Proceedings of the 30th Conference
on Decision and Control
Brighton, England • December 1991

COLLISION AVOIDANCE IN A MULTIPLE-ROBOT SYSTEM USING INTELLIGENT CONTROL AND NEURAL NETWORKS [1]

Kang G. Shin and Xianzhong Cui

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

ABSTRACT

A new hierarchical scheme is proposed to coordinate multiple robots in a common workspace for collision avoidance by combining the techniques of intelligent control and neural networks (NNs). The high level in the hierarchy is formed by a knowledge-based coordinator (KBC) and an NN-based predictor, and the low level consists of the robots to be coordinated. The basic idea is to preview the effects of each command on the robots using the predictor, and then guide the KBC to modify the command for collision avoidance.

Under the assumption that both path planning and trajectory planning do not deal with collision avoidance, adding the KBC does not impose any constraints on the design of the robots' servo controllers. In other words, the robots' dynamic properties are figured in the coordination without affecting the internal structure and parameters of each robot's control system. It is therefore possible to coordinate multiple robots — equipped with commercially-designed servo controllers — working in a common workspace without collision.

The knowledge acquisition and representation of collision detection and avoidance for both cylindrical- and revolute-type robots are discussed in detail. Design of the NN-based predictor and the KBC are summarized. Finally, the proposed scheme is tested extensively via simulations for both types of robots, showing promising performance.

1 INTRODUCTION

Effective application of industrial robots to increase productivity and improve product quality calls for the development of an intelligent control system for them. Particularly, multiple robots need to be coordinated in order to perform such sophisticated manufacturing tasks as assembly. In a multiple-robot system, not only must each robot have good behavior, but also must multiple robots be coordinated to achieve the desired performance. One of the challenging problems in developing intelligent robot control systems is how to coordinate multiple robots in a common workspace without colliding with each other.

A robot control system usually consists of four-level hierarchy: task planning, path planning, trajectory planning, and servo control. The problem of collision avoidance among robots[2] can be solved at the path planning level by considering collision between the robot and the fixed/static obstacles in the workspace. By "path planning", we mean off-line geometric planning in robots' workspace. Generally, there are two approaches to path planning: graph search and use of potential field [3]. In the former, a collision-free path is obtained by searching a graph which is derived from geometric constraints. The latter assumes an artificial potential field applied to obstacles and the goal position. A collision-free path is then planned along the curve of minimum potential.

Collision avoidance can also be achieved by finding collision-free trajectories with optimization or search methods. A simple solution is to keep all other robots away from the workspace if it is occupied by a robot. Obviously, this scheme lacks the flexibility of allowing more than one robot to jointly accomplish a complex task such as assembly, and requires a longer time to complete a task, since the usage of the workspace is strictly sequential. Shin and Zheng developed a simple scheme for planning trajectories of two robots working in the same workspace by minimizing the robots' operation time while avoiding collision between them [4]. Their basic idea is to delay one robot by a minimum amount of time to avoid collision with the other robot. However, an exact collision time between the two robots that is required for this scheme is difficult to obtain, since a precise trajectory for a given robot is not always available.

In practice, the desired path and trajectory of each robot are determined by guiding the robot through the workspace with a joystick, and its servo controller is designed independently of, and separately from, the other robots. To coordinate such robots in a common workspace, one has to devise a scheme for on-line collision detection and avoidance.

Regardless of collision-avoidance scheme used, it is essential to track a robot's desired trajectory precisely, which in turn calls for high-performance servo controllers. Otherwise, collision may occur even if the desired trajectory is planned to be collision-free. This implies that the dynamics of multiple robots must be figured in their coordination. An on-line coordinator is thus needed to guide the robots using sensory information. This on-line coordination is commonly termed the *path finding* problem. Since path finding does not always guarantee the robots to achieve the goal, a high-level planner is still necessary. However, the existence of on-line coordination will ease the burden on both path planning and trajectory planning for collision detection and avoidance. The path finding problem for a multiple-robot system is the main subject of this paper.

There are only a few papers dealing with on-line coordination of multiple robots for collision avoidance. The work described in [2] is a typical example. The potential problems of this scheme are its computational complexity, and its restriction to the robots with cylindrical joints. Potential-field methods are also used to deal with the path finding problem [7, 8]. However, it is difficult to use these methods for moving obstacles or on-line coordination of multiple robots. Thus, we want to address the problem of on-line coordination of multiple robots which are equipped with commercial-designed, simple controllers.

Most industrial robots are designed to work as a stand-alone device and are usually equipped with PID-type servo controllers. Thus, it is reasonable to assume that:

A1. The desired path of a robot is obtained by teaching, and thus, avoids collision only with fixed obstacles in the workspace.

A2. Trajectory planning does not deal with the problem of avoiding collision between robots.

A3. Collision avoidance is not a subject to consider when designing servo controllers. Servo controllers are commercially-designed and independent of one another.

---

[1] The work reported in this paper was supported in part by the National Science Foundation under Grant No. DMC-8721492. Any opinions, findings, and recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the NSF.

[2] The term "robot" will henceforth mean "robotic manipulator" and the two terms will be used interchangeably, unless stated otherwise.

A4. No detailed knowledge on the dynamic structure and/or parameters of each robot and its servo controller is available.

It is in general very difficult to coordinate multiple robots under the above (realistic) assumptions. This coordination problem can be viewed as having a hierarchical structure, in which the internal structure and parameters of low-level subsystems — individual robots — must not be affected by adding a high-level coordinator. None of the schemes mentioned earlier satisfies this requirement, since they are not intended for on-line coordination of multiple robots which are equipped with commercially-designed servo controllers. Thus, a new method must be developed to achieve the on-line coordination of multiple robots.

Using a hierarchical structure, we develop a practical, yet general, design method for coordinating multiple robots in a common workspace by combining the techniques of intelligent control and neural networks (NNs). The high level consists of a knowledge-based coordinator (KBC) and an NN-based predictor. The coordinated robots form the low level. It is assumed that each robot is equipped with a commercially-designed servo controller. The detailed structure and/or parameters of low-level subsystems are unknown to the KBC, thus allowing the individual subsystems to be designed separately in isolation. This implies that some commercially-designed robots can be coordinated to work in a common workspace without collision.

Section 2 states the problem of coordinating multiple robots for collision avoidance and the basic principles of the KBC. Section 3 discusses knowledge acquisition and representation for the coordination problem through an example of two cylindrical robots, including collision detection and avoidance. Section 4 briefly describes the design of NN-based predictor and the KBC, both of which are detailed in our early papers [5, 6]. For two cylindrical robots, the proposed scheme has been tested via simulations, and the results are summarized in Section 5. The scheme is also tested to coordinate two revolute robots, and the details are presented in Section 6. The redundancy problem can be easily handled by developing a set of rules for collision detection and avoidance. The paper concludes with Section 7.

2 PROBLEM STATEMENT AND BASIC IDEAS OF SOLUTION

The simplest configuration of the multiple robots in a common workspace is shown in Fig. 1 which is similar to that of [2]. The relationship between the base coordinates of the two robots are expressed by the distance d_{12} and the angles ϕ_{01} and ϕ_{02}. For simplicity, only collision avoidance in a two-dimensional workspace is considered, implying that there are no constraints on the vertical movement of the robots. Thus, for the collision avoidance problem, one can assume that each robot has three degrees of freedom (DOFs): one translational and one rotational joint in $X-Y$ plane of the world coordinate, and one translational joint in the vertical (Z) direction. The right-of-way is given to robot 1, that is, robot 1 is the master and will follow its desired trajectory, while robot 2 must be coordinated to avoid collision. The coordinator should be able to easily switch the role of the master and slave robots, when the trajectory of the slave robot cannot be modified for collision avoidance due to some constraints. The coordinator must not only generate a sequence of commands for collision avoidance, but also monitor each robot's dynamic response to the commands.

Another example is to coordinate two revolute robots in a two-dimensional workspace, as shown in Fig. 2. In this workspace, if $\Omega_0 \subset \Omega$ is the space in which the angle of link 2 is 0 or π for each robot, then, for each point $(x, y) \in \Omega \backslash \Omega_0$, there are two different link configurations allowing the end-effector to reach the point. In other words, a 2-link robot is redundant in this workspace. Robot 1 is designated as the master and will follow its desired trajectory, while robot 2 has to be coordinated for collision avoidance. Under assumptions *A1* — *A4*, this redundant case is more difficult to coordinate than that of cylindrical robots.

In a hierarchical structure, the only action to take at high level is to issue a sequence of appropriate commands to low-level subsystems, so that the internal structure and parameters of one level will not be affected by adding another level. These commands are defined as the reference inputs to the low-level subsystems. For multiple robots in a common workspace, the reference inputs to the low level are the desired trajectory of each robot without considering the presence of other robots. The purpose of the coordinator is to modify these desired trajectories to avoid collision among the robots.

Though robot 1 is supposed to follow its desired trajectory, its actual, precise position[3] and angle at each instant are not known due to tracking errors and system disturbances and/or noises. The key problem in accomplishing on-line collision avoidance is to predict d steps ahead the positions and angles of the robots. Let $Y_{id}(k) \equiv [P_{id}(k),\ \phi_{id}(k)]^T$ be the reference input of robot i, $i = 1, 2$, where $P_{id}(k)$ and $\phi_{id}(k)$ are the robot's desired position and angle at time k, respectively. Suppose the robots' predicted positions corresponding to each choice of $Y_{id}(k)$ are available, and other constraints are represented by a set of production rules, such as "IF the trajectory of the slave robot cannot be changed due to the limits of joint torque **THEN** switch the designation of master and slave robots." In each sampling interval, collision avoidance is accomplished by iteratively trying different reference inputs and adjusting them according to the predicted tracking error.

Our basic idea is to estimate the effects of the reference inputs using a predictor and modify them using a KBC so as to avoid collision. The conceptual structure of this scheme is shown in Fig. 3. Using the predictor, one can foresee the effects of each reference input on the robots' future positions. Given the predicted positions and angles of the robots, the simplified knowledge on how to coordinate the robots for collision avoidance can be stated as follows:

- Modify the reference input and feed it to the predictor.
- **IF** the predicted positions and angles do not lead to collision, **THEN** feed the reference input to the robots, **ELSE** re-modify it.

By using the robots' predicted positions and angles, the knowledge of how to coordinate multiple robots to avoid collision becomes clear, thereby simplifying the design of the KBC's knowledge base. To complete the design of the coordinator, the following three problems will be addressed in the following sections. (1) How is the knowledge of collision detection and avoidance acquired and represented for a specified configuration of robots? (2) How is a multiple-step predictor designed? (3) Given predicted robots' positions and angles, how is the KBC designed in order to modify the reference inputs of the robots?

Clearly, the predictor plays an important role in our scheme. Since the detailed structure and/or parameters of the robots to be coordinated are unknown, the multiple-step predictor will be designed by using NNs. Note that the NN's potential for control applications lies in the following properties: (1) they can approximate any continuous mapping, (2) they achieve this approximation by learning, and (3) it is easy to accomplish parallel processing and fault tolerance. However, NNs *alone* cannot form an intelligent control/coordination system. NNs lack the ability of logical reasoning and decision making, interpretation of environmental changes, and quick response to unexpected situations. Consequently, one must have a KBC. Despite its drawbacks, the NN-based predictor establishes an explicit relationship between the future response of the robots and the reference inputs to them, simplifying the KBC's knowledge base that guides the modification process of the reference inputs. One can also add easily to the knowledge base such rules as the mechanical constraints, operation monitoring, system protection, and dynamically designating the master and the slave robots or switching between the different coordination schemes. The KBC emphasizes system coordination, while the ability of learning relies mainly on the NN. That is, the NN will

---

[3] The term "position" will henceforth be used to denote the distance from the robot's end-effector to the origin of its base coordinates.

adapt itself to the model/parameter uncertainties, disturbance, and component failures.

3 KNOWLEDGE ACQUISITION AND REPRESENTATION FOR COLLISION AVOIDANCE

The problem of path finding is divided into two parts: (1) *collision detection*, and (2) *collision avoidance*. The former is to find an algorithm and a set of rules for collision detection, and the later is to design a maneuvering strategy for collision avoidance. Both of these problems will be addressed using different examples. In this section, the path-finding problem is solved for the configuration of two cylindrical robots, and two revolute robots will be treated in Section 6.

3.1 Collision Detection for Two Cylindrical Robots

In Fig. 1, let $P_i(k)$ and $\phi_i(k)$, $i = 1, 2$, represent the position and angle of robot i at time k, respectively. Their d-step ahead predictions at time k are denoted as $\hat{P}_i(k+d/k)$, $\hat{\phi}_i(k+d/k)$. Suppose robot 1 is given the right-of-way, then a fictitious permanent colliding robot is defined with its position $P_c(k+d)$ and angle $\phi_c(k+d)$ as follows [2]:

$$P_c(k+d) = \sqrt{(d_{12})^2 + (\hat{P}_1(k+d/k))^2 - 2d_{12}\hat{P}_1(k+d/k)\cos(\hat{\phi}_1(k+d/k) - \phi_{01})} \tag{3.1}$$

$$\phi_c(k+d) = \arctan\frac{\hat{P}_1(k+d/k)\ \sin(\hat{\phi}_1(k+d/k))}{d_{12}\ -\ \hat{P}_1(k+d/k)\ \cos(\hat{\phi}_1(k+d/k))}. \tag{3.2}$$

To guarantee collision avoidance in the presence of tracking error and prediction error, position and angular safety margins are defined by $P_s \geq 0$ and $\phi_s \geq 0$. Without the loss of generality and also for simplicity, it is assumed $\phi_{01} = 0$ and $\phi_{02} = \pi$. There are then six different possible configurations for the two robots, and one of them is shown in Fig. 4. To detect a possible collision, the estimated angular margin is defined by

$$\Delta\hat{\phi}_s = \begin{cases} \Delta\hat{\phi} - \phi_s, & \text{if} \geq 0 \\ \Delta\hat{\phi} + \phi_s, & \text{if} < 0 \end{cases} \tag{3.3}$$

where $\Delta\hat{\phi} = (\phi_{02}\ -\ \hat{\phi}_2(k+d/k))\ -\ \phi_c(k+d)$.

Collision can therefore be detected by the following rules:

R3-1: **IF** $\phi_c(k+d) \geq 0$,
IF $\Delta\hat{\phi}_s \leq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c(k+d)|$
AND $(\hat{P}_2(k+d/k) \geq P_c(k+d) - P_s)$,
THEN a collision is detected.

R3-2: **IF** $\phi_c(k+d) < 0$,
IF $\Delta\hat{\phi}_s > 0$ **AND** $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c(k+d)|$
AND $(\hat{P}_2(k+d/k) \geq P_c(k+d) - P_s)$,
THEN a collision is detected.

3.2 Collision Avoidance of Two Cylindrical Robots

In this section, it is assumed that the position and angle predictions are available. (The design of such a predictor will be described in Section 4.)

The dynamics of each robot can be represented as a well-defined mapping. It is a mapping from reference input (desired trajectory) and historical output (actual position and angle) to the present output, though the internal structure and parameters of the mapping are unknown. This mapping can be conceptually represented as

$$\mathbf{Y}_i(k) \ = \ \mathbf{F}(\bar{\mathbf{Y}}_i,\ \bar{\mathbf{Y}}_{id}),\ \ i \ = \ 1,\ 2, \tag{3.4}$$

where $\bar{\mathbf{Y}}_i \ \equiv \ (\mathbf{Y}_i(k-1),\ \mathbf{Y}_i(k-2),\ \cdots,\ \mathbf{Y}_i(k-k_1))$,

$$\bar{\mathbf{Y}}_{id} \ \equiv \ (\mathbf{Y}_{id}(k),\ \mathbf{Y}_{id}(k-1),\ \cdots,\ \mathbf{Y}_{id}(k-k_2))\ .$$

$\mathbf{Y}_i(k) \equiv [P_i(k),\ \phi_i(k)]^T$ is the actual output of the robot i at time k, $\mathbf{F} :\ \mathbf{R}^2 \times \mathbf{R}^2 \rightarrow \mathbf{R}^2$, and $k_1,\ k_2 \geq 0$ are constant integers. We want to iteratively modify the reference inputs to the robots, $\mathbf{Y}_{id}$, by checking the possible collision in future. Because one of the two robots is given the right-of-way, the KBC only needs to modify the other robot's reference input for collision avoidance. The proposed algorithm for coordinating two robots to avoid collision is then given below, where the superscript i denotes the iteration count.

(1) Using the predictor, compute the predicted positions and angles of the two robots, $\hat{P}_1(k+d/k)$, $\hat{\phi}_1(k+d/k)$, $\hat{P}_2(k+d/k)$ and $\hat{\phi}_2(k+d/k)$.

(2) Compute the position and angle of the fictitious permanent colliding robot, $P_c(k+d)$ and $\phi_c(k+d)$, using Eqs. (3.1) and (3.2).

(3) Set $i \leftarrow 0$ and let $\hat{P}_2^i(k+d/k) \equiv \hat{P}_2(k+d/k)$, $\hat{\phi}_2^i(k+d/k) \equiv \hat{\phi}_2(k+d/k)$.

(4) Compute the angular margins, $\Delta\hat{\phi}_s$, using Eq. (3.3).

(5) Detect collision using R3-1 and R3-2. If no collision is detected, then terminate.

(6) Modify the reference input of robot 2 using the KBC, and compute the corresponding predicted position and angle of robot 2, $\hat{P}_2^{i+1}(k+d/k)$ and $\hat{\phi}_2^{i+1}(k+d/k)$.

(7) Set $i \leftarrow i+1$ and repeat steps (4) — (6).

The remaining problems are then how to design the predictor and how to modify the reference input for collision avoidance, which are the subjects of next section.

4 DESIGN OF THE NN-BASED PREDICTOR AND KNOWLEDGE-BASED COORDINATOR

We have already developed a general-purpose predictor in [6] using neural networks. The basic structure of the NN-based predictor is a multilayer perceptron with a modified back propagation algorithm. For the application of collision avoidance, for robot i, the inputs of the NN-based predictor are

$$P_{id}(k+i_1),\ \cdots P_{id}(k),\ P_{id}(k-1),\ \cdots P_{id}(k-i_2) \quad \text{and}$$
$$\phi_{id}(k+i_1),\ \cdots \phi_{id}(k),\ \phi_{id}(k-1),\ \cdots \phi_{id}(k-i_2) \quad i = 1,\ 2.$$

$i_1,\ i_2 \geq 0$ are constant integers. The outputs of the NN-based predictor are

$$\hat{P}_i(k+d/k),\ \hat{\phi}_i(k+d/k), \quad i = 1,\ 2, \quad \text{and} \quad d = 1,\ 2,\ \cdots,\ n.$$

Since one robot is given the right-of-way, only one reference input is modified at a time. Suppose $\mathbf{Y}_i^d \equiv [P_i^d,\ \phi_i^d]^T, i = 1,\ 2$, is a collision-free trajectory, then we want to modify the reference input $P_{id}(k)$ and $\phi_{id}(k)$ such that the actual output of the robot following the collision-free trajectory. Note that the collision-free trajectory $\mathbf{Y}_i^d$ is not specified explicitly, and the predicted tracking error is monitored by using the collision detection rules. Suppose only the position of robot 2 is modified, and the predicted position tracking error $\hat{P}_2(k+2/k)\ -\ P_2^d(k+d)$ is monitored. For the modification algorithm stated in Section 3, we have already developed a KBC in [5]. See [5] for detailed accounts of knowledge representation, inference process, solution existence, and system stability.

5 SIMULATION OF TWO CYLINDRICAL ROBOTS FOR COLLISION AVOIDANCE

To demonstrate the capability of the proposed scheme, two cylindric robots in a common workspace are simulated. The simulation arrangement is shown in Fig. 5, where both robots move simultaneously. Each simulated robot consists of the first and third link of a Stanford/JPL robotic manipulator [1], while keeping the other joint angles at zero. The simplified model is

$$\begin{aligned}\ddot{\phi}_i &= 0.7027\,\tau_{i1}\\ \ddot{P}_i &= 0.1379\,\tau_{i3} + 8.7555, \qquad i = 1,\ 2,\end{aligned}$$

where ϕ_i and P_i are the angle and position of robot i's joint 1 and 3, respectively, and τ_{i1}, τ_{i3} are the corresponding joint torque and force. For each robot, a PD controller is designed

$$\begin{aligned}\tau_{i1} &= 37.7(\phi_{id} - \phi_i) + 14.7(\dot{\phi}_{id} - \dot{\phi}_i),\\ \tau_{i3} &= 192.2(P_{id} - P_i) + 74.7(\dot{P}_{id} - \dot{P}_i) - 63.5.\end{aligned}$$

An NN-based predictor is designed for each robot. All the dynamic parameters and controller structure/parameters are unknown to both the NN-based predictor and the KBC. The NN-based predictor has six INPUT nodes, eight HIDDEN nodes, and six OUTPUT nodes. The outputs of the predictor are

$$\hat{P}_i(k+d/k), \qquad \hat{\phi}_i(k+d/k), \qquad d = 1,\ 2,\ 3,$$

and the inputs of the predictor are

$$P_{id}(k),\ P_{id}(k-1),\ P_{id}(k-2),\ \phi_{id}(k),\ \phi_{id}(k-1),\ \phi_{id}(k-2),$$

for robot i, $i = 1, 2$. In the KBC, 2-step ahead predictions are used to guide the modification of the reference inputs. With the safety margins $P_s = 0.2m$, $\phi_s = 0.25rad.$, and sampling interval $T_s = 0.01sec.$, under the KBC's coordination, the actual paths of the two robots are plotted in Fig. 5. Robot 1 follows its own desired trajectory while robot 2 moves cautiously to avoid collision. In the simulation, only the position reference input of robot 2 is modified. Note that instead of modifying the position reference input, either the angular reference input or robot 1's reference inputs can be modified if collision cannot be avoided by modifying robot 2's position reference input alone. The simulation is also performed in presence of process noise to test the noise-rejection ability of the KBC and the NN-based predictor, showing that the coordinator still works well.

6 COLLISION AVOIDANCE FOR TWO REVOLUTE ROBOTS

As mentioned earlier, in a two-dimensional workspace, a robot with two revolute joints is redundant. It is difficult to coordinate two revolute robots, especially when we have to rely solely on mathematical synthesis. However, even in such a case we can derive a set of rules for collision detection and avoidance. We want to keep the set of rules simple, but there is a tradeoff between the simplicity of rules and the conservativeness of collision detection. This tradeoff becomes more pronounced in the development of rules for two revolute robots working in a common workspace. In what follows, we will consider the case of two revolute robots with emphasis on the development of rules for collision detection and the strategy for collision avoidance.

Consider two revolute robots working in a two-dimensional workspace, as shown in Fig. 2. Let $q_{i1}(k)$, $q_{i2}(k)$ be the joint angles of robot i at time k, $i = 1,\ 2$, and $\hat{q}_{i1}(k+d/k)$ and $\hat{q}_{i2}(k+d/k)$ be their d-step ahead predictions, respectively. (In what follows, $\hat{q}_{i1}(k+d/k)$ and $\hat{q}_{i2}(k+d/k)$ will be represented by $\hat{q}_{i1}$ and $\hat{q}_{i2}$, respectively, to simplify the notation.) Two fictitious permanent colliding robots — corresponding to the two possible configurations of robot 1 in Fig. 2 — can be defined with positions P_{c0}, P_{c1} and angles ϕ_{c0}, ϕ_{c1},

$$\begin{aligned}P_{c0} &= \sqrt{(\hat{P}_{10})^2 + (d_{12})^2 - 2\,\hat{P}_{10}\,d_{12}\,\cos\hat{\phi}_{10}}, \quad \text{and}\\ \phi_{c0} &= \arctan\left(\frac{\hat{P}_{10}\ \sin\hat{\phi}_{10}}{d_{12} - \cos\hat{\phi}_{10}}\right), \end{aligned} \tag{6.1}$$

$$\begin{aligned}P_{c1} &= \sqrt{(L_{11})^2 + (d_{12})^2 - 2\,L_{11}\,d_{12}\,\cos\hat{q}_{11}}, \quad \text{and}\\ \phi_{c1} &= \arctan\left(\frac{L_{11}\ \sin\hat{q}_{11}}{d_{12} - \cos\hat{q}_{11}}\right), \end{aligned} \tag{6.2}$$

where L_{ij} is the length of link j of robot i, d_{12} the distance between the bases of two robots, and

$$\begin{aligned}\hat{P}_{i0} &= \sqrt{(L_{i1})^2 + (L_{i2})^2 + 2\,L_{i1}\,L_{i2}\,\cos\hat{q}_{i2}}, \qquad i = 1,\ 2,\\ \hat{\phi}_{i0} &= \arcsin\left(\frac{L_{i1}\sin\hat{q}_{i1} + L_{i2}\sin(\hat{q}_{i1} + \hat{q}_{i2})}{\hat{P}_{i0}}\right), \qquad j = 1,\ 2.\end{aligned}$$

We want to define a permanent colliding robot corresponding to different values of P_{c0}, P_{c1} and ϕ_{c0}, ϕ_{c1}. The position of the permanent colliding robot can be conservatively selected as

$$P_c = \min(P_{c0},\ P_{c1}), \tag{6.3}$$

and its angle ϕ_c will be defined according to different cases.

6.1 Rules for Collision Detection and Strategies for Collision Avoidance

There are three robot configurations depending on the angle of a permanent colliding robot: (1) $\phi_{c0} \geq 0$, $\phi_{c1} \geq 0$, (2) $\phi_{c0} < 0$, $\phi_{c1} < 0$, and (3) $\phi_{c0} \geq 0$, $\phi_{c1} < 0$ or $\phi_{c0} < 0$, $\phi_{c1} \geq 0$. For each of these cases, robot 2 can be represented by the predicted position $\hat{P}_2$ and the predicted angle $\hat{\phi}_2$, where

$$\hat{P}_2 = \max(\hat{P}_{20},\ L_{21}), \tag{6.4}$$

and $\hat{\phi}_2$ will be computed for different cases. For collision detection, the angular and length margins are defined by

$$\Delta\hat{\phi}_s = \begin{cases}\Delta\hat{\phi} - \phi_s, & \text{if } \Delta\hat{\phi} \geq 0\\ \Delta\hat{\phi} + \phi_s, & \text{if } \Delta\hat{\phi} < 0\end{cases} \qquad \text{and} \quad \Delta\hat{P}_s = P_c - (\hat{P}_2 + P_s), \tag{6.5}$$

where $\Delta\hat{\phi} = (\phi_{02} - \hat{\phi}_2) - \phi_c$. Then, a set of rules for collision detection are derived for each case. Fig. 6 shows one of the configurations of Case 1 with safety margins $\phi_s = 0$, $P_s = 0$. The rules for detecting collision between two revolute robots in a two-dimensional workspace are summarized below.

Case 1: Both $\phi_{c0} \geq 0$, and $\phi_{c1} \geq 0$

Define $\phi_c = \max(\phi_{c0},\ \phi_{c1})$ and $\hat{\phi}_2 = \max(\hat{\phi}_{20},\ \hat{q}_{21})$, then the rules are

R6–1: **IF** $\Delta\hat{\phi}_s \leq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

R6–2: **IF** $\Delta\hat{\phi}_s \leq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| > |\phi_c|$,
THEN set $\hat{\phi}_2 = \min(\hat{\phi}_{20},\ \hat{q}_{21})$, compute $\Delta\hat{\phi} = (\phi_{02} - \hat{\phi}_2) - \phi_c$
and $\Delta\hat{\phi}_s$ as in Eq. (6.5),
IF $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

Case 2: Both $\phi_{c0} < 0$, and $\phi_{c1} < 0$

Define $\phi_c = \min(\phi_{c0},\ \phi_{c1})$ and $\hat{\phi}_2 = \min(\hat{\phi}_{20},\ \hat{q}_{21})$, then the rules are

R6–3: **IF** $\Delta\hat{\phi}_s \geq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

R6–4: **IF** $\Delta\hat{\phi}_s \geq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| > |\phi_c|$,
THEN set $\hat{\phi}_2 = \max(\hat{\phi}_{20},\ \hat{q}_{21})$, compute $\Delta\hat{\phi} = (\phi_{02} - \hat{\phi}_2) - \phi_c$
and $\Delta\hat{\phi}_s$ as in Eq. (6.5),
IF $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

Case 3: $\phi_{c0} \geq 0,\ \phi_{c1} < 0$ **or** $\phi_{c0} < 0,\ \phi_{c1} \geq 0$.

Define $\phi_c = \max(|\phi_{c0}|,\ |\phi_{c1}|)$ and $\hat{\phi}_2 = \max(\hat{\phi}_{20},\ \hat{q}_{21})$, then the rules are

R6–5: **IF** $\Delta\hat{\phi}_s \leq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| \leq |2\ \phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

R6–6: **IF** $\Delta\hat{\phi}_s \leq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| > |2\ \phi_c|$,
THEN set $\hat{\phi}_2 = \min(\hat{\phi}_{20},\ \hat{q}_{21})$, compute $\Delta\hat{\phi} = (\phi_{02} - \hat{\phi}_2) - \phi_c$ and $\Delta\hat{\phi}_s$ as in Eq. (6.5),
IF $\left|\Delta\hat{\phi}_s\right| \leq |2\ \phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

Clearly, the rules for collision detection are much more complicated than those for two cylindrical robots because of the redundancy. However, these rules can be simplified if the desired trajectories of the two robots are given *a priori*, as discussed in Section 6.2.

It is assumed that the desired trajectories of both robots are planned while considering fixed obstacles (not the robots) in the workspace. Using a multi-step predictor and the rules developed above, a possible collision can be detected. Once a collision is detected, the desired trajectory of robot 2 will be modified in order to avoid the collision. Though either increasing or decreasing the speed of robot 2 may avoid the collision, the reasonable maneuver is to slow down robot 2 since the maximum speed and acceleration/deceleration are usually bounded. This implies that the reference input be modified in one direction (that is, increase or decrease). In order to give robot 2 a sufficient time so that it can maneuver to avoid the anticipated collision, the safety margins ϕ_s and P_s are added to the length and angular margins as in Eq. (6.5). Moreover, it is possible to modify the desired trajectory of one joint to avoid collision, which will in turn simplify the modification process of the reference inputs.

6.2 Simulation of Two Revolute Robots for Collision Avoidance

The arrangement of the simulated robots is given in Fig. 7, and the dynamic and kinematic parameters of the two robots are identical. It is assumed that each robot is equipped with a PD–type servo controller, and the sampling interval is 0.01 sec. The desired trajectories are symmetric, and their values at some special points are given in the following table, where the subscript d represents the desired value.

Robot 1	Joint 1			Joint 2		
	q_{11d}	$\dot{q}_{11d}$	$\ddot{q}_{11d}$	q_{12d}	$\dot{q}_{12d}$	$\ddot{q}_{12d}$
initial values, $t = t_0$	90°	0	0	0°	0	0
middle point, $t = t_f/2$	0°	max	max	90°	0	0
final values, $t = t_f$	−90°	0	0	0°	0	0

Robot 2	Joint 1			Joint 2		
	q_{21d}	$\dot{q}_{21d}$	$\ddot{q}_{21d}$	q_{22d}	$\dot{q}_{22d}$	$\ddot{q}_{22d}$
initial values, $t = t_0$	90°	0	0	0°	0	0
middle point, $t = t_f/2$	180°	max	max	−90°	0	0
final values, $t = t_f$	270°	0	0	0°	0	0

An NN–based predictor is designed for each robot. There are six INPUT nodes with inputs

$$q_{i1d}(k),\ q_{i1d}(k-1),\ q_{i1d}(k-2),\ q_{i2d}(k),\ q_{i2d}(k-1),\ q_{i2d}(k-2),$$

eight HIDDEN nodes and six OUTPUT nodes with outputs,

$$\hat{q}_{i1}(k+d/k),\ \hat{q}_{i2}(k+d/k), \quad d = 1,\ 2,\ 3,$$

for robot i, $i = 1,\ 2$.

In this example, the moving directions of both robots are the same. Therefore, one can always derive a suitable permanent colliding robot by choosing

$$\phi_c = \max(\phi_{c0},\ \phi_{c1}), \quad \text{and} \quad P_c = \min(P_{c0},\ P_{c1}).$$

Then the rules for collision detection can be simplified and are listed below.

(1) Compute $\hat{\phi}_2 = \max(\hat{\phi}_{20},\ \hat{q}_{21})$, and $\hat{P}_2 = \max(\hat{P}_{20},\ L_{21})$.

(2) Compute angular and length margins $\Delta\hat{\phi}_s$, and $\Delta\hat{P}_s$ as in Eq. (6.5).

(3) Detect collision using the following rules.

R6–7: **IF** $\phi_c \geq 0$ **AND** $\Delta\hat{\phi}_s \leq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

R6–8: **IF** $\phi_c < 0$ **AND** $\Delta\hat{\phi}_s \geq 0$ **AND** $\left|\Delta\hat{\phi}_s\right| \leq |\phi_c|$ **AND** $\Delta\hat{P}_s \leq 0$,
THEN a collision is detected.

The angular and length safety margins are set to $\phi_s = 0.2$ *rad.* and $P_s = 0.05$ *m*, and 2–step ahead predictions are used for collision detection. Fig. 8 shows the actual trajectory of robot 2, showing the slow–down of robot 2 to avoid collision.

7 CONCLUSION

Combining the techniques of intelligent control and neural networks, a new hierarchical scheme is proposed to coordinate multiple robots in a common workspace to avoid collision. The high level includes the proposed KBC and NN-based predictor, and the low level consists of the robots to be coordinated. The KBC foresees the effects of each reference input on the low level by using the predictor, and modifies the reference input guided by the predictions in order to avoid collision among the robots.

The proposed scheme assumed that both path planning and trajectory planning did not consider collision avoidance, and adding the KBC did not impose any constraints on the design of the servo controller. This may relax the usual requirements (for example, the knowledge of exact dynamics) imposed on path planning, trajectory planning, and servo controllers design. Other constraints, such as joint torque limits and master/slave assignment, can be easily added to the knowledge base. Since the internal structure and parameters of the individual robot's control system are not affected, one can coordinate multiple robots — equipped with built–in servo controllers — working in a common workspace without collision. The simple structure and algorithm, no constraints on the design of individual robot control systems, and good simulation results make the proposed scheme attractive for many industrial applications.

References

[1] A. K. Bejczy, "Robot Arm Dynamics and Control", *JPL Technical Report 33 – 669*, p. 93, Pasadena, CA, Fed. 1974.

[2] E. Freund and H. Hoyer, "Real–Time Pathfinding in Multirobot Systems Including Obstacle Avoidance", *The Int'l J. of Robotics Research*, 7(1): 42 – 70, Feb. 1988.

[3] O. Khatib, "Real Time Obstacle Avoidance for Manipulator and Mobile Robots", *The Int'l J. of Robotics Research*, 5(1): 90 – 98, Spring 1986.

[4] K. G. Shin and Q. Zheng, "Minimum Time Trajectory Planning for Dual Robot Systems", *Proc. of the 28th IEEE Conf. on Decision and Control*, pp. 2506 – 2511, Dec. 1989.

[5] K. G. Shin and X. Cui, "Design of a Knowledge–based Controller for Intelligent Control Systems", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 21, No. 2, pp. 368 – 375, March/April, 1991.

[6] K. G. Shin and X. Cui, "Design of a General–Purpose MIMO Predictor with Neural Networks", submitted for publication and a summary was published in the *Proc. of the 13th IMACS World Congress on Computation and Applied Mathematics*, July 1991, Dublin, Ireland.

[7] R. B. Tilove, "Local Obstacle Avoidance for Mobile Robots Based

on the Method of Artificial Potential", *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pp. 566 – 571, 1990.

[8] C. W. Warren, ""Multiple Robot Path Coordination Using Artificial Potential Fields", *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pp. 500 – 505, 1990.

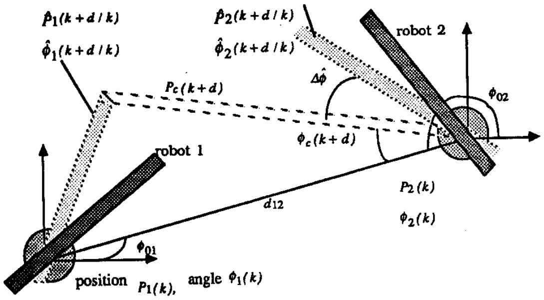

Fig. 1. Basic configuration of two robots working in a common workspace.

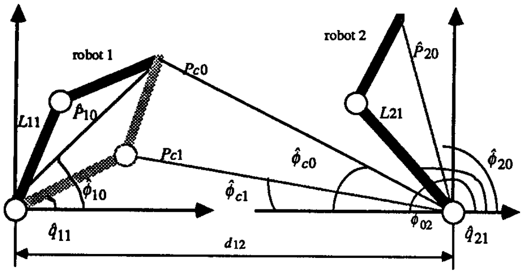

Fig. 2. Definitions of variables for two revolute robots.

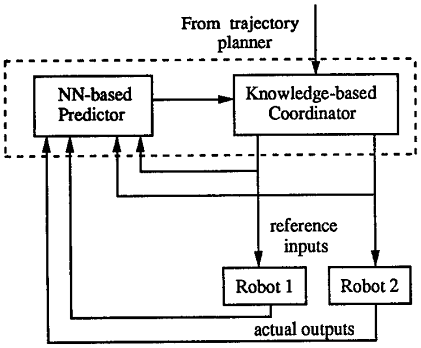

Fig. 3. Conceptual structure of the knowledge-based coordination system.

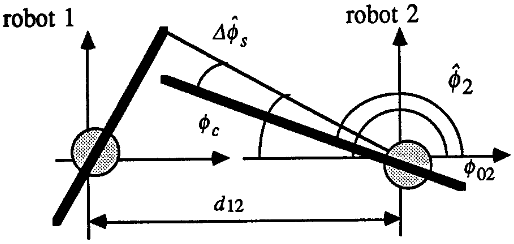

Fig. 4. Collision detection.

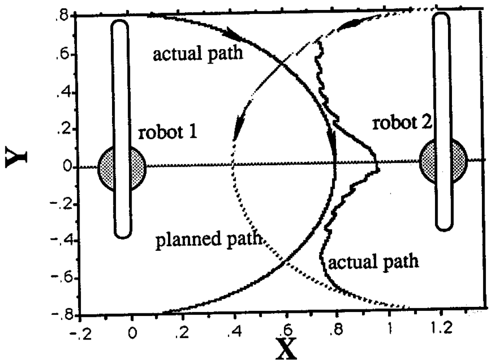

Fig 5. Actual paths of the two robots.

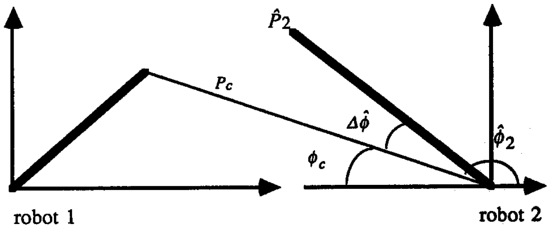

Fig. 6. Collision detection for two revolute robots.

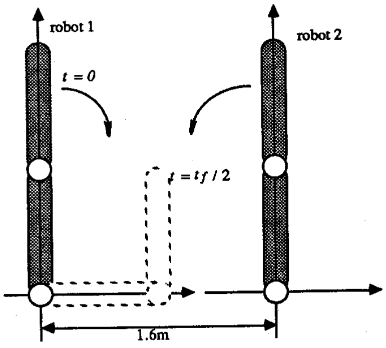

Fig. 7. Configuration of two revolute robots in a common workspace.

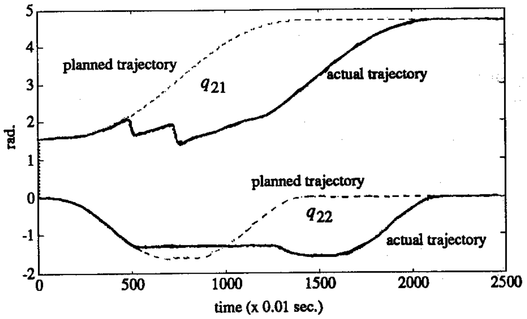

Fig. 8. The actual trajectories of robot 2 with the KBC.