b \end{bmatrix}, \quad J := \begin{bmatrix} J_E \\ I \\ J_N \end{bmatrix}$$

where $b^- \in R^{3p}$, and $\forall j \in \{1, \ldots, 3p\}$, $b_j^- \ll 0$.

The dynamic equation (14) shows that the dual neural network is piecewise linear without any high-order nonlinear term and without using any analog multiplier or penalty parameter. Consequently, its architecture is much simpler, as opposed to other neural network approaches [18], [20], [21]–[23]. See Fig. 2. In the kinematic control process, the desired velocity vector $\dot{r}_E(t)$ is input into the dual network, and at the same time, the dual network outputs the signal $\dot{\theta}(t)$ as the collision-free optimal solution.

As shown in [24] and [25], starting from any initial state, the output $\dot{\theta}(t)$ of the dual neural network (14) is convergent to optimal solution of the constrained collision-free inverse kinematics (10)–(13). Moreover, the dual network has exponential convergence property if there exists $\lambda > 0$ such that $\|g(JJ^Tu - u) - JJ^Tu\| \geqslant \lambda\|u - u^*\|$, where u^* is the optimality point of state u.

The above convergence property is based on the existence of the nonempty feasible region defined by (11)–(13). In the event of empty feasible region, the neural network will give a least-squares solution to (10)–(13). In addition, the final position tracking error is upper bounded by $\varepsilon(t_f - t_0)$, where t_0 and t_f are, respectively, the starting and final time of an end-effector task, and the coefficient ε can be made arbitrarily small by increasing the design parameter μ [25].

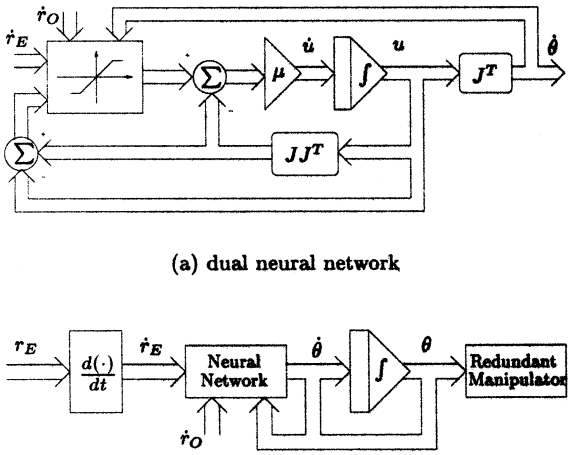


Fig. 2. Block diagram of the neural online solver for the obstacle-avoidance kinematic control scheme (10)–(13).

V. Simulation Results

Consider the 7-DOF Mitsubishi PA10 robot arm equipped with a 1-DOF long tool of which the structure parameters and joint limits are shown in [18], except for $d_7 = 0.28$ m, $d_8 = 0.20$ m, $\theta_8^{\pm} = 2\pi$ rad, $\dot{\theta}_8^{\pm} = 2\pi$ rad/s, and

$$T_{78} = \begin{bmatrix} \cos(\theta_8) & 0 & \sin(\theta_8) & d_8\sin(\theta_8) \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_8) & 0 & \cos(\theta_8) & d_8\cos(\theta_8) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

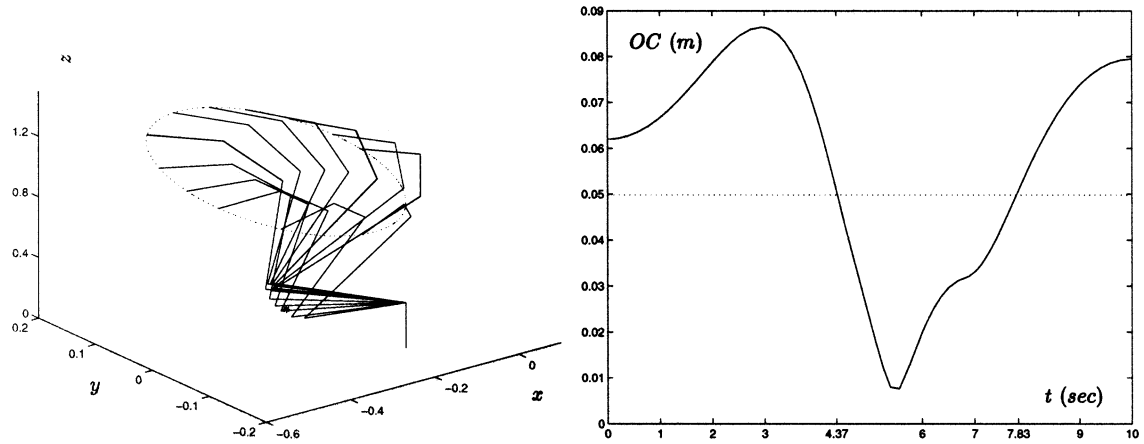
In this study, only the positioning of the long tool end-point is concerned, and thus, $m = 3$ and $n = 8$.

A. Point Obstacle Avoidance

In this example, the desired end-point motion of the long-tool PA10 robot arm is a 3-D circular path of radius 0.2 m, where the slope angle made with xy-plane is $\pi/6$ rad. The manipulator initial joint vector is $\theta(0) = [0, -\pi/4, 0, \pi/2, 0, -\pi/4, 0, 0]^T$ rad, and the dual network design parameter $\mu = 10^4$. The obstacle point O is at $[-0.35, -0.05, 0.3]$ m, and the inner/outer safety thresholds are, respectively, $d_1 = 0.05$ and $d_2 = 0.10$ m.

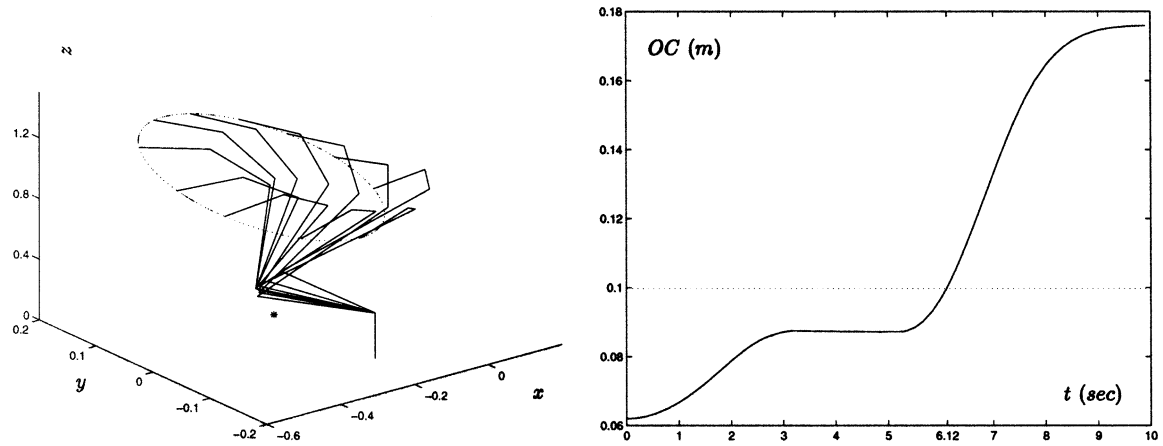
Fig. 3(a) shows the motion trajectory of the PA10 robot arm without considering the point obstacle O (denoted by $*$) in its work envelope. Although the maximum Cartesian positioning error is very small ($<$0.15 mm) and joint variables are kept within their mechanical limits, the solution fails due to the collision with the obstacle during the time period $[4.37, 7.83]$ s, which is shown in Fig. 3(b). For comparison, Fig. 4 shows the results of the dual neural network under the proposed obstacle-avoidance QP formulation (10)–(13). Specifically, Fig. 4(a) and (b) shows that the manipulator motion trajectory has been kept away from the obstacle while the end-effector tracking the desired circular path. Compared with Fig. 3(b), the new minimum link-obstacle distance in Fig. 4(b) is always greater than the inner safety threshold d_1, and during the buffer zone $[d_1, d_2]$, the distance never decreases again. Simulation data also show that the joint variables have been always kept within their mechanical limits, and the maximum Cartesian positioning error is less than 0.12 mm.

As presented in Fig. 5, the simulation results based on two point-obstacles also substantiate the effectiveness of the proposed obstacle-avoidance QP problem formulation and the efficiency of the dual neural network online solver for kinematic control of redundant manipulators.



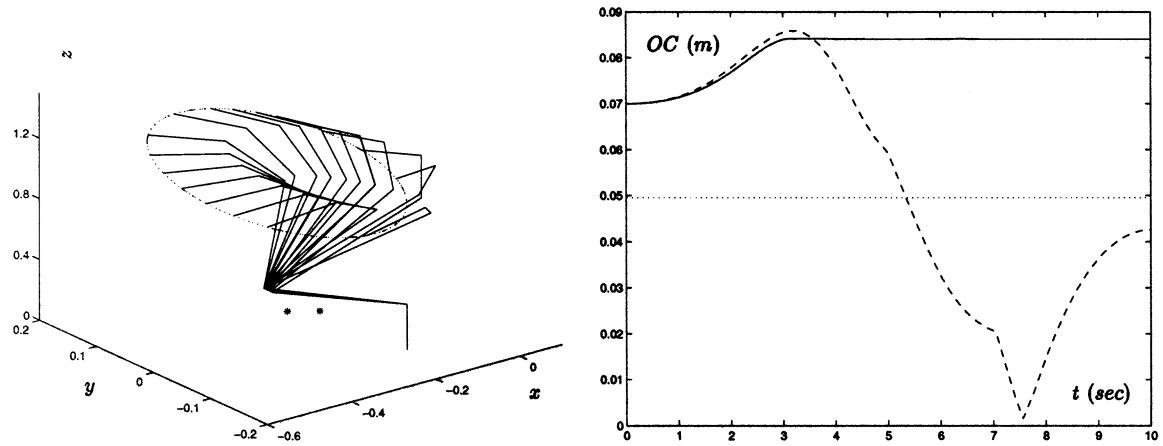
(a) manipulator motion trajectory (b) minimum link-obstacle distance

Fig. 3. Simulated motion of the PA10 robot arm without considering collision with point obstacle $*$.



(a) manipulator motion trajectory (b) minimum link-obstacle distance

Fig. 4. Simulated motion of the PA10 robot arm synthesized by the dual neural network under the limited-joint-range collision-free QP formulation (10)–(13).



(a) manipulator motion trajectory (b) minimum link-obstacle distance

Fig. 5. Simulated motion of the manipulator in the presence of two point obstacles where the solid and dashed lines in subplot (b) correspond to the kinematic control with and without consideration of the second obstacle, respectively.

B. Window-Shaped Obstacle Avoidance

Another illustrative example of sensor- or model-based kinematic control is to avoid window or truss-structured obstacles. In sensor-based control, the minimum distance between window edges and manipulator links is determined from sensor information to know locations of obstacle point O and critical point C, which can be simulated by using optimization routines such as MATLAB "CONSTR" with small calculation error [6], [30], [31]. For model-based kinematic control, a parametric geometry-calculation scheme has been well developed in [2] and [32] to determine the obstacle point O, critical point C, and the minimum link-obstacle distance OC.

In this subsection, the model-based method is preferably studied to complement the efficient parametric calculation scheme of [2]. Simply saying, the window edge vector $\overline{QP}$ and the manipulator link vector $\overline{NM}$ are, respectively, defined as

$$\overline{QP} = \begin{bmatrix} QP_x \\ QP_y \\ QP_z \end{bmatrix} = \begin{bmatrix} P_x - Q_x \\ P_y - Q_y \\ P_z - Q_z \end{bmatrix}$$

$$\overline{NM} = \begin{bmatrix} NM_x \\ NM_y \\ NM_z \end{bmatrix} = \begin{bmatrix} M_x - N_x \\ M_y - N_y \\ M_z - N_z \end{bmatrix}.$$

Therefore, the parametric equations are, respectively

$$\text{Edge: } \begin{cases} x = Q_x + QP_x\omega \\ y = Q_y + QP_y\omega \\ z = Q_z + QP_z\omega \end{cases}, \quad 0 \leqslant \omega \leqslant 1$$

$$\text{Link: } \begin{cases} x = N_x + NM_x\kappa \\ y = N_y + NM_y\kappa \\ z = N_z + NM_z\kappa \end{cases}, \quad 0 \leqslant \kappa \leqslant 1.$$

The common normal with edge and link is thus defined as

$$\overline{EL} = \begin{bmatrix} EL_x \\ EL_y \\ EL_z \end{bmatrix} = \begin{bmatrix} QN_x + NM_x\kappa - QP_x\omega \\ QN_y + NM_y\kappa - QP_y\omega \\ QN_z + NM_z\kappa - QP_z\omega \end{bmatrix}.$$

Since $\overline{EL}$ is perpendicular to $\overline{NM}$ and $\overline{QP}$, the inner products are zeros, i.e., [2, eq. (18)]:

$$\begin{aligned} NM_x \cdot EL_x + NM_y \cdot EL_y + NM_z \cdot EL_z = 0 \\ QP_x \cdot EL_x + QP_y \cdot EL_y + QP_z \cdot EL_z = 0. \end{aligned} \tag{17}$$

For the two-variable two-equation system (17), κ and ω can be easily solved. Then, totally, three cases and eight subcases are classified in [2] for determining exactly obstacle point O, critical point C, and minimum distance OC.

However, in the literature, there is still one case neglected—the unusual situation of link and edge being parallel. This parallel case might be encountered in the multilink multiedge obstacle-avoidance application/simulation in the form of underdetermination of κ and ω from (17). Specifically, the two equations in (17) are linearly dependent, and there appears a singular 2×2 coefficient matrix. To efficiently handle the parallel case of (17), a twice normal-finding algorithm has been developed and verified.

Proposition 2: In a parallel situation, parameters κ and ω cannot be solved directly from (17) but can be determined as

$$\kappa = g_0^1\left(\frac{b_1}{A_{11}}\right), \quad \omega = g_0^1\left(\frac{b_1 - A_{11}g_0^1(b_1/A_{11})}{A_{12}}\right) \tag{18}$$

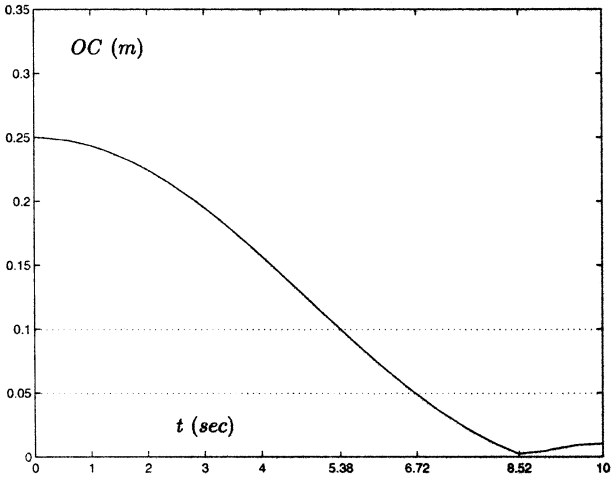


Fig. 6. Minimum link-obstacle distance without consideration of avoiding the window-shaped obstacle.

where $g_0^1(\cdot)$ is a scalar piecewise linear function with lower and upper bounds, respectively, as 0 and 1, of which the definition is the same as the vector-valued function (16).

Proof: See the Appendix.

In the remaining of this subsection, the following numerical example of window-shaped obstacle avoidance is discussed. The desired Cartesian trajectory of the manipulator end-effector is to move along a straight line through a window with the motion distance 0.4 m. The initial joint vector is still $\theta(0) = [0, -\pi/4, 0, \pi/2, 0, -\pi/4, 0, 0]$ rad, and correspondingly, the initial end-effector position is $[-0.106\,07, 0, 1.093\,17]$ m. To show the significance and necessity of the kinematic control with the obstacle-avoidance feature, four vertices of the window are deliberately fixed as

$$\begin{bmatrix} 0.20 \\ 0.25 \\ 1.40 \end{bmatrix} \text{m}, \quad \begin{bmatrix} 0.20 \\ 0.25 \\ 0.80 \end{bmatrix} \text{m}, \quad \begin{bmatrix} -0.4 \\ 0.25 \\ 0.80 \end{bmatrix} \text{m}, \quad \begin{bmatrix} -0.4 \\ 0.25 \\ 1.40 \end{bmatrix} \text{m}.$$

Without considering the window-edge obstacles in its work envelope, the PA10 robot arm is simulated. From Fig. 6 and simulation data, we can see that the third manipulator link enters into the critical zone of the window lower edge (i.e., $OC \leqslant d_1 = 0.05$ m) from the time instant $t = 6.72$ s, then collides with the window edge at $t = 8.52$ s, and finally cuts through the window lower edge during the remaining time period $[8.52, 10]$ s. In order to remedy the collision problem while performing the end-effector path-following task (e.g., robotic welding inside a window), a collision-free solution has to be applied.

For comparison, Fig. 7 shows the results of the dual neural network (14) under the proposed obstacle-avoidance QP formulation (10)–(13) for this window-edge avoidance task. Specifically, the shortest distance between manipulator links and window edges illustrated in Fig. 7 is always greater than the inner safety threshold $d_1 = 0.05$ m, and thus, the new solution is collision-free. When the shortest distance OC enters into its buffer zone $[0.05, 0.10]$ m at $t = 5.3$ s, a deceleration appears by imposing (9) to drive the vulnerable manipulator link(s) away from window edges. Thus, the distance OC increases quickly and then becomes larger than the outer safety threshold $d_2 = 0.1$ m from the time instant $t = 6.1$ s. This implies no link collision with window edges and also confirms the explanation for Fig. 7. In addition, as shown in simulation data, the joint variables have been always kept within their mechanical limits, and the maximal positioning error is less than 0.1 mm. This demonstrates the effectiveness and generality of the proposed inequality-based collision-free QP problem formulation (10)–(13) and dual neural network (14).

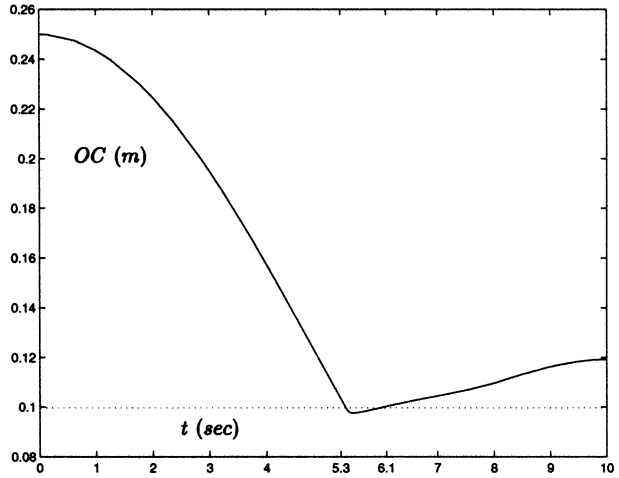


Fig. 7. Minimum link-obstacle distance with considering the window-edge avoidance.

VI. Concluding Remarks

In the paper, a neural network approach is presented for collision-free motion control of redundant manipulators. An improved problem formulation has been proposed that represents the collision-free requirement as dynamic inequality constraint and incorporates joint physical limits into an optimization problem. For computing online solution of the obstacle-avoidance kinematics problem, a dual neural network is developed. Simulation results based on the PA10 robot arm demonstrate the effectiveness of the new problem formulation and its dual neural network on both sensor-based and model-based obstacle avoidance of redundant manipulators.

Appendix

Proof of Proposition 1: Consider an escape velocity $\dot{r}_O$ of variable magnitude γ applied to the critical point C to drive the link L away from the obstacle O. In mathematics

$$J_O(\theta)\dot{\theta} = \gamma\bar{\dot{r}}_O = \gamma \begin{bmatrix} x_C - x_O \\ y_C - y_O \\ z_C - z_O \end{bmatrix} \tag{19}$$

where $J_O(\theta)$ is the critical-point Jacobian matrix without orientation information, $\gamma \geqslant 0$ is a variable magnitude of escape velocity $\dot{r}_O$, and $[x_C,\ y_C,\ z_C]$ and $[x_O,\ y_O,\ z_O]$ are, respectively, the critical-point and obstacle-point coordinates with respect to the base frame. Using non-negative variables γ_x, γ_y, and γ_z to represent magnitude components of escape velocity $\dot{r}_O$, respectively, in the x, y, and z-axis directions, (19) can be further generalized as

$$J_O(\theta)\dot{\theta} = \begin{bmatrix} \gamma_x(x_C - x_O) \\ \gamma_y(y_C - y_O) \\ \gamma_z(z_C - z_O) \end{bmatrix}.$$

By defining J_{Ox}, J_{Oy}, and J_{Oz} as row vectors of $J_O(\theta)$, the above matrix equality constraint is equivalent to

$$J_{Ox}\dot{\theta} = \gamma_x(x_C - x_O), \qquad \gamma_x \geqslant 0 \tag{20}$$

$$J_{Oy}\dot{\theta} = \gamma_y(y_C - y_O), \qquad \gamma_y \geqslant 0 \tag{21}$$

$$J_{Oz}\dot{\theta} = \gamma_z(z_C - z_O), \qquad \gamma_z \geqslant 0. \tag{22}$$

Since γ_x, γ_y, and γ_z are non-negative variables, we can reformulate (20)–(22) as the following ones by using the bi-polar signum function

$$J_{Ox}\dot{\theta} = \gamma_x \text{sgn}(x_C - x_O), \qquad \gamma_x \geqslant 0$$

$$J_{Oy}\dot{\theta} = \gamma_y \text{sgn}(y_C - y_O), \qquad \gamma_y \geqslant 0$$

$$J_{Oz}\dot{\theta} = \gamma_z \text{sgn}(z_C - z_O), \qquad \gamma_z \geqslant 0.$$

By simple algebraic operations, it follows that

$$-\text{sgn}(x_C - x_O)J_{Ox}\dot{\theta} = -\gamma_x \leqslant 0$$

$$-\text{sgn}(y_C - y_O)J_{Oy}\dot{\theta} = -\gamma_y \leqslant 0$$

$$-\text{sgn}(z_C - z_O)J_{Oz}\dot{\theta} = -\gamma_z \leqslant 0.$$

In matrix form, the above inequalities is rewritten exactly as (8) with $\bar{\dot{r}}_O$ defined in (7) and $J_N(\theta) := -\text{sgn}(\bar{\dot{r}}_O) \diamond J_O(\theta)$. Thus, in other words, from the above derivation, we know that the inequality-based obstacle-avoidance constraint (8) is actually a variable-magnitude escape velocity method. □

Proof of Proposition 2: The two-variable two-equation system (17) can be rewritten in the following standard form:

$$\begin{aligned} A_{11}\kappa + A_{12}\omega &= b_1 \\ A_{21}\kappa + A_{22}\omega &= b_2 \end{aligned} \tag{23}$$

where

$$\begin{aligned} A_{11} &= NM_x^2 + NM_y^2 + NM_z^2 := NM^2 \\ A_{12} &= -(NM_xQP_x + NM_yQP_y + NM_zQP_z) \\ A_{21} &= -A_{12} \\ A_{22} &= -(QP_x^2 + QP_y^2 + QP_z^2) := QP^2 \\ b_1 &= -(NM_xQN_x + NM_yQN_y + NM_zQN_z) \\ b_2 &= -(QP_xQN_x + QP_yQN_y + QP_zQN_z). \end{aligned}$$

In a nonparallel situation, before following the classification routine [2], the parameters κ and ω can be easily solved by

$$\begin{bmatrix} \kappa \\ \omega \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

However, in a parallel situation, κ and ω cannot be solved by the above nonparallel method since the coefficient matrix A is singular in view of the following properties:

$$\begin{aligned} \frac{A_{11}}{A_{21}} &= \frac{A_{12}}{A_{22}} = \frac{b_1}{b_2} \\ \frac{NM_x}{QP_x} &= \frac{NM_y}{QP_y} = \frac{NM_z}{QP_z}. \end{aligned} \tag{24}$$

However, because A_{11} and A_{22} are both nonzero, by property (24), A_{12} is nonzero. Therefore, a twice normal-finding algorithm based on (23) can be used to solve for a suitable pair of κ and ω.

Step 1) Assume $\omega = 0$. Find $\kappa = b_1/A_{11}$ by locating the common normal via (23). Since κ is required to be within $[0,\ 1]$, we have $\kappa = g_0^1(b_1/A_{11})$ in (18).

Step 2) Based on the above κ value, another common normal is located via (23) to solve

$$\omega = (b_1 - A_{11}\kappa)/A_{12} = (b_1 - A_{11}g_0^1(b_1/A_{11}))/A_{12}.$$

Then, by forcing ω within the range $[0,\ 1]$, we have $\omega = g_0^1((b_1 - A_{11}g_0^1(b_1/A_{11}))/A_{12})$ in (18).

The twice normal-finding method is actually a geometric method with validity shown in Fig. 8. It follows from subplots of Fig. 8 or their variants that a suitable pair of κ and ω can be always found by the proposed twice normal-finding method, i.e., (18). □

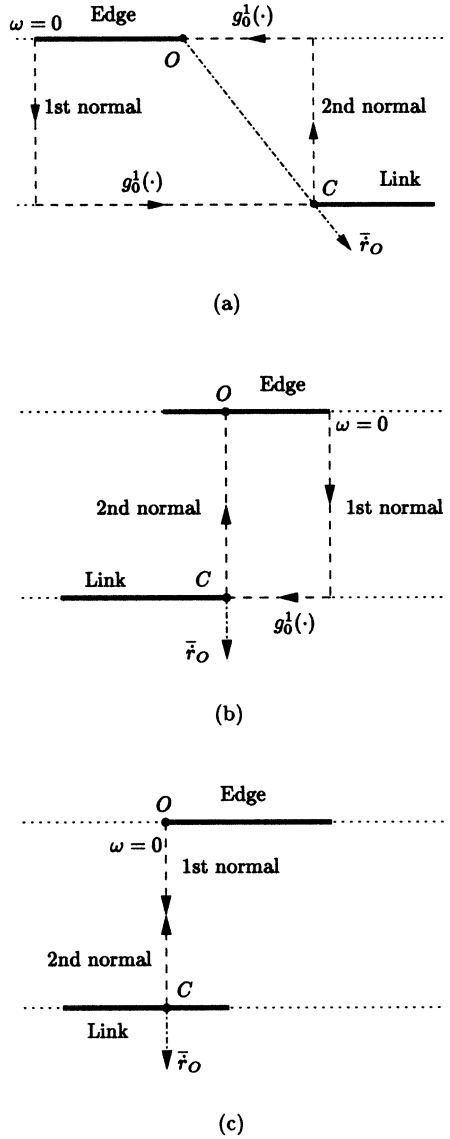


Fig. 8. Three possible subcases used to show the validity of the twice normal-finding method (18).

REFERENCES

[1] A. A. Maciekewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, pp. 109–117, 1985.

[2] F.-T. Cheng, Y.-T. Lu, and Y.-Y. Sun, "Window-shaped obstacle avoidance for a redundant manipulator," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 806–815, Dec. 1998.

[3] W. Tang, L. Lam, and J. Wang, "Kinematic control and obstacle avoidance for redundant manipulators using a recurrent neural network," in *Proc. Int. Conf. Artificial Neural Networks*, 2001, pp. 922–929.

[4] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. London, U.K.: Springer-Verlag, 2000.

[5] ——, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE J. Robotics Automat.*, vol. 4, pp. 403–410, Aug. 1988.

[6] K. Glass, R. Colbaugh, D. Lim, and H. Seraji, "Real-time collision avoidance for redundant manipulators," *IEEE Trans. Robotics Automat.*, vol. 11, pp. 448–457, June 1995.

[7] H. Ding and S. P. Chan, "A real-time planning algorithm for obstacle avoidance of redundant robots," *J. Intel. Robot. Syst.*, vol. 16, pp. 229–243, 1996.

[8] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. J. Robot. Res.*, vol. 10, no. 4, pp. 410–425, 1991.

[9] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robotics Automat.*, vol. 13, pp. 398–410, June 1997.

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, pp. 90–99, 1986.

[11] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: Theory and experiments," *IEEE Trans. Syst, Man, Cybern.*, vol. 20, pp. 1423–1436, Dec. 1990.

[12] J.-O. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Trans. Robotics Automat.*, vol. 8, pp. 338–349, June 1992.

[13] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robotics Automat.*, vol. 16, pp. 615–620, Oct. 2000.

[14] F.-T. Cheng, T.-H. Chen, Y.-S. Wang, and Y.-Y. Sun, "Obstacle avoidance for redundant manipulators using the compact QP method," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, 1993, pp. 262–269.

[15] J. Guo and T. C. Hsia, "Joint trajectory generation for redundant robots in an environment with obstacles," *J. Robot. Syst.*, vol. 10, pp. 199–215, 1993.

[16] F.-T. Cheng, T.-H. Chen, and Y.-Y. Sun, "Resolving manipulator redundancy under inequality constraints," *IEEE Trans. Robotics Automat.*, vol. 10, pp. 65–71, Feb. 1994.

[17] A. S. Deo and I. D. Walker, "Minimum effort inverse kinematics for redundant manipulators," *IEEE Trans. Robotics Automat.*, vol. 13, pp. 767–775, Oct. 1997.

[18] J. Wang, Q. Hu, and D. Jiang, "A Lagrangian network for kinematic control of redundant manipulators," *IEEE Trans. Neural Networks*, vol. 10, pp. 1123–1132, Sept. 1999.

[19] I. A. Gravagne and I. D. Walker, "On the structure of minimum effort solutions with application to kinematic redundancy resolution," *IEEE Trans. Robotics Automat.*, vol. 16, pp. 855–863, Dec. 2000.

[20] W. Tang and J. Wang, "A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architecture complexity," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, pp. 98–105, Dec. 2001.

[21] Z. Mao and T. C. Hsia, "Obstacle avoidance inverse kinematics solution of redundant robots by neural networks," *Robotica*, vol. 15, pp. 3–10, 1997.

[22] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. London, U.K.: World Scientific, 1998.

[23] H. Ding and J. Wang, "Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 269–276, May 1999.

[24] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, pp. 147–154, Feb. 2001.

[25] Y. Zhang, J. Wang, and Y. Xu, "A dual neural network for bi-criteria kinematic control of redundant manipulators," *IEEE Trans. Robotics Automat.*, vol. 18, pp. 923–931, Dec. 2002.

[26] H. Seraji, M. K. Long, and T. S. Lee, "Motion control of 7-DOF arms: The configuration control approach," *IEEE Trans. Robotics Automat.*, vol. 9, pp. 125–139, Apr. 1993.

[27] D. Nair and J. K. Aggarwal, "Moving obstacle detection from a navigating robot," *IEEE Trans. Robotics Automat.*, vol. 14, pp. 404–416, June 1998.

[28] I. Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE Trans. Robotics Automat.*, vol. 14, pp. 969–978, Dec. 1998.

[29] S. Shoval and J. Borenstein, "Using coded signals to benefit from ultrasonic sensor crosstalk in mobile robot obstacle avoidance," in *Proc. IEEE Int. Conf. Robotics Automat.*, vol. 3, 2001, pp. 2879–2884.

[30] E. Cheung and V. Lumelsky, "Development of sensitive skin for a 3D robot arm operating in an uncertain environment," in *Proc. IEEE Int. Conf. Robotics Automat.*, vol. 2, 1989, pp. 1056–1061.

[31] C. L. Boddy and J. D. Taylor, "Whole-arm reactive collision avoidance control of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robotics Automat.*, vol. 3, 1993, pp. 382–387.

[32] H. Zghal, R. V. Dubey, and J. A. Euler, "Collision avoidance of a multiple degree of redundancy manipulator operating through a window," *J. Dyn. Syst., Meas., Contr.*, vol. 114, pp. 7171–7211, 1992.