CSE HANDBOOK
CRC Press
1996

33

Robotics

33.1 Introduction 734
33.2 Robot Workcells 735
33.3 Workcell Command and Information Organization 737
Intelligent Control Architectures • Behaviors and Hybrid Systems Design • Workcell Planning, Coordination, and Control Structure
33.4 Commercial Robot Configurations and Types 740
Manipulator Performance • Common Kinematic Configurations • Drive Types of Commercial Robots • Commercial Robot Controllers
33.5 Robot Kinematics, Dynamics, and Servo-level Control 744
Kinematics and Jacobians • Robot Dynamics and Properties • Robot Servo-level Motion Control • Robot Force/Torque Servocontrol • Motion Trajectory Generation
33.6 End Effectors and End-of-Arm Tooling 752
Part Fixtures and Robot Tooling • Grippers and Fingers • Robot Wrist Mechanisms • Robot/Tooling Process Integration and Coordination
33.7 Sensors 756
The Philosophy of Robotic Workcell Sensors • Types of Sensors • Sensor Data Processing • Vision for Robotics
33.8 Workcell Planning 762
Workcell Behaviors and Agents • Task Decomposition and Planning • Task Matrix Approach to Workcell Planning • Path Planning
33.9 Job and Activity Coordination 772
Matrix Rule-Based Job Coordination Controller • Process Integration, Digital I/O, and Job Coordination Controller Implementation • Coordination of Multiple Robots
33.10 Error Detection and Recovery 775
Error Detection • Error Recovery
33.11 Human Operator Interfaces 776
Levels of User Interface • Mechanisms for User Interface
33.12 Robot Workcell Programming 777
Robot Programming Languages • V+, A Representative Robot Language
33.13 Mobile Robots and Automated Guided Vehicles 780
Mobile Robots • Automated Guided Vehicle Systems

F. L. Lewis
University of Texas at Arlington

M. Fitzgerald
University of Texas at Arlington

K. Liu
University of Texas at Arlington

33.1 Introduction

The word *robot* was introduced by the Czech playright Karel Čapek in his 1920 play *Rossum's Universal Robots.* The word *robota* in Czech means simply work. In spite of such practical beginnings, science fiction writers and early Hollywood movies have given us a romantic notion of robots and expectations that they will revolutionize several walks of life including industry. However, many of the more far-fetched expectations from robots have failed to materialize. For instance, in underwater assembly and

0-8493-2909-4/97/$0.00+$.50

Path Planning

The path planning problem [Latombe 1991] may be decomposed into motion path planning, grasp planning, and error detection and recovery; only the first is considered here. *Motion path planning* is the process of finding a continuous path from an initial position to a prescribed final position or goal without collision. The output of the path planner for robotic workcells is a set of path *via points* which are fed to the machine trajectory generator (discussed previously). *Off-line path planning* can be accomplished if all obstacles are stationary at known positions or moving with known trajectories. Otherwise, *on-line or dynamic* path planning is required in real time; this often requires techniques of *collision or obstacle avoidance*. In such situations, paths preplanned off line can often be modified to incorporate collision avoidance. This subsection deals with off-line path planning except for the portion on the potential field approach, which is dynamic planning. See Zhou [1996] for more information.

Initial and final positions may be given in any coordinates, including the robot's joint space. Generally, higher level workcell components think in terms of Cartesian coordinates referred to some world frame. The Cartesian position of a robot end effector is given in terms of three position coordinates and three angular orientation coodinates; therefore, the general 3D path planning problem occurs in $\Re^6$. If robot joint-space initial and final positions are given one may work in *configuration space*, in which points are specified by the joint variable vector $\mathbf{q}$ having coordinates $\mathbf{q}_i$, the individual joint values. For a six-degrees-of-freedom arm, configuration space is also isomorphic to $\Re^6$. Path planning may also be carried out for initial and final values of *force/torque*. In 3D, linear force has three components and torque has three components, again placing the problem in $\Re^6$. Hybrid position/force planning is also possible. In this subsection path planning techniques are illustrated in $\Re^2$, where it is convenient to think in terms of planning paths for mobile robots in a plane.

If the number of degrees of freedom of a robot is less than six, there could be problems in that the manipulator may not be able to reach the prescribed final position and the via points generated in the planning process. Thus, the path planner must be aware of the limitations of the individual robots in its planning process; in fact, it is usually necessary to select a specific robot agent for a task prior to planning the path in order to take such limitations into account.

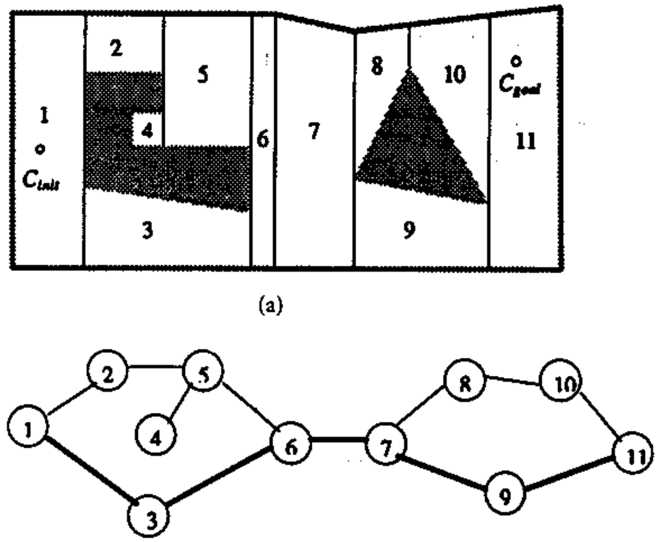

FIGURE 33.25 Cell decomposition approach to path planning: (a) free space decomposed into cells using the vertical-line-sweep method and (b) connectivity graph for the decomposed space. (*Source:* Courtesy of Zhou, C. 1996. Planning and intelligent control. In *CRC Handbook of Mechanical Engineering.* F. Kreith, ed. CRC Press, Boca Raton, FL.)

Cell Decomposition Approach

In the *cell decomposition approach to path planning,* objects are enclosed in polygons. The object polygons are expanded by an amount equal to the radius of the robot to ensure collision avoidance; then, the robot is treated simply as a moving point. The free space is decomposed into simply connected free-space regions within which any two points may be connected by a straight line. When the Euclidean metric is used to measure distance, convex regions satisfy the latter requirement. A sample cell decomposition is shown in Fig. 33.25. The decomposition is not unique; the one shown is generated by sweeping a vertical line across the space. Based on the decomposed space, a *connectivity graph* may be constructed, as shown in the figure. To the graph may be added weights or costs at the arcs or the nodes, corresponding to distances traveled, etc. Then, graph search techniques may be used to generate the shortest, or otherwise least costly, path.

Road Map Based on Visibility Graph

In the road map approach the obstacles are modeled as polygons expanded by the radius of the robot, which is treated simply as a moving point. A **visibility graph** is a nondirected graph whose nodes are the vertices of the polygons and whose links are straight line segments connecting the nodes without intersecting any obstacles. A *reduced visibility graph* does not contain links that are dominated by other links in terms of distance. Figure 33.26 shows a reduced visibility graph for the free space. Weights may be assigned to the arcs or nodes and graph search techniques may be used to generate a suitable path. The weights can reflect shortest distance, path smoothness, etc.

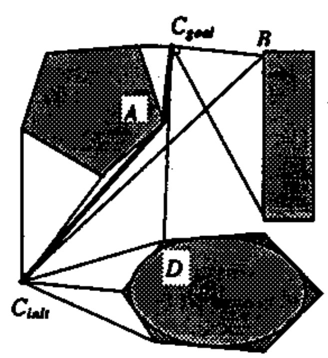

FIGURE 33.26 Road map based on visibility graph. (*Source:* Courtesy of Zhou, C. 1996. Planning and intelligent control. In *CRC Handbook of Mechanical Engineering.* F. Kreith, ed. CRC Press, Boca Raton, FL.)

Road Map Based on Voronoi Diagram. A **Voronoi diagram** is a diagram where the path segment lines have equal distance from adjacent obstacles. In a polygonal space, the Voronoi diagram consists of straight lines and parabolas: when both adjacent object segments are vertices or straight lines, the equidistant line is straight, when one object is characterized by a vertex and the other by a straight line, the

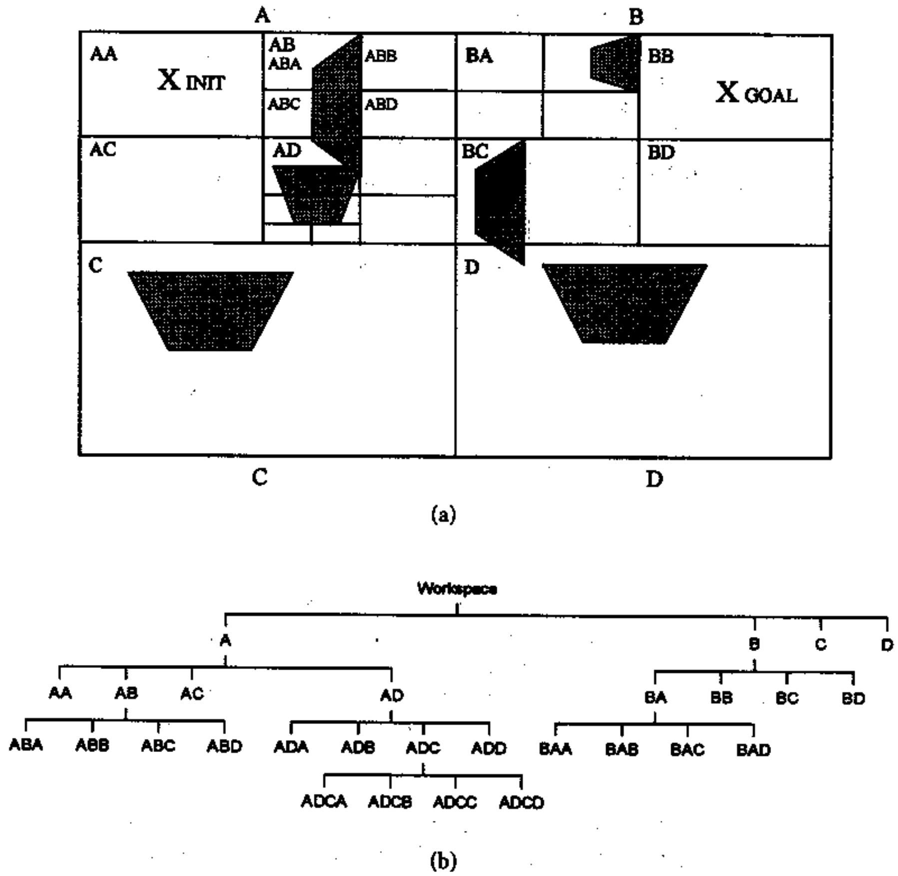

FIGURE 33.27 Quadtree approach to path planning: (a) quadtree decomposition of the work area and (b) quadtree constructed from space decomposition.

equidistant line is parabolic. In the Voronoi approach, generated paths are generally longer than in the visibility graph approach, but the closest point of approach (CPA) to obstacles is maximized.

Quadtree Approach

In the quadtree approach, Fig. 33.27(a) rectangular workspace is partitioned into four equal quadrants labeled A, B, C, D. Suppose the initial point is in quadrant A with the goal in quadrant B. If there are obstacles in quadrant A, it must be further partitioned into four quadrants AA, AB, AC, AD. Suppose the initial point is in AA, which also contains obstacles; then AA is further partitioned into quadrants AAA, AAB, AAC, AAD. This procedure terminates when there are no obstacles in the quadrant containing the initial point. A similar procedure is effected for the goal position. Based on this space decomposition, the quadtree shown in Fig. 33.27(b) may be drawn. Now, tree search methods such as A^* may be used to determine the optimal obstacle-free path.

The quadtree approach has the advantage of partitioning the space only as finely as necessary. If any quadrant contains neither goal, initial point, nor obstacles, it is not further partitioned. If any quadrant containing the initial position or goal contains no obstacles, it is not further partitioned. In 3D, this approach is called *octree*.

Maneuvering Board Solution for Collision Avoidance of Moving Obstacles

The techniques just discussed generate a set of via points between the initial and final positions. If there are moving obstacles within the free-space regions, one may often modify the paths between the via points online in real-time to avoid collision. If obstacles are moving with constant known velocities in the free space, a technique used by the U.S. Navy based on the *maneuvering board* can be used for on-line obstacle avoidance. Within a convex free-space region, generated for instance by the **cell decomposition** approach, one makes a *relative polar plot* with the moving robot at the center and other moving objects plotted as straight lines depending on their relative courses and speeds. A steady bearing and decreasing range (SBDR) indicates impending collision. Standard graphical techniques using a parallel ruler allow one to alter the robot's course and/or speed to achieve a prescribed CPA; these can be converted to explicit formulas for required course/speed changes. An advantage of this technique for mobile robots is that the coordinates of obstacles in the relative polar plot can be directly measured using onboard sonar and/or laser range finders. This technique can can be modified into a *navigational technique* when some of the stationary obstacles have fixed absolute positions, such obstacles are known as *reference landmarks*.

Potential Field Approach

The potential field approach [Arkin 1989] is especially popular in mobile robotics as it seems to emulate the reflex action of a living organism. A fictitious attractive potential field is considered to be centered at the goal position [Fig. 33.28(a)]. Repulsive fields are selected to surround the obstacles [Fig. 33.28(b)]. The sum of the potential fields [Fig. 33.28(c)] produces the robot motion as follows. Using $\mathbf{F}(\mathbf{x}) = ma$, with m the vehicle mass and $\mathbf{F}(\mathbf{x})$ equal to the sum of the forces from the various potential fields computed at the current vehicle position $\mathbf{x}$, the required vehicle acceleration $a(\mathbf{x})$ is computed. The resulting motion avoids obstacles and converges to the goal position. This approach does not produce a global path planned a priori. Instead, it is a real-time on-line motion control technique that can deal with moving obstacles,

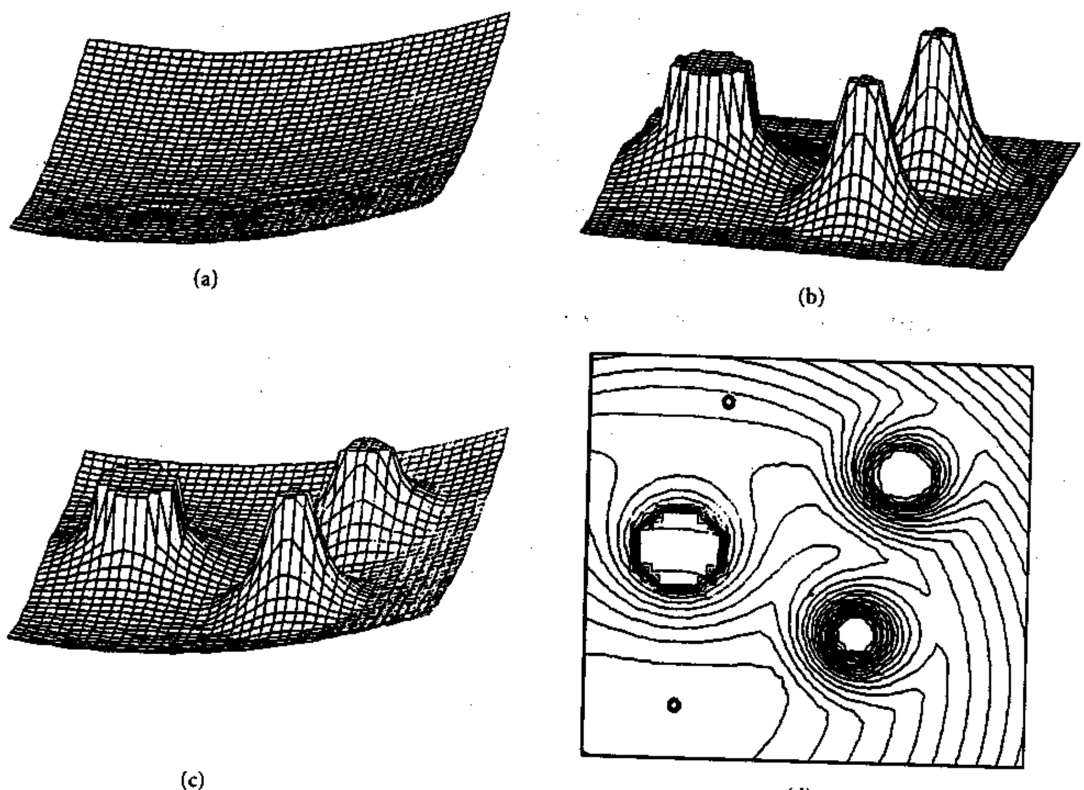

FIGURE 33.28 Potential field approach to navigation: (a) attractive field for goal at lower left corner, (b) repulsive fields for obstacles, (c) sum of potential fields, and (d) contour plot showing motion trajectory. (*Source:* Courtesy of Zhou, C. 1996. Planning and intelligent control. In *CRC Handbook of Mechanical Engineering*. F. Kreith, ed. CRC Press, Boca Raton, FL. With permission.)

particularly if combined with maneuvering board techniques. Various methods have been proposed for selecting the potential fields; they should be limited to finite influence distances, or else the computation of the total force $\mathbf{F}(\mathbf{x})$ requires knowledge of all obstacle relative positions.

The potential field approach is particularly convenient as the force $\mathbf{F}$ may be computed knowing only the *relative positions* of the goal and obstacles from the vehicle; this information is directly provided by onboard sonar and laser readings. The complete potential field does not need to be computed, only the force vector of each field acting on the vehicle. A problem with the potential field approach is that the vehicle may become trapped in *local minima* (e.g., an obstacle is directly between the vehicle and the goal); this can be corrected using various techniques, including adding a dither force to get the vehicle out of these false minima. The potential field approach can be combined with Lyapunov analysis techniques to integrate the path planning and trajectory following servocontrol functions of a mobile robot [Jagannathan et al. 1994].

In fact, Lyapunov functions and potential fields may simply be added in an overall controls design technique.

Emergent Behaviors. The responses to individual potential fields can be interpreted as *behaviors* such as seek goal, avoid obstacle, etc. Potential fields can be selected to achieve specialized behaviors such as *docking* (i.e., attaining a goal position with a prescribed angle of approach) and remaining in the center of a corridor (simply define repulsive fields from each wall). The sum of all of the potential fields yields an *emergent behavior* that has not been preprogrammed (e.g., seek goal while avoiding obstacle and remaining in the center of the hallway). This makes the robot exhibit behaviors that could be called intelligent or self-determined.