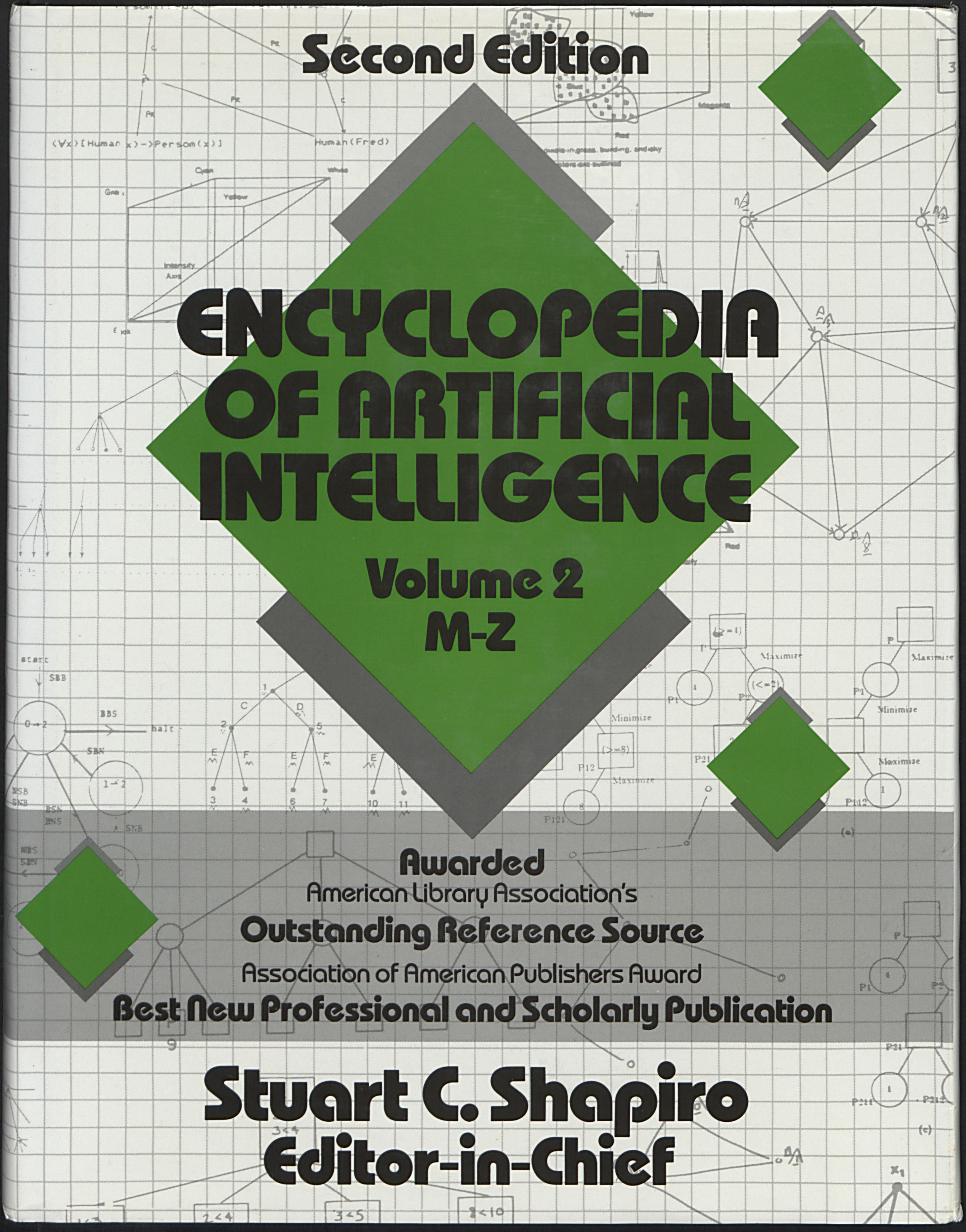

Second Edition

ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE

Volume 2
M–Z

Awarded
American Library Association's
Outstanding Reference Source
Association of American Publishers Award
Best New Professional and Scholarly Publication

Stuart C. Shapiro
Editor-in-Chief

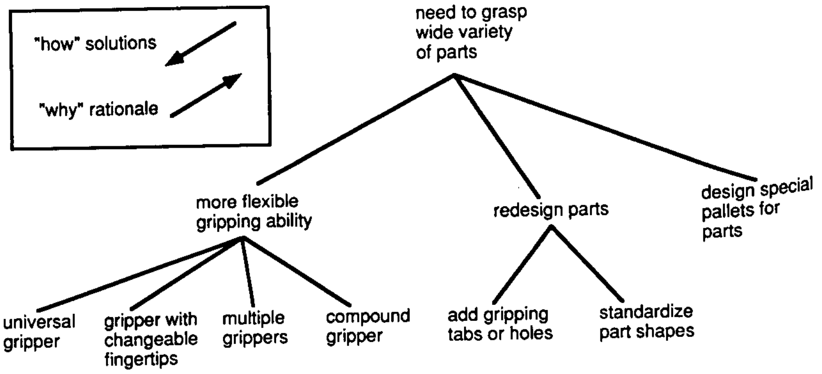
"how" solutions
"why" rationale
need to grasp wide variety of parts
more flexible gripping ability
redesign parts
design special pallets for parts
universal gripper
gripper with changeable fingertips
multiple grippers
compound gripper
add gripping tabs or holes
standardize part shapes

Figure 15. A "how/why" diagram of solutions and rationale for a design problem involving a need to grasp a wide range of parts.

parts, in which the robot, the parts, the end-effector, and the environment are all candidates for redesign.

BIBLIOGRAPHY

J. D. Abramowitz, J. W. Goodnow, and B. Paul, "Pennsylvania Articulated Mechanical Hand," *International Conference on Computers in Engineering,* ASME, Chicago, Ill., Aug. 1983.

F. Y. Chen, "Force Analysis and Design Consideration of Grippers," *Industrial Robot* (UK) **9,** 243–249 (Dec. 1982).

M. R. Cutkosky, "On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks," *IEEE Trans. Robotics Autom.* **5,** 269–279 (June 1989).

M. R. Cutkosky and P. K. Wright, "Friction Stability and the Design of Robotic Fingers," *Int. J. Robotics Res.* **5,** 20–37 (1986*a*).

M. R. Cutkosky and P. K. Wright, "Active Control of a Compliant Wrist in Manufacturing Tasks," *ASME J. Eng. Indust.* **108,** 36–43 (1986*b*).

M. R. Cutkosky, J. M. Jourdain, and P. K. Wright, "Skin Materials for Robotic Fingers," *Proceedings of the IEEE Conference on Robotics and Automation,* Raleigh, N.C., Mar. 1987, pp. 1649–1654.

M. R. Cutkosky, E. Kurokawa, and P. K. Wright, "Programmable Comformable Clamps," *Autofact 4,* Philadelphia, Pa., 11.51–11.58 (Nov. 1982).

P. Datseris and W. Palm, "Principals on the Development of Mechanical Hands Which Can Manipulate Objects by Means of Active Control," *Paper 84-DET-37,* ASME, 1984.

R. L. Eckert, "Codes and Classification Systems," in N. L. Hyer, ed., *Group Technology at Work,* Society of Manufacturing Engineers, Dearborn, Mich., 1984, pp. 43–51.

E. Glassman, R. L. Hollis, and R. H. Taylor, "Compliant Gripper," *IBM Tech. Discl. Bull.* **27**(1B), 772–773 (June 1984).

S. Hirose and Y. Umetani, "The Development of Soft Gripper for the Versatile Robot Hand," *Mech. Mach. Theory* (GB), 351–358 (1978).

J. M. Hollerbach, "Robot Hands and Tactile Sensing," in Grimson and Patil, eds., *AI in the 1980's and Beyond,* MIT Press, Cambridge, Mass., 1987, pp. 317–342.

R. L. Hollis, A. P. Allan, and S. Salcudeanm, "A Six Degree-of-Freedom Magnetically Levitated Variable Compliance Fine Motion Wrist," in R. Bolles and B. Roth, eds., *Robotics Research, the Fourth International Symposium,* MIT Press, Cambridge, Mass., 1988, pp. 65–73.

R. Howe and M. R. Cutkosky, "Touch Sensing for Robotic Manipulation and Recognition," submitted to *Robotic Review* (May 1990).

T. Iberall and C. MacKenzie, "Opposition Space and Human Prehension," in *Dextrous Robot Hands,* Springer-Verlag, New York, 1990, Chapt. 2.

S. Jacobsen, J. Wood, D. F. Knutti, and K. B. Biggers, "The Utah/M.I.T. Dextrous Hand: Work in Progress," *Int. J. Robotics Res.* **3**(4) (Winter 1984).

R. S. Johansson and G. Westling, "Roles of Glabrous Skin Receptors and Sensorimotor Memory in Automatic Control of Precision Grip when Lifting Rougher or More Slippery Objects," *Exp. Brain Res.* **56,** 550–564 (1984).

I. Kato, *Mechanical Hands Illustrated,* Hemisphere Publishing Corporation, New York, 1982.

G. M. Kenaley and M. R. Cutkosky, Electrorheological Fluid-Based Robotic Fingers with Tactile Sensing," *Proceedings of the IEEE Conference on Robotics and Automation,* Scottsdale, Ariz., May 14–19, 1989, pp. 132–136.

J. Kerr and B. Roth, "Analysis of Multifingered Hands," *Int. J. Robotics Res.* **4,** 3–17 (1986).

L. Leifer, "Rehabilitative Robots," *Robotics Age,* 4–15 (May/June 1981).

Z. Li, P. Hsu, and S. Sastry, "Grasping and Coordinated Manipulated by a Multifingered Hand," *Int. J. Robotics Res.* **8,** 33–50 (1989).

G. Lundstrom, B. Glemme, and B. W. Rooks, *Industrial Robots—Gripper Review,* International Fluidics Services Ltd., 35–39 High Street, Kempston, Bedford, England, 1977.

J. Napier, "The Evolution of the Hand," *Sci. Am.* **207,** 56–62 (Dec. 1962).

H. R. Nicholls and M. H. Lee, "A Survey of Robot Tactile Sensing Technology," *Int. J. Robotics Res.* **8,** 3–30 (June 1989).

T. Okada, "Object Handling System for Manual Industry," *IEEE Trans. Syst. Man Cybern.* **SMC-9,** 79–89 (Feb. 1979).

E. Pearce, E. Pfaff, J. McManus, and A. Yee, *Flexible Robotic Servo Gripper,* Final report, Master's engineering design project, Mechanical Engineering Dept., Stanford University, 1986.

P. Rabischong, "Phylogeny of the Hand," in R. Tubiana, ed., *The Hand,* W. B. Saunders Co., Philadelphia, Penn., 1981, Chapt. 4.

J. K. Salisbury, "Kinematic and Force Analysis of Articulated Hands," in M. T. Mason and J. K. Salisbury, eds., *Robot Hands and the Mechanics of Manipulation,* MIT Press, Cambridge, Mass., 1985, section I.

Simrit Corp., *Pneumatically Operating Gripping and Clamping Elements,* Arlington Hts., Ill., 1984.

Robots

David Nitzan, ROBOTICS, 1375-1398

From Shapiro, Stuart C., Editor-in-Chief, Encyclopedia of Artificial Intelligence, 2nd Ed., John Wiley & Sons, Inc., NY, 1992.

ROBOTICS

The objective of this article is to survey the state of the art of intelligent robots. By way of introduction, the terms robot and artificial intelligence are defined, intelligent robots are classified according to their level of intelligence, social and technoeconomic incentives for the development of intelligent robots are discussed, and the article touches on the socioeconomic impacts of this development. Past accomplishments and present issues in major robotics research areas are covered.

ROBOT CHARACTERISTICS

Capability, Components, and Intelligence

Several definitions for the term robot have been proposed (Jablonowski and Posey, 1985). None of these definitions are adequate because they exclude robot intelligence of any kind. Hence, the following definition is proposed (Nitzan and co-workers, 1983):

> A robot is a general-purpose machine system that, like a human, can perform a variety of different tasks under conditions that may not be known a priori.

Being a general-purpose machine system, the terms robot and robot system are regarded as synonymous. A robot system may include any of the following major functional components:

Effectors. "Arms," "hands," "legs," and "feet" (see ROBOT HANDS AND END EFFECTORS; ROBOTS, LEGGED).

Sensors. Contact and noncontact (see SENSORS AND SENSOR FUSION).

Computers. Top-level controller and lower level controllers (including communication channels).

Auxiliary Equipment. Tools, jigs, fixtures, tables, pallets, conveyors, part feeders, etc.

A robot (or robot system) is controlled by a single top-level computer (or controller). A group of such systems, which may or may not interact, are regarded as separate robots if they are not controlled by the same top-level computer; adding a single computer above them will merge these systems into a single robot.

Although a robot performs some human tasks and there is a similarity between the functional components of a robot and those of a human (or a human team), a robot is not required to act or look like a human. It should, however, be able to perform tasks that require flexibility and artificial intelligence. Flexibility means the ability to perform a class of different tasks; artificial intelligence means the ability of a machine system to perceive conditions that may not have been known *a priori*, to decide what actions should then be performed, and to plan those actions accordingly. Some of the potential robot tasks can be performed by humans; others cannot (eg, in a high radiation environment).

Classification

Like human intelligence, robot intelligence is variable. This observation is compatible with the Japanese classification of industrial robots into five categories (Sadamoto, 1981):

A slave manipulator teleoperated by a human master.

A limited-sequence manipulator (further classified into hard-to-adjust and easy-to-adjust categories).

A teach-replay robot.

A computer-controlled robot.

An intelligent robot.

Learning from Biological Systems

Current robot capabilities to act, sense, and think are in many respects inferior to those of animals, in general, and humans, in particular. By studying biological systems, principles may be discovered that can be used, perhaps by analogy, to improve the functional components of a robot as well as their cooperation. Using such a bionic approach may lead to improvements in robot effectors (eg, the automatic feedback control of cooperative flexible arms, fingers, and legs or the dexterity of multifinger hands), sensors (eg, integration of several sensors in parallel) (see SENSORS AND SENSOR FUSION), and computer processing, eg, representation of knowledge (see KNOWLEDGE REPRESENTATION) and reasoning (qv). These improved capabilities will advance sensor-guided manipulation (eg, picking one of jumbled objects in a bin), perception (eg, recognizing, locating, and inspecting objects in cluttered environments or outdoor scenes), and other activities in which these components are integrated.

Incentives for Intelligent Robot Development

The important incentives for the developments of intelligent robots are social and technoeconomic.

Social Incentives. The most important incentive for developing robots should be social, replacing humans who perform undesired jobs by machines. Japan, for example, is planning to embark on a large-scale program for the development of robots operating in hazardous environments (Umetani and Yonemoto, 1983). The ranking of robot development should thus be ordered according to job undesirability, ie, jobs that are

Lethal (eg, in a high radiation environment).

Harmful (eg, paint spraying, handling toxic chemicals).

Hazardous (eg, combat, fire fighting).

Strenuous (eg, lifting heavy loads or visual inspection).

Noisy (eg, forging, riveting)

Dull (eg, sorting, assembling).

Technoeconomic Incentives. The second most important incentive for robot development is technoeconomic: reducing the cost of manufacturing products and improving their quality.

Current Limitations. In spite of the strong social and economic incentives mentioned above, only a very small fraction of the entire human workforce in the world has been replaced by industrial robots. Furthermore, Engleberger (1980) estimates that the growth rate of the total number of industrial robots (excluding teleoperators (qv) and limited-sequence manipulators) will rise from 2,000 per year in 1980 to 40,000 per year in 1990; these figures correspond to a yearly replacement of about 0.003–0.06% of the total blue-collar workforce in the industrialized countries. Such a low rate of growth of robot population has resulted primarily from the following limitations of today's industrial robots (Nitzan and co-workers, 1983).

Insufficient Material-Handling Flexibility. Workpieces and other objects can be handled only if they are indexed within tolerances that match the accuracy of the robot manipulator. Such restriction limits the flexibility of manufacturing, especially in batch production of a mix of products.

Open-Loop Control. Jobs that require closed-loop feedback control to correct local errors cannot be performed. For example, today's arc-welding robots cannot track a joint of randomly variable shape and gap in one pass and adjust the torch movement and welding parameters accordingly; this limitation excludes these robots from a large market.

Inability to Detect and Correct Errors. Detection of unexpected errors and the recovery from them cannot be done; a robot system cannot verify that all the robot actions have been executed as planned. The resulting penalty may be costly. For example, the cost of debugging and repairing a final assembly may be several orders of magnitude higher than the cost of correcting that error in process, whereas subassemblies are easily accessible.

Restricted Mobility. The locomotion of today's robotic carts is restricted to fixed guidance (eg, by buried cables or painted lines). These carts cannot navigate freely, avoid obstacles, or find their targets in an unstructured environment. Such a restriction limits the flexibility of material handling in batch production (see Robots, mobile).

Future Capabilities. The best way to overcome the limitations of today's muscle-only robots is to provide them with intelligence, ie, adaptive sensing and thinking capabilities. Such intelligent robots will be able to compete more effectively with not only blue-collar workers but also white-collar workers. Most industrial companies have not yet agreed with this observation but they will, eventually, when the threat of worldwide market competition becomes unbearable.

Socioeconomic Problems. Development of intelligent robots may raise many problems, the major one of which is unemployment (Ayres and Miller, 1983). Obstructing development of intelligent robots by the labor unions will only worsen the unemployment problem because other countries, especially Japan, will proceed with such development and, as a result, foreign competition will become stronger. This complex problem will probably be mitigated by three factors (Nitzan and co-workers, 1976).

New Related Jobs. An increased demand for skills related to intelligent robots directly (eg, engineering, computer programming, and manufacturing) and indirectly (eg, professional training, marketing, shipping, and servicing).

New Unrelated Jobs. A shift to other jobs, especially in the service industry (thus raising the standard of living).

Fewer Working Hours. Reducing the working hours per week with no reduction in the standard of living.

Whether these factors will be able to solve the unemployment problem remains to be seen. In the meantime, the current rate of intelligent-robot development is low, amounting to robot evolution rather than robot revolution. Such evolution will enable society to adjust gradually, without adverse repercussions, to the advent of the intelligent robot.

Technical Approach. The technical approach to intelligent robot development should be based on the application of AI techniques to robotics under four engineering constraints:

High Reliability. The robot must be robust; if it fails, it should be able to detect the error and recover from it or call for help.

High Speed. The robot should be able to perform its functions as fast as necessary.

Programmability. The robot should be flexible (able to perform a class of different functions for a variety of tasks), easily trainable (for new tasks or modification of old ones), and intelligent (able to perceive problems and solve them).

Low Cost. The cost of the robot should be low enough to justify its application.

Clearly, these constraints may conflict with each other. For example, increasing the robot speed or lowering its cost may also lower its reliability. A trade-off, therefore, must be engineered for different applications according to the significance of each constraint.

RESEARCH AND DEVELOPMENT TOPICS

As shown above, a robot system may be divided into effectors, sensors, computers, and auxiliary equipment. Robotics research and development topics associated with these major functional components include manipulation (of arms), end effectors, and mobility; sensing (in general), noncontact sensing, and contact sensing; adaptive control (which utilizes sensors to monitor and guide effector actions); and robot programming and manufacturing process planning (which generate task-specific computer programs that are executed by the top-level and lower level controllers). Past achievements and research issues related to each of these topics are described briefly in the following sections.

Manipulation

Robot manipulation entails the kinematics, motion trajectories, dynamics, and control of a robot arm.

Kinematics. The location (position and orientation) of a robot wrist in a frame attached to the base of the robot arm is described in two ways:

Joint Coordinates. The angles of the rotary joints and the lengths of the sliding joints of the arm.

World Coordinates. The Cartesian coordinates of the wrist position and the direction cosines defining the wrist orientation.

Joint coordinates must be used to command the robot arm to arrive at a given wrist location. On the other hand, humans prefer to describe the wrist location in terms of world coordinates. Hence, means are provided for transforming from one set of coordinates to another.

Joint-to-World Coordinate Transformation. A frame of Cartesian coordinates (x, y, z) is attached to every arm joint according to a set of rules proposed by Denavit and Hartenberg (1955). The homogeneous coordinates $(x, y, z, 1)$ of a given point in each joint frame are converted to those of a neighboring one by means of a transform: a 4×4 homogeneous coordinate-transformation matrix. Multiplying the transforms of all the arm joints results in the arm-to-wrist transform whose elements, expressed in terms of the joint coordinates, describe the position and orientation (direction cosines) of the arm wrist (Rosen and co-workers, 1974; Paul, 1981).

World-to-Joint Coordinate Transformation. Given the position and orientation of the wrist of an arm, solving for the corresponding joint coordinates is less systematic and more difficult than vice versa. Each arm has a unique solution in which joint coordinates are computed sequentially in a fixed order (Rosen and co-workers, 1974).

Motion Trajectories. *Workstation Transforms.* Denoting the homogeneous coordinates $(x, y, z, 1)$ of a point P in a frame F by $P(F)$, then $P(F_1) = [F_1/F_2] * P(F_2)$, where $[F_1/F_2]$ is the 4×4 transform from frame F_1 to frame F_2 (Fig. 1a). Note that $P(F_2) = [F_2/F_1] * P(F_1)$, where $[F_2/F_1]$ is the inverse of $[F_1/F_2]$.

Consider a robot arm with an end effector mounted on its wrist moving in a workstation, and let A, W, E, and C denote frames attached to the arm, the wrist, the end effector, and the current (instantaneous) action target of the robot, respectively. Given transforms $[A/C]$, $[C/E]$, and $[W/E]$, the unknown transform $[A/W]$ is computed from the relation $[A/W] = [A/C] * [C/E] * [E/W]$, where $[E/W]$ is the inverse of $[W/E]$ (Fig. 1b). Knowing $[A/W]$, the corresponding joint coordinates are then computed using the world-to-joint arm solution.

Smooth Path. Paul (1975) developed a technique for moving the robot end effector along a path consisting of straight segments and smooth transitions between them (Paul, 1981; Rosen and co-workers, 1976; Taylor, 1979). A straight segment is obtained by interpolating world and joint coordinates between its end locations while maintaining a constant linear velocity. A smooth transition between two straight segments is obtained by bypassing their intersection point along a parabolic curve tangent to both segments. Each pair of straight segments may be defined in either a single frame or in two frames, each of which may be moving with a constant velocity relative to the arm frame. For example, a robot spot-welding gun may move with velocity V_1 along a straight line in the arm frame toward a moving conveyor, veer and change its velocity smoothly, and continue to move with velocity V_2 along a second straight line in the conveyor frame toward a welding spot. Training for a task to be performed on a moving line is done while the workpiece is stationary.

Dynamics. The dynamics of a manipulator relate the applied forces–torques to the joint motion (positions, velocities, and accelerations as functions of time). Two dynamics problems are distinguished (Featherstone, 1983):

The Forward Problem. Given the applied forces and torques, solve for the resulting joint motions.

The Inverse Problem. Given the joint motions, solve for the required forces–torques.

The inverse problem is more important for real-time arm control and, fortunately, easier than the forward problem.

Different numerical methods have been proposed for solving the equations of motion of an n-joint manipulator. These methods are assessed on the basis of their computational complexity (the number of additions and multiplications required). The methods are based on either the Lagrangian formulation (Paul, 1981; Hollerbach, 1980) or the Newton-Euler formulation (Luh and co-workers, 1980). The Lagrangian formulation is less efficient, but Silver (1982) showed that the two formulations are essen-

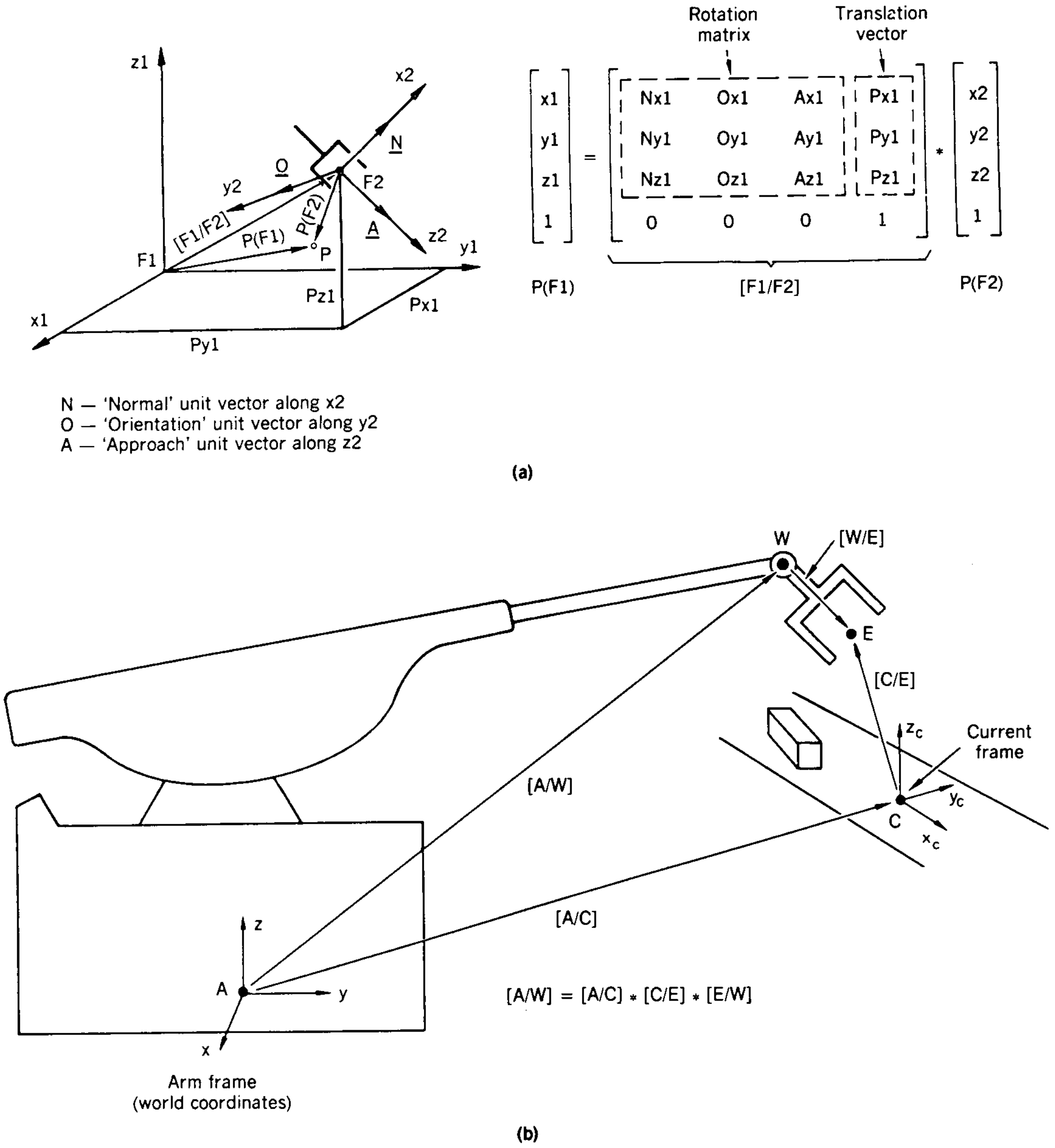

Figure 1. Arm-transformation matrices: (*a*) Transformation matrix: N = normal unit vector along *x*2, O = orientation unit vector along *y*2, A = approach unit vector along *z*2. (*b*) Matrix description of arm system.

tially equivalent and that the complexity of computation depends on its structure and the representation of the rotational dynamics. Kane and Levinson (1983) proposed another approach that uses explicit dynamic equations without the unnecessary computations entailed in either of the above formulations and, hence, should be more efficient.

Control. The motion of a manipulator in a free space is usually controlled by means of a position servo in each joint. If the manipulator is required to move while exerting a specified force/torque on an object, appropriate force/torque servo and position servo must be executed simultaneously (Paul, 1981). Such a hybrid control can be achieved by replacing the position servos of selected joints by force/torque servos so that the manipulator is free to move in the specified direction while the prescribed forces/torques are applied to the object-surface normals. For example, grinding a horizontal surface requires simultaneous control of the position of the grinding wheel and the vertical force it exerts. As another example, inserting a peg into a hole requires motion along their common axis while exerting zero forces along two directions normal to it. Stopping the motion of a manipulator can be controlled by specifying a given force/torque condition.

End-Effectors

An end-effector is a functional device attached to the wrist of a robot arm (see ROBOT HANDS AND END-EFFECTORS). Four types of end-effector are distinguished: hand, tool, hand/tool holder, and micromanipulator. Each of these should be small, light, fast, accurate, multifunctional, and inexpensive.

Hand. A hand, the major function of which is to grasp objects, includes a number (eg, two or three) of fingers attached to its palm. Each finger should have humanlike structural dexterity and rigidity (bone), object-grasping compliance (flesh), surface tactile sensors (skin), and proximity sensors for collision avoidance (no human equivalence). The hand may be equipped with a wrist force sensor, a visual sensor (eye), a range sensor, or any other sensor. To minimize the number of wires between the hand and the robot controller, local signals and hand functions should be processed by microprocessors mounted on the hand itself (resulting in a smart hand).

Three-finger hands have been built at the Electrotechnical Laboratory (Okada and Tsuchiya, 1977) and at Stanford University (Salisbury and Craig, 1982). The Salisbury hand is shown in Figure 2. A kinematic analysis of the latter hand yields a large number (373) of different ways the hand can grasp an object. Hand control is very complex; it is equivalent to the control of three cooperating three-joint arms with force sensors.

Tool. A tool may be a spot-welding gun, an arc-welding torch, a wrench, or any other device that performs a certain task. If a sensor is mounted on or near the tool, it must be ensured that neither one will prevent the other from access to its target. For example, using visual sensing to guide a robot to arc weld workpieces with corners may require that the sensor and the welding torch be free to move relative to each other.

Hand/Tool Holder. A hand/tool holder is a device mounted between the arm wrist and a hand or a tool for one of two purposes:

Quick hand/tool changing and mounting for different tasks, which may be achieved by a standard latch–unlatch device or a hand.

Local accommodation, which may be implemented passively by a remote center compliance (RCC) device, developed at Draper Laboratories (Drake, 1977), or actively by a force/torque sensor. For example, in Figure 3 plastic parts are assembled by two two-finger PUMA (Unimation, Inc.) hands, one mounted on an RCC (made by Lord Corp.) and guided locally by its eye and the other mounted on an *xyz* force/torque sensor.

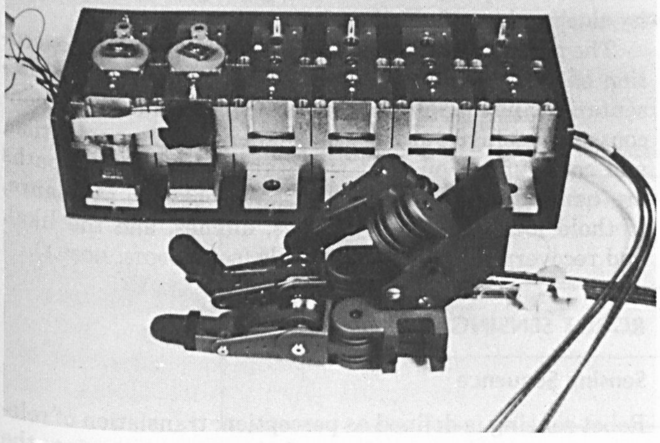

Figure 2. Three-finger hand. Courtesy of Salisbury, MIT.

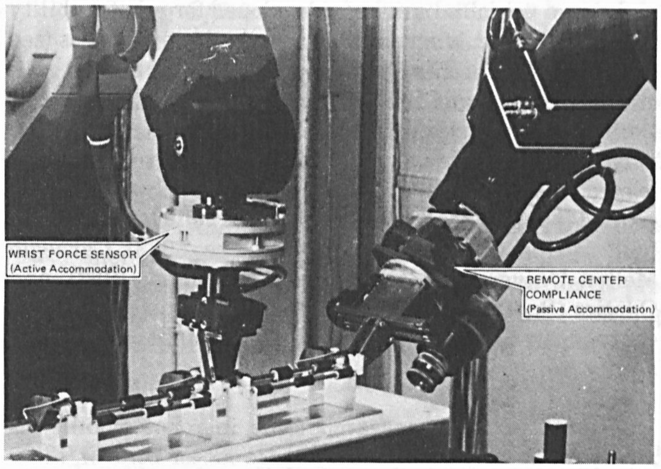

Figure 3. PUMA hands with passive and active accommodation devices.

Micromanipulator. The function of a micromanipulator is to correct a locational error that is measured by a robot sensor and is smaller than the spatial resolution of the arm wrist. Having a much smaller inertia, the micromanipulator is also much faster than the arm. A micromanipulator may be an *xyz* device or a multifinger hand.

Mobility

Most industrial robots today are anchored to fixed locations; a few have a limited mobility on tracks mounted on the factory floor or on a gantry (see Robots, mobile). There are also mobile carts that transport workpieces, but these carts can move only in a structured environment, eg, by following buried cables or painted lines. Robot mobility, however, is also needed for a wide variety of robot functions in unstructured environments, such as mining, military operations, and aid to the handicapped. Some robot-mobility issues follow.

Surfaces and Locomotion. The mechanism for the robot locomotion depends strongly on the type of surface the robot must be able to move on. Indoor surfaces include floors, ramps, stairs, and cluttered environments. Outdoor surfaces include roads, smooth ground (flat and slanted), terrain with holes and ditches, and terrain with large obstacles.

Robot locomotion is realized with wheels, tracks, and legs. Wheels perform well if the terrain is not rough and the traction is sufficient. Tracks perform well if the terrain slope is not too high or no major obstacles are encountered. In a recent development at Hitachi (Iwamoto and co-workers, 1983) the loop of each track forms a triangle whose shape is adjusted according to the terrain, thus enabling the vehicle to pass over different obstacles and climb up and down stairways.

Legged vehicles have been developed for robot mobility in rough terrain, where wheels and tracks are useless (see Robots, legged). The major issues are stability, gait, strength, speed, and control. Static stability is achieved if the vertical projection of the center of gravity is within the polygon formed by the vehicle's feet on the ground. A six-legged vehicle with at least three legs always on the ground is inherently stable. Six-legged vehicles have been built at Ohio State University (McGhee, 1983), Carnegie Mellon University (Raibert and Sutherland, 1983), and Odetics (Bartholet, 1983). The Odetics vehicle (Fig. 4) is characterized by high strength-to-weight ratio and agility.

Control. A major research issue in robot mobility is autonomous control, which includes motor control, sensing, navigation, communication, obstacle avoidance, and task performance. SRI's Shakey the Robot (qv) (Nitzan, 1981) was developed in the 1960s as an intelligent mobile robot with these properties. Shakey (Fig. 5) had autonomous wheel-drive control and visual, range, and binary tactile sensors; navigated through laboratory rooms; communicated with its brain (a DEC PDP-10 computer) via a radio link; avoided obstacles; and pushed boxes according to the plan of a task it was assigned to do. Since then other similar robots have been developed in the United States (eg, at Carnegie Mellon University and MIT), France (eg, LAAS in Toulouse), and Japan (eg, Mechanical Engineering Laboratory in Tsukuba). These robots use wheels to move on laboratory floors and shaft encoders to sense their two-dimensional positions. To correct for random locational errors, some of them use infrared beacons, a directable laser range finder, and visual perception. Sonar sensors (which are inexpensive) are commonly used to avoid obstacles, but they are not adequate for navigation because of their poor resolution, short range, and vulnerability to specular reflection.

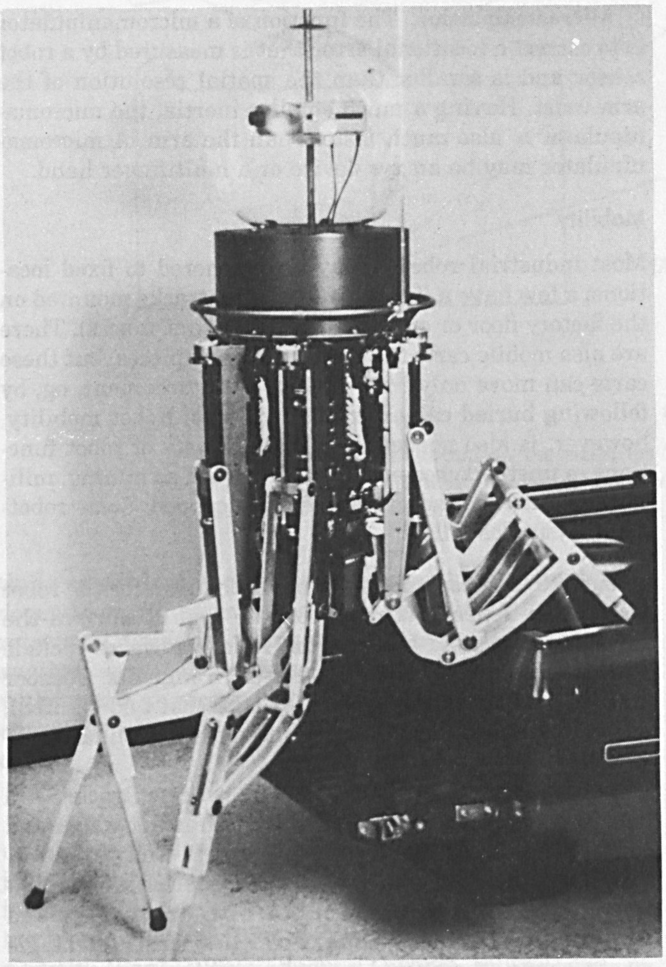

Figure 4. Six-legged walking vehicle. Courtesy of Odetics, Inc. (Anaheim, Calif.)

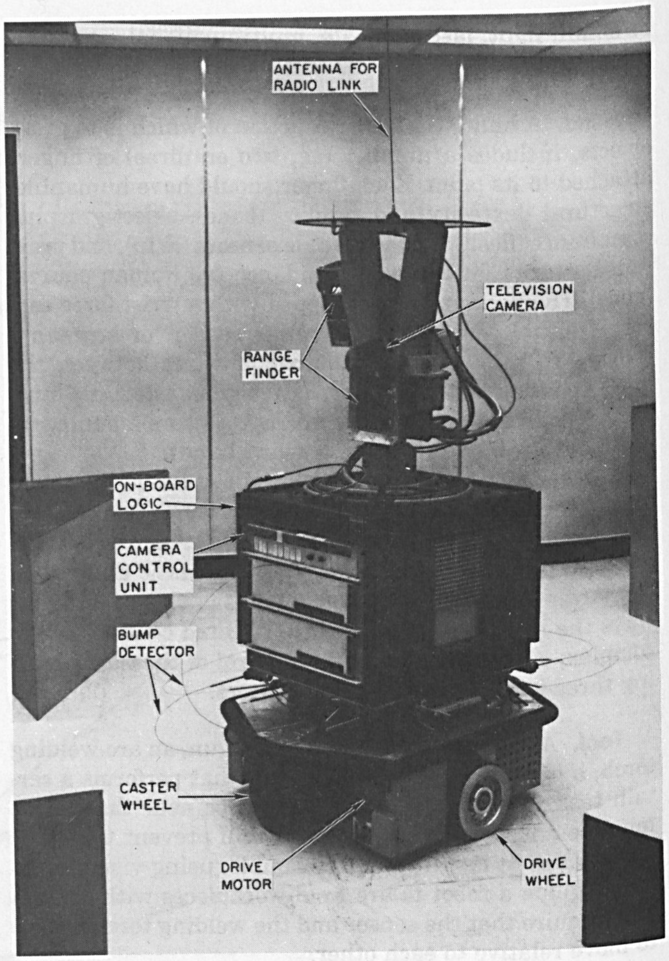

Figure 5. SRI's Shakey, a mobile robot.

The primary research issues in planning for and execution of robot mobility along long-range paths are representation and mapping of a three-dimensional world, noncontact sensing, outdoor visual perception, navigation, and communication, and those along short-range paths are noncontact and contact sensing, obstacle avoidance, foothold location (to avoid holes, ditches, and the like), and recovery from accidental falls (see Robots, mobile).

ROBOT SENSING

Sensing Sequence

Robot sensing is defined as perception: translation of relevant characteristic or relational object properties into the information required to control the robot in performing a given robot function (Nitzan and co-workers, 1983) (see also SENSORS AND SENSOR FUSION). The object properties may be geometric, mechanical, optical, acoustic, material, electric, magnetic, chemical, and the like. Robot functions may be passive (eg, RECOGNIZE, LOCATE, and INSPECT) or active (eg, GRASP, TRANSPORT, and WELD). Each of these functions may be expanded by lower level robot functions (eg, RECOGNIZE = (TAKE-PICTURE, FIND-EDGE, EXTRACT-FEATURES, INTERPRET-FEATURES)), or be used to define higher level ones (eg, using the functions FIND, ACQUIRE, HOLD, MOVE, ALIGN, INSERT, and VERIFY to define ASSEMBLE). As shown in Figure 6, a robot sensing sequence is performed in the following steps.

1. *Transducing*. Converting (in hardware) the relevant object properties into a signal.
2. *Processing*. Transforming the signal into the required information, usually in two substeps: preprocessing, improving the signal (usually in hardware), eg, filtering out noise, and interpreting, analyzing the improved signal and extracting the required information (usually in software).

The above steps probably cannot be implemented by a general-purpose system. Instead, each step or substep should be performed by effective hardware–software schemes, regarded as tools, that depend on the environmental conditions and the specified robot functions. To be able to carry out a wide variety of sensing tasks, the sensing system will consist of sets of tools, or toolboxes, and a knowledge-based supervisor that can select the best tools for each given task.

Sensing Strategy

If the extracted information is not sufficient, the sensing sequence is modified and repeated in order to obtain complementary information. Three cases are distinguished as follows.

1. *Supplementary Images*. Imaging additional surfaces, which are hidden from a fixed sensor in the previous sensing sequence(s), by means of a sensor mounted on the robot end effector or multiple sensors at different viewpoints.
2. *Sensing Efficiency*. Achieving efficient sensing by first using coarse resolution and then fine resolution, such as, recognizing an object, then locating it precisely; and recognizing and locating an object approximately, then inspecting some of its windows, where distinctive features or defects may be found, with fine resolution.
3. *Multisensing*. Utilizing different sensors to supplement a sensor output, such as recognizing and locating an object with vision and then locating it precisely and verifying its grasp with tactile sensing, ie, verifying the sensing of one sensor by another.

Sensor Signals

A given object property may be measured by different sensor signals, such as light intensity, range, acoustic, tactile, force, and temperature (Table 1). A point signal is distinguished from an array (one- or two-dimensional) of point signals. As shown in Table 2, each of these signals may be generated by different sensor transducers, eg, a point light intensity may be transduced by a photocell, a photomultiplier, a 1-dimensional array, or a two-dimensional array.

Research Issues

Robot sensing could be advanced by developing

Sensor transducers that have a higher resolution, higher speed, smaller size, and lower cost.

Faster hardware–software processors that can process a larger amount of sensor signals and extract more information.

Sensor modeling and planning, including sensor selection, for a given task and off-line signal prediction.

NONCONTACT SENSING

Noncontact sensing is based on a signal generated by a transducer that is not in physical contact with the object it senses. Noncontact sensing is classified according to the type of signal, ie, light intensity (or, briefly, intensity), range, acoustic, temperature, chemical, etc. Noncontact sensing for robot applications has so far been based primarily on intensity and range signals; future robot applications should also utilize the other types of signals.

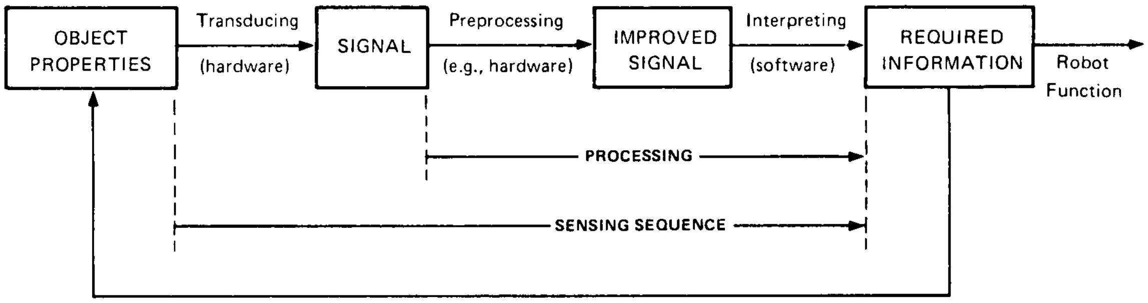

Figure 6. Robot sensing diagram.

Table 1. Measurement of Object Properties: Possible Signals for Each Object Property

	Signal										
Object Property	Intensity (Point/Array)		Range (Point/Array)		Acoustic (Point/Array)		Tactile (Point/Array)		Force (Point)	Temperature (Point/Array)	
Geometric											
Centroid		×		×				×			
Edge, corner		×		×		×		×			
Surface		×		×		×		×			
Volume				×		×					
Width		×		×		×	×	×			
Texture		×		×		×		×			
Shape		×		×		×		×			
Proximity			×	×	×	×	×	×			
Mechanical											
Weight									×		
Force–torque									×		
Pressure							×	×	×		
Optical											
Reflectance	×	×									
Color	×	×									
Acoustic											
Reflectance					×	×					
Material											
Hardness							×	×	×	×	×
Temperature										×	×

The output of the signal-transduction step is an image if it consists of a two-dimensional array of sensory data values. Thus it is possible to distinguish between an intensity image, which consists of $N \times M$ picture elements, or pixels, and a range image, which consists of $N \times M$ range elements, or rangels, where N and M are resolution integers. An intensity image provides information about the reflectance of object surfaces in the scene, but it may be ambiguous geometrically because of the loss of one-dimensional information in the process of transforming a three-dimensional world into a two-dimensional gray-level image. A range image, on the other hand, provides three-dimensional information directly but no reflectance information. Intensity and range images are thus complementary and should, therefore, be in exact registration to simplify the analysis of their integrated information.

Table 2. Measurement of Object Properties: Possible Sensor Transducers for Each Signal

Signal	Transducers
Intensity	
Point	Photocell, photomultiplier, array (1-D, 2-D)
Array	Array or equivalent (lower dimensional array scanning)
Range	
Point	Projector (laser, planar light)/receiver (photomultiplier, array, two arrays), acoustic
Array	Scanning projector (laser, planar light)/receiver (photomultiplier, array), two 2-D arrays or equivalent
Acoustic	
Point	Acoustic transducer
Array	Array of acoustic transducers or equivalent
Tactile	
Point	Microswitch, array of tactile transducers
Array	Array of tactile transducers or equivalent
Force (point)	Force transducer
Temperature	
Point	Thermocouple, infrared transducer
Array	Array of infrared transducers or equivalent

Intensity transducers include a photocell, a photomultiplier, a one-dimensional array camera, and a two-dimensional array camera. Intensity transducers require the following improvements (Nitzan, 1981).

Chips with higher precision, improved quality, color discrimination, higher resolution (eg, 1024 × 1024 or even 2048 × 2048 pixels).

Lenses with lower distortion and better focus in the infrared region.

Fast, computer-controlled adaptive lens opening and focusing as well as intensity thresholding.

Indirect measurement of range or surface orientation may be inferred from molecular two-dimensional images under certain conditions (Mundy, 1975; Jarvis, 1983; Nitzan and co-workers, 1986; Horn, 1975); but this subject is beyond the scope of this article. The following section focuses on direct range measurement and its current problems.

Direct Range Measurement

Two basically different techniques can be used to measure range directly: triangulation and time of flight.

Triangulation Techniques. Triangulation is based on elementary geometry (Fig. 7a): given the baseline of a triangle, ie, the distance between two of its vertices, and the angles at these vertices, the range from one of the vertices

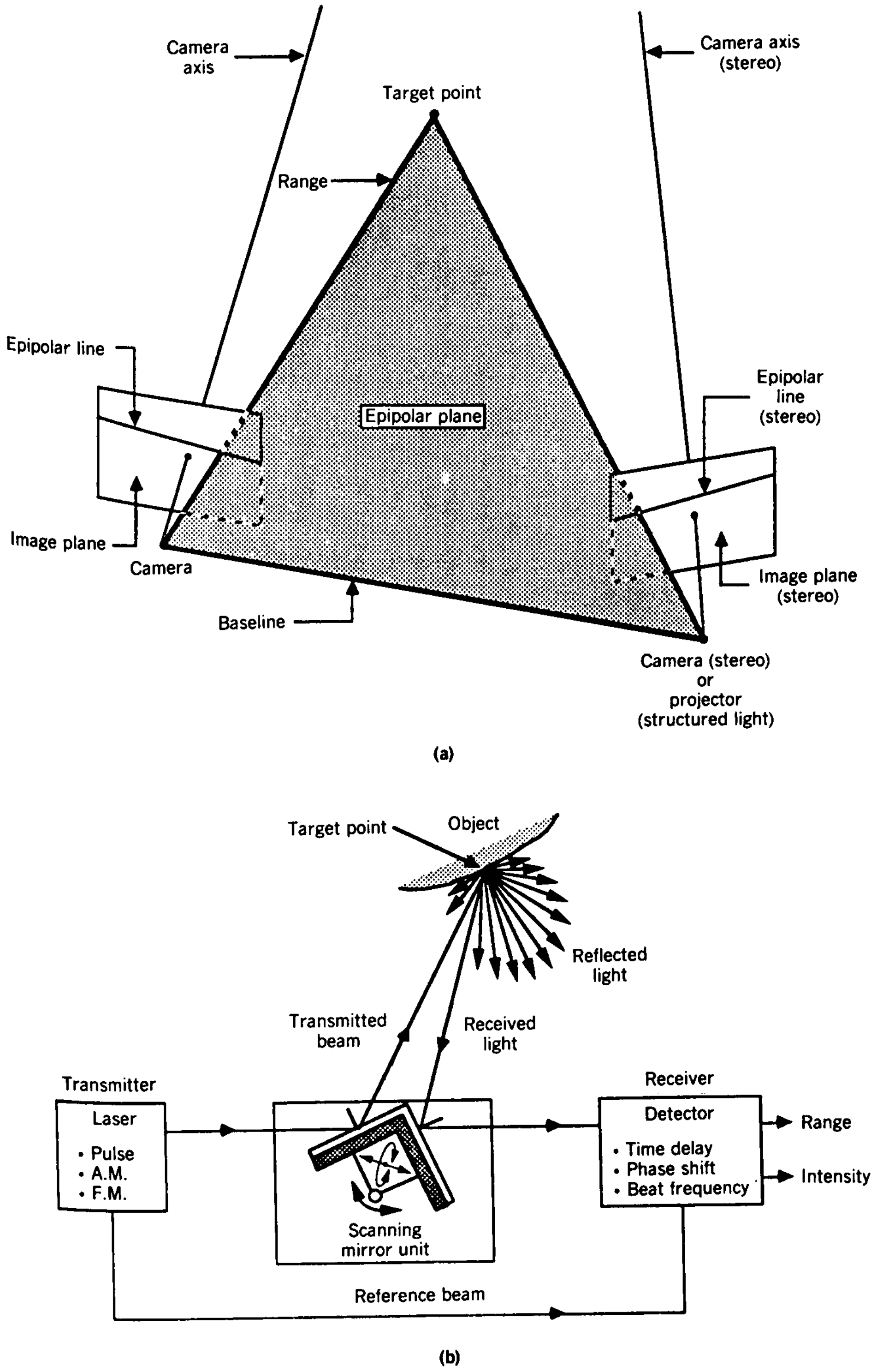

Figure 7. Direct range-sensing schemes: (*a*) Triangulation range sensing (Nitzan and co-workers, 1977). (*b*) Time-of-flight laser range sensing. Courtesy of Bolles, SRI International.

to the third one is computed as the corresponding triangle side. Triangulation techniques are subdivided into two schemes: stereo, using ambient light and two cameras (a passive scheme), and structured light, using a projector of controlled light and camera (an active scheme). The plane in which the triangle lies is called the epipolar plane, and its line of intersection with a camera image plane is called the epipolar line.

The main drawbacks of any triangulation technique are missing data for points in the scene that are not seen from both vertices of the triangle. This problem can be partially solved in two ways:

Decreasing the baseline (this remedy, however, will increase the measurement errors).

Using multiple cameras for the stereo scheme or multiple projectors and cameras for the structured light scheme (this provision will also reduce the measurement errors and mitigate the problem of occlusion, including self-occlusion, in machine vision, but it will increase the cost, complexity, and measurement time of the system).

Stereo. Relying on passive ambient light, triangulation stereo techniques use an image sensor (in particular, a TV camera) at each of two triangle vertices. A stereo pair of images can be obtained either from two static cameras (at different locations) or from one camera that is moved between two locations.

In addition to the missing-data problem, the main issue in stereo vision is the correspondence problem: how to match corresponding points in stereo images reliably and quickly. This problem has no solution if the two images have uniform reflectance. Conversely, the correspondence problem becomes easier as the stereo images include more intensity features, such as edges, especially if they are perpendicular to the epipolar line. These features should be extracted on the basis of microconstraints as well as macroconstraints. For example, local intensity changes imply edge points, but if these points are too isolated to be linked into a continuous edge, they should be disregarded. The effect of the correspondence problem is an increase in the measurement time.

Structured Light. One way to dispose of the correspondence problem is to use active light: a scheme in which one of the stereo cameras is replaced by a source of specially controlled illumination, called structured light. The structured light may be projected serially, by scanning a collimated light beam (usually a laser), or in parallel, either by diverging a laser beam with a cylindrical lens or by using a slit or a slide projector. The structured light may consist of single or multiple light patterns, each of which may be a straight line (a beam), planar, or nonplanar.

In addition to the missing-data problem, the structured light scheme entails two issues:

Specular Reflection. Reflection from a mirrorlike surface may result in no range measurement if the reflected light does not reach the camera and false (larger or smaller) measured range values if the reflected light is subsequently reflected by other surfaces before part of it reaches the camera.

Slow Measurement.. Serial projection of multiple light planes requires too much time for data acquisition.

The latter problem can be mitigated by projecting the light planes in parallel, but this entails determination of the correspondence between each light plane and the image of its intersection with the target. As a trade-off between serial vs parallel projection, the following time-coded light pattern projection method was proposed (Altschuler and co-workers, 1981): each plane among a set of different light planes is turned on or off during each of a sequence of time slots according to a given code, and the resulting images are decoded to determine the correspondence between each plane and its image.

Time-of-Flight Techniques. A time-of-flight range sensor (Fig. 7b) includes a signal transmitter and a signal receiver consisting of a collector of part of the signal reflected by the target and the electronics for measuring the round-trip travel time of the returning signal and its intensity. Two types of signal are practical: ultrasound (such as used by the Polaroid range sensor) and laser light. Ultrasound is much more adversely affected by surface specularity and has a much poorer spatial resolution than laser light; hence, let us consider only laser light.

Time-of-flight laser range sensors use a scanning mirror to direct the transmitted laser beam along pan-and-tilt orientations with equal angular increments in order to obtain a range image consisting of $N \times M$ rangels. Like with triangulation range sensing, by also measuring the intensity of the reflected light, we obtain an intensity image consisting of $N \times M$ pixels in complete registration with the range image. On the other hand, the missing-data problem that is inherent in triangulation range sensing is eliminated by mounting the laser transmitter coaxially with the receiver's reflected light collector.

Three schemes can be distinguished for measuring the length of the transmitter–target–receiver optical path in time-of-flight laser range sensing (Fig. 7b):

1. *Pulse Time Delay.* Using a pulsed laser and measuring the time of flight directly (Johston, 1973); this scheme requires advanced electronics.
2. *AM Phase Shift.* Using an amplitude-modulated laser and measuring the phase shift, which is proportional to the time of flight (Nitzan and co-workers, 1977).
3. *FM Beat.* Using "chirps" of laser waves that are frequency modulated as a linear function of time and measuring the beat frequency, which is proportional to the time of flight (Goodwin, 1985).

Time-of-flight laser range sensors have the following problems:

Specular Reflection. Reflection from a mirrorlike surface may result in no range measurement if the reflected light does not reach the receiver and larger measured range values if the reflected light is subsequently reflected by other surfaces before part of it reaches the receiver.

Slow Measurement. A long integration time is required to reduce the photon noise (and other types of noise) to an acceptable level, especially if the target is dark. For given values of target reflectance, incidence angle, range, and measurement error, the integration time is inversely proportional to the product of the transmitted laser power and the area of the receiver's collector (Nitzan and co-workers, 1977).

Ambiguity in AM Phase Shift. If the phase shift ϕ between the transmitted light and the received light in an amplitude-modulated scheme may exceed 2π, the (true) range r is ambiguous: $r = n\lambda + r(\phi)$, where $n = 0, 1, 2, \ldots$, λ is the wavelength of the modulation frequency, and $r(\phi)$ is the measured range assuming that $0 \le \phi \le 2\pi$; detection of range discontinuities, which are unexpected based on the world knowledge, may be used to determine the value(s) of n.

Processing of light-intensity images and range images are discussed separately in the following two sections.

Intensity Image Processing

Light-intensity image processing in robotic applications is intended primarily to recognize, locate, and inspect objects from an intensity range. Assuming that the image of an object in a given stable state is invariant to its location (position and orientation), processing of three types of intensity image are distinguished below: a complete and isolated outline, a partial or connected (not isolated) outline, and a gray-level image of an object. Processing the intensity images of three-dimensional objects in general is a more difficult task.

Complete and Isolated Outline. The complete outline of an isolated two- or three-dimensional object in a given stable state is the boundary between the top-view image (viewed from infinity) of the object and its background. An object outline can be sensed in two ways:

By Detecting the Black-to-White and White-to-Black Transitions in a Binary Image (Silhouette). The main problems here are obtaining a high black–white contrast when using front illumination (it is relatively easy with back illumination) and adjusting the threshold that converts the image into binary pixels dynamically as the task and lighting conditions vary.

By Detecting the Edges between Dark and Bright Regions in a Gray-Level Image. Edges may be detected by using several methods, such as the Sobel operator (Duda and Hart, 1973) and thinning the resulting "thick edges" or by applying the MIT zero-crossing operator (Hildreth, 1982) which extracts chains of zero-valued pixels from the convolution of the gray levels with the Laplacian of a bivariate Gaussian distribution function. The main problems here are lack of robustness (obtaining false edge points and missing true ones), linking the edge points into expected outlines, and relatively lengthy computation time.

The SRI vision module (Gleason and Agin, 1979) implements the outline interpretation (recognition, location, and inspection) by matching the features of each "blob," or connected region, of the measured outline with those of a model (prototype). The vision module utilizes many global features, such as the outline's area, perimeter, area–perimeter-squared ratio, radius vectors (minimum, maximum, and average) from the centroid to the perimeter, first and second moments, and number and area of holes. Two recognition schemes are distinguished.

Nearest Neighbor Classifier. Selection of the nearest object prototypes in a multifeature space.

Decision Tree. Sequentially divides object prototypes into two groups along tree branches according to the largest gap between the values of the most distinctive feature until reaching a leaf.

The SRI vision module (Fig. 8) consists of three components: one to four TV cameras, a hardware preprocessor, and vision software stored in a microcomputer. Several companies (eg, Machine Intelligence Corp. and Automatix Inc.) have been manufacturing industrial vision systems based on the SRI vision module.

Partial or Connected Outline. Consider two- or three-dimensional objects that may be partially viewed, overlapping without altering their top views substantially or touching. Object outlines are extracted from binary or gray-level images using the same techniques as for an isolated object. The observable outline of any one of these objects may be partial, complete, or connected with another outline; hence, matching cannot be based on global features.

Object recognition and location based on any of these outlines was achieved at SRI (Bulles and Cain, 1982) by focusing on local features, such as small holes, convex corners, and concave corners, which are much smaller than the image of the entire object. As a local feature is detected, the shape and location of its closest local features are extracted and compared with those of a model until there is sufficient match to hypothesize the object's identity and location. The hypothesis is then verified by comparing the observable-object outline, excluding the local features matched previously, with the model outline. For example, local features extracted from the silhouette of four overlapping or touching door hinges were analyzed by this match–verify system and the resulting outlines, shown in Figure 9, were as expected. The system is robust, but further research is needed to explain (by reasoning about object outlines) why portions of some objects are not visible.

Gray-Level Image. Techniques for matching the gray-level image of an object with that of a model are covered in several textbooks (Duda and Hart, 1973; Rosenfeld and

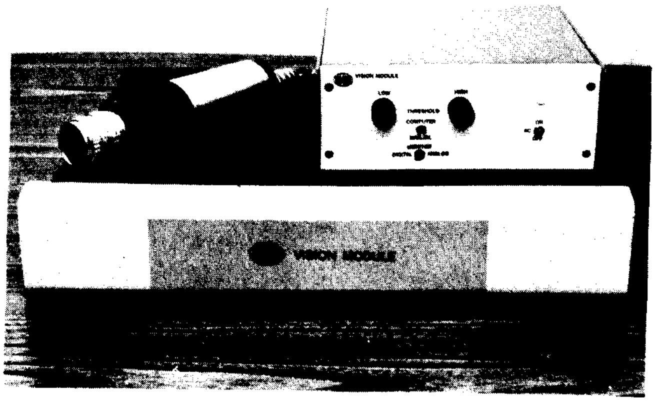

Figure 8. The SRI vision module.

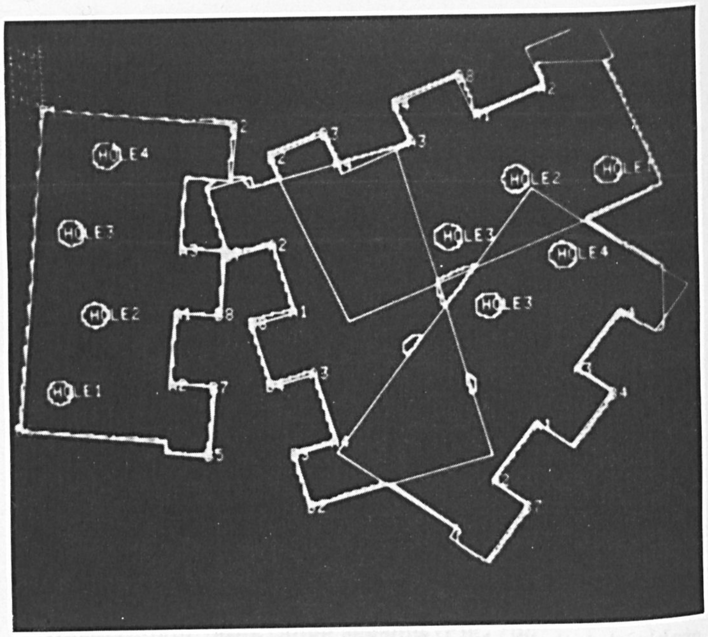

Figure 9. Using local features to locate overlapping or touching door hinges. Courtesy of Bolles, SRI International.

Kak, 1976; Ballard and Brown, 1982; Horn, 1986). These techniques vary over a wide range of complexity, depending on the tasks. One technique, explained previously, is to detect expected edges between dark and bright regions and match the global or local features of these edges with those of an object model. Two other matching techniques are as follows.

Average Gray-Level Matching. The average gray level of a windowed (partial) or the entire image of an indexed object (ie, placed in a known location) is compared with that of a model to determine if there is an acceptable match; simple and fast, this technique is used in commercial visual inspection systems (Miyagawa and co-workers, 1983).

Statistical Gray-Level Matching. A gray-level histogram (a function showing, for each gray-level increment, the number of pixels in the image whose gray-levels lie within that increment) of an image of an indexed object is compared with that of a model to detect missing or severely damaged parts, and a gray-level joint distribution of the object and its model are used to detect missing or misplaced parts (Barnard, 1980).

Range Image Processing

Range image processing is intended primarily to recognize, locate, and inspect three-dimensional objects (see also RANGE DATA ANALYSIS). Basically, such processing is implemented by extracting three-dimensional geometric features and matching them with those of a model. Matching the three-dimensional features of an isolated object whose image is variable is much more difficult than matching the two-dimensional features of an isolated object whose image is invariable. Matching the features of three-dimensional objects that are piled on a tray or jumbled in a bin is even more difficult because each object may have infinite possible orientations and may be partially occluded by other objects. Compared with visual image processing, range image processing has barely scratched the surface; much research is still needed in this area.

Because range data describe the surface of a solid object (not its interior), the object should be represented by its surface (not its volume). Three useful representations of three-dimensional surfaces are considered below: faces, generalized cylinder surfaces, and volumetric.

Faces. Object faces are usually represented mathematically by a number of unbounded planar or curved surfaces (eg, cylindrical, conic, or spherical) that confine the object. These surfaces may intersect along edges that, in turn, may intersect at vertices. Geometric modeling in most computer-aided design (qv) (CAD) systems is based on this representation.

Geometric features (surfaces, edges, and vertices) seen by a range sensor can be extracted from its range image. The challenge here is to extract these features reliably and quickly from range data that may be incomplete and noisy. Previous range image processing (Nitzan and co-workers, 1977; Nitzan, 1972; Duda and co-workers, 1979) has extracted the following features:

Jump edges, the boundaries between occluding and occluded surfaces (as seen by the range sensor), which are characterized by range discontinuities; jump edges constitute the portion of the occluding object outline where no contact is made with any other object.

Convex or concave edges, which are characterized by discontinuity in the range gradient.

Planar surfaces, classified into horizontal, vertical, and slanted ones (an example is shown in Fig. 10).

More recently, Bolles and Horaud (1986) developed a system for determining the location (position and orientation) of each visible part in a bin of jumbled pads of the

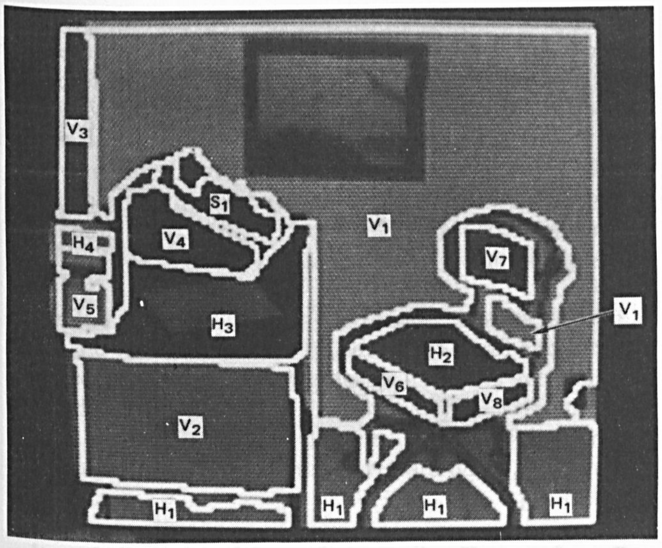

Figure 10. Office scene partitioned into planar surfaces. H = horizontal, V = vertical, S = slanted (Duda and co-workers, 1979).

same type. The system generates a hypothesis about the location of each part by extracting three-dimensional edge features of three types (straight dihedral, circular dihedral, and straight tangential) from range data and matching a sufficient number of distinctive features to those in the part's model. For example, Figure 11a depicts the measured height of seven castings; higher points are brighter. The system analyzes these range data, extracts three-dimensional edge features, and hypothesizes the casting locations, as shown in Figure 11b. Next, the resulting hypothesis of patches is verified by predicting the range values on the object's surface and comparing them to the measured data (Fig. 11c). Furthermore, the system locates the pads on top of the pile so that a robot hand can pick them up.

Generalized Cylinder Surfaces. A generalized cylinder is the volume swept by a two-dimensional region moving along and normal to an axis (straight or curved) in a three-dimensional space.

Surfaces of generalized cylinders were used to represent a manufactured doll (Agin and Binford, 1973) and other complex objects (Nevatia and Binford, 1973). Generalized cylinder surfaces were extracted from range data (obtained by using planar light slices) and matched with their models.

Contour lines are the intersections of a set of known planes with a three-dimensional object of an arbitrary shape. The planes may be parallel and equidistant (as in topographical maps), radial with equal angular increments, and the like. Contour lines may be viewed as an extended representation of a generalized cylinder surface formed by sweeping a region of arbitrary shape along an axis that may branch into separate axes. The intersections of a set of light planes with an arbitrary object, used to measure its range image, constitute the visible portions of its contour lines. As such, contour-line representation is compatible with range measurement based on structured

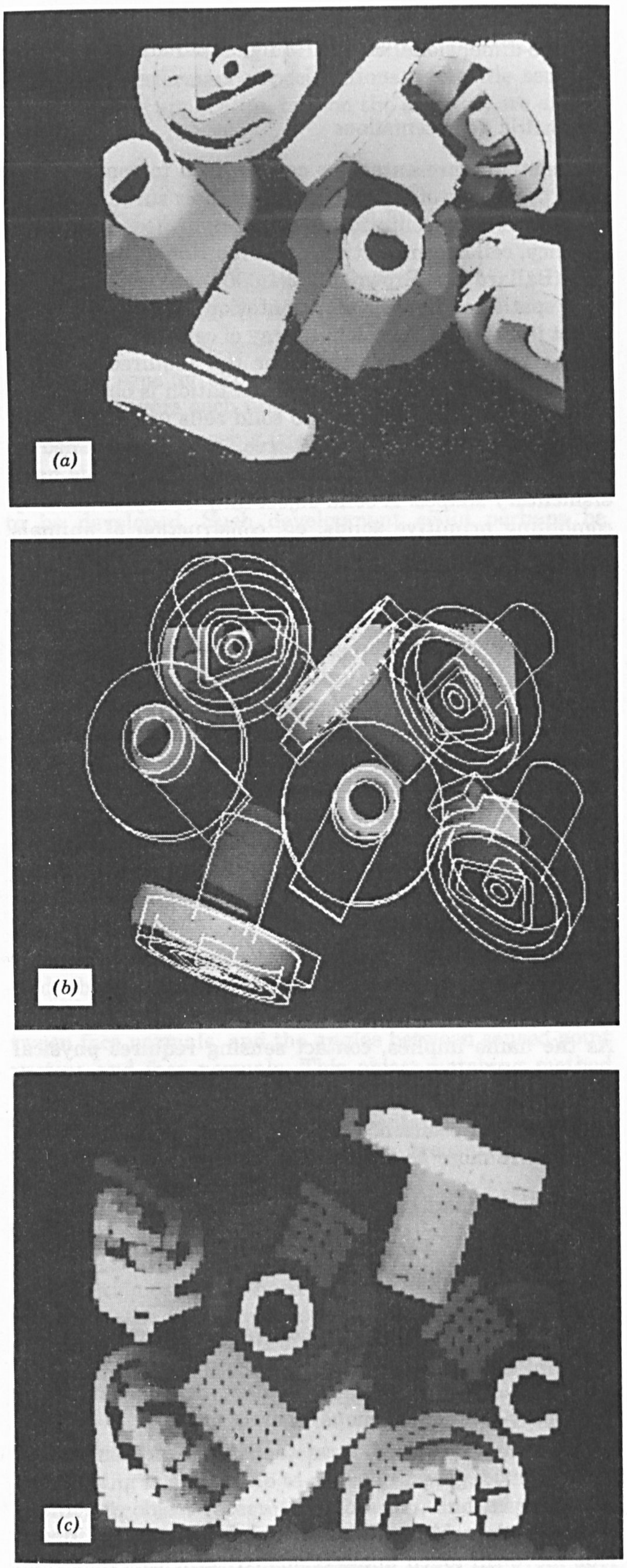

Figure 11. Locating castings in a bin. (*a*) Measured height data (higher points are brighter). (*b*) Hypothesized castings. (*c*) Hypothesis verification. Courtesy of Bolles and Horaud, SRI International.

light and can be used to inspect the shape and dimensions of three-dimensional objects (Mulgaonkar and co-workers, 1985).

Volumetric Representations

Volumetric representations can be used to compute the mass properties of objects as well as their surfaces. Three volumetric representations are distinguished: spatial occupancy, cell decomposition, and constructive solid geometry (Ballard and Brown, 1982).

A spatial occupancy representation (voxels) of a solid object is a three-dimensional array of cells (eg, cubes); the higher the resolution the larger the required memory space. A cell decomposition representation is obtained by subdividing a solid object into solid cells (no holes). One such representation is the "Oct-tree," which is obtained by recursive subdivision of the solid until all the cells have elementary shapes. A solid object may be represented by combining primitive solids, eg, construction of animals using cylinders.

Research Issues

Although impressive progress has been made in edge & local feature detection (qv) and relational optical flow (Hildreth, 1983), and three-dimensional vision, these areas require further research. Other research issues in noncontact sensing include parallel computation, object representation, extraction of intrinsic and relational features from surface images, and strategies for matching these features to object models. Different applications have required different sensor types and different object representations. The question is how much of these differences is economical and how much is inherent.

CONTACT SENSING

As the name implies, contact sensing requires physical contact with an object whose properties are measured. Contact-sensor signals include tactile, force/torque, temperature, and position. Of these, tactile and force/torque sensors are more general.

Tactile Sensors

Classification. Tactile sensors may be classified according to the following criteria:

Resolution. A single element (eg, a microswitch) vs an array of elements.

Dynamic Range. Binary (contact or no contact) vs gray-level (continuous) contact force.

Directionality. Normal vs tangential force measurement.

Applications. A binary microswitch may be used in various move-till-touch applications (Nitzan, 1981), such as reaching a target, preventing collision damage, robot training, object grasping, and dimensional measurement.

Arrays of tactile transducers (binary or gray-scale) mounted on compliant fingers of a robot hand are applicable to object recognition (Hillis, 1982), grasping, locating, inspection, and slip detection. Rather than being detected, a slip may be prevented by increasing the normal force, N, just before the sensed tangential force reaches $\mu_0 N$, where μ_0 is the coefficient of static friction between the object and the tactile transducers. Some tactile sensors may also be used to measure the hardness and thermal properties of materials (Dario and De Rossi, 1985); these properties may supplement three-dimensional features in object recognition.

In a survey conducted by Harmon (1982) the following applications for tactile sensing were identified: bin picking, adaptive grasping, assembly, dimensional inspection, shape detection, temperature measurement, tight-part mating, electronic-component insertion, wire-harness construction, limp-material handling, fruit picking, and cow milking.

Transducers. Tactile transducers may be based on the following technologies (Dario and De Rossi, 1985; Harmon, 1982).

Pressure-Sensitive Resistivity. Simple, inexpensive, and heat resistant but lacking sensitivity.

Semiconductors. Small and sensitive but fragile and sensitive to the environment.

Piezoelectric Transduction. Potentially useful but lacking dc response.

Capacitive Transduction. Potentially useful but too sensitive to external fields.

Optoelectronic Transduction. Highly sensitive detection of light whose intensity varies by force-sensitive mechanical means, which are bulky.

Piezoelectric and Pyroelectric Transduction. Potentially useful for sensing pressure and thermal properties, but sensing them separately is electronically difficult.

A few examples follow.

Hillis (1982) developed a 16 × 16-element tactile sensor consisting of two 16-conductor sheets aligned perpendicularly to each other and separated by a thin elastic medium. The local resistance between each pair of crossing conductors decreases (nonlinearly) as the applied pressure increases. A tactile-pressure image is obtained by applying a voltage to one column conductor at a time and measuring the current flowing in each of the row conductors as they are grounded one at a time. Hillis realized one conductor sheet by an etched flexible printed-circuit board, the other by an anisotropically conductive silicon rubber, and the elastic medium by a nylon woven mesh. Further development of this sensor was terminated due to problems with material manufacturing and robustness (Hollerbach, 1985).

Boi (1984) developed a 6 × 6-element tactile sensor consisting of six conductors etched on an elastic/dielectric layer and six conductors etched normally to the ones above on a flexible printed-circuit board (Fig. 12). The local capacitance between each pair of crossing conductors increases as the applied pressure increases because the

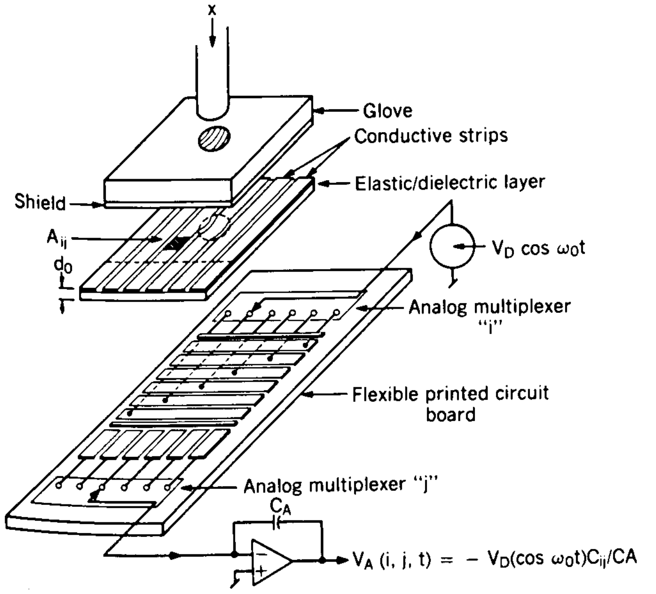

Figure 12. Exploded view of capacitive tactile sensor (Boie, 1984).

distance between them decreases. This sensor scheme appears encouraging and is being pursued further at MIT (Hollerbach, 1985).

A 3 × 6-element tactile sensor combining a transducer (pressure-sensitive conductive elastomer) and a VLSI signal processor has been designed and built by Raibert and Tanner (Raibert, 1982) (Fig. 13). The quality of the transduced data was not good because of the hysteresis and poor mechanical ruggedness of available elastomers and because of the difficulties in placing analog electronics on a chip (Raibert, 1984). Raibert (1984) has attempted to overcome these problems by developing a scheme for measuring local pressure digitally.

A survey of tactile transducers by Dario and De Rossi (1985) includes the following ones: optoelectronic transducers using optical fibers (the Jet Propulsion Laboratory in Pasadena, Calif., and MIT in Cambridge, Mass.) and LED–photo detectors (developed at SRI International and manufactured by Lord Corp.), piezoresistive transducers (Carnegie Mellon University in Pittsburgh, Pittsburgh, Pa., and LAAS in Toulouse, France), piezoelectric–pyroelectric transducers using polyvinylidene fluoride polymers (PVF2) (University of Pisa, Italy).

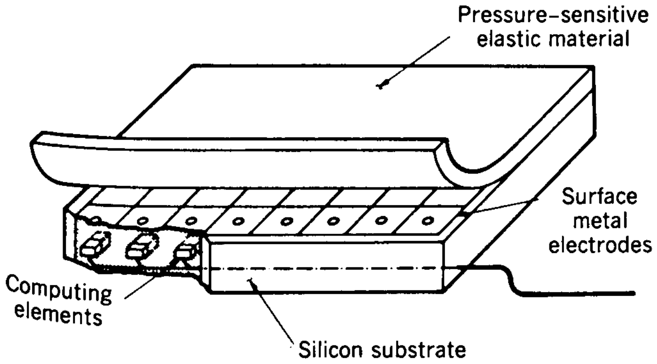

Figure 13. VLSI tactile array sensor (Raibert and Tanner, 1982).

Desired performance specifications for tactile sensing depend on the application but, on the average, are as follows (Harmon, 1982):

Resolution (5 × 10 to 10 × 20 elements with 1–2 mm spacing, matching the human fingertip).
Sensitivity (1–10 g).
Dynamic range (1000:1).
Response Time (1–10 ms).
Nonlinearity is acceptable but hysteresis is not.
Robust sensing skin on a compliant material (flesh) that covers each finger of a robot hand.
Local data processing (resulting in a smart hand).

Tactile sensors that meet all of these specifications are yet to be developed. Such development could perhaps be helped by studying the mechanisms entailed in human sensing. Most attempts to develop humanlike tactile sensors have failed so far, primarily due to material problems.

Tactile Image Processing. A tactile image generated by a set of transducer elements that have been activated by contacting an object may be binary or gray-level. Hence, binary or gray-level vision-processing techniques are applicable directly to tactile image processing (Hillis, 1982).

Grimson and Lozano-Perez (1984) developed a method for recognition and location of an isolated polyhedral three-dimensional object with three degrees of freedom based on a set of surface-point positions and normals measured by tactile sensors. The large number of possible interpretations in matching the measured points with modeled object surfaces is reduced considerably by using constraints on the distances between faces, the angles between face normals, and the angles between sensed-point vectors and face normals. This object-matching method was later extended (Grimson and Lozano-Perez, 1985) to cases in which the position and orientation of planar-surface patches (or linear segments) are measured by a sensor (eg, gray-level visual sensor).

A research issue is how to perform image interpretation that is unique to contact sensing. For example, interpreting tactile data associated with object grasping entails friction (which is hard to model) and grasp location. Another issue is dynamic vs static tactile sensing, how to interpret dynamic tactile data, how the information it provides differs from that of static tactile data, and what the analogy with dynamic and static vision is.

Applicability of Tactile versus Noncontact Sensing. Tactile sensing is applicable to object grasping, but its applicability to object recognition, location, and inspection is questionable because these tasks can be done today much more effectively by visual sensing.

Compared with noncontact (vision and range) sensing of a three-dimensional object, today's tactile sensing has inferior performance specifications, has insufficient techniques for image interpretation, must entail manipulation (eg, by a robot arm and hand), and may cause some object displacement. On the other hand, tactile sensing is direct, requires no illumination, and can provide information about the object grasping, including slip detection or prevention. In addition, noncontact sensing is inferior if the surface of the object is occluded (eg, during object grasping), is either too dark (eg, in deep sea) or too specular (causing transducer blooming or misleading multiple reflections), or is characterized by uniform reflectance despite range variations.

In conclusion, tactile sensing should complement noncontact sensing, not compete with it. Before such sensor integration can be achieved, however, tactile sensing needs to be advanced to a level comparable to that of noncontact sensing.

Force Sensors

A six-axis wrist force sensor measures the three components of force and three components of torque acting between the wrist of a robot arm and its end effector which, in turn, exerts the measured force or torque on an object.

Force transduction is achieved by measuring the deflection of compliant sections as a result of the applied force and torque. Force transducers include piezoelectric material and semiconductor strain gauges. The accuracy of force transducers may be increased by mounting them directly on the end effector fingers rather than on the wrist. Figure 14 shows a strain-gauge wrist sensor developed at SRI (Rosen and co-workers, 1974).

Research Issues

Research issues include improving the performance of tactile transducers (see above specifications), analysis of tactile gray-level images (for object handling, recognition, location, and inspection), friction modeling (for slip detection or prevention), and the applicability of tactile sensing with and without noncontact sensing.

Other research issues are effective discrete and continuous sensing, contact location and friction, applicability of contact sensing, combined force–position control, force control of cooperating arms or fingers, performance prediction, and pattern recognition.

ADAPTIVE CONTROL

Overview

The dynamic behavior and positioning accuracy of a robot arm under fixed control vary with the arm configuration and load. An arm with unknown or time-varying parameters could, in principle, be controlled dynamically using adaptive control (see Robot control systems) by adjusting the parameters according to the position, velocity, and acceleration servo errors. The problem is that the theory

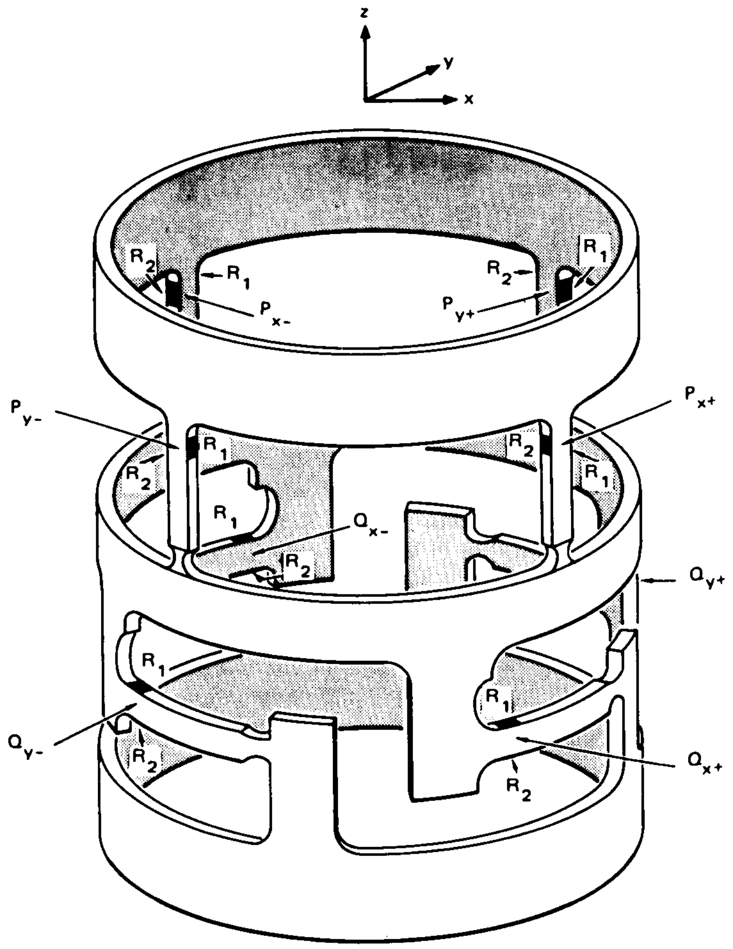

Figure 14. Strain-gauge wrist force sensor.

for adaptive control of nonlinear, stochastic systems is inadequate (Sanderson and Weiss, 1985). Craig (1985) divided the dynamic model into two parts: one with known parameters (eg, link inertia) and the other with a known structure but unknown parameters (eg, friction and load inertia) that may vary in time. He proposed a scheme using an adaptation law that adjusts the estimated values of the unknown parameters in a closed loop until they converge to values resulting in zero servo errors. The scheme is based on nonlinear equations of motion and uses Liapunov (Hahn, 1963) function to guarantee stability but is not fast. Controlling a robot arm dynamically by evaluating its unknown and time-varying parameters in order to minimize the servo errors is a research issue.

Adaptive control of a robot arm may be implemented by using sensors, in particular visual and range sensors, to measure the location (position and orientation) of its end effector relative to a target object, despite measurement delay and noise. Sanderson and Weiss (1983) distinguished between two visual feedback representations: a position-based feedback, whose parameters are relative locations, and an image-based feedback, whose parameters are image features. The latter is inherently faster (because it entails no feature-to-space computation delay) but its feedback is nonlinear. They also distinguished between two types of joint-control structures: look-and-move and visual tracking.

Examples

Adaptive control demonstrations at SRI included the following.

Compressor-Cover Assembly. Using global and local features, a compressor housing and its cover were located by a vision subsystem, picked up by a Unimate arm, and placed on an xy table (Fig. 15). Guided by the vision subsystem, the xy table moved to each of eight positions where an Auto-Place limited-sequence manipulator bolted the cover while the table was free to move slightly to accommodate locational errors. Visual and positional sensing were used to verify this operation (McGhie and Hill, 1978).

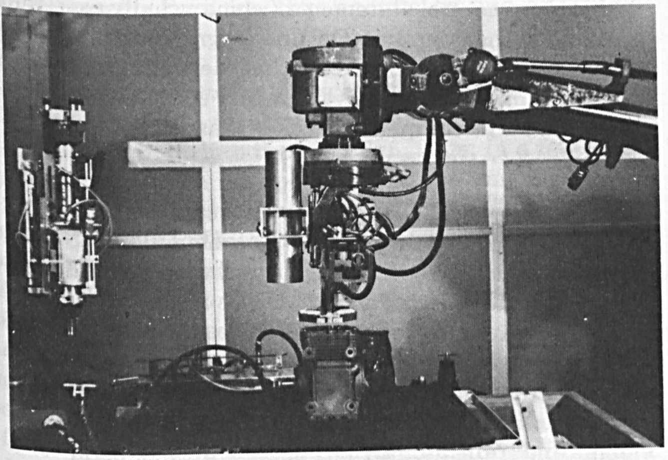

Figure 15. Compressor-cover assembly

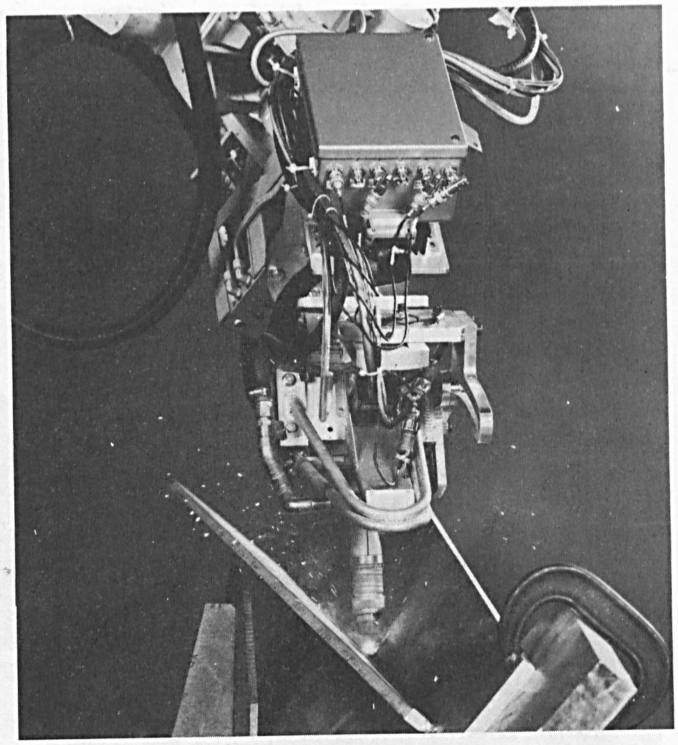

Figure 16. Visually guided arc-welding robot. Courtesy of Kremers and co-workers, SRI International.

Tracking. Visual–range servoing techniques were developed using a projector of a light plane and a TV camera mounted on the end effector of a Unimate arm and applied to simulated spot welding on a moving line and tracking a cornered path in three dimensions (Agin, 1979).

Arc Welding. Using a laser scanner range sensor mounted on its wrist, a Cincinnati-Milacron T3 arm was guided by the vision subsystem in one-pass arc welding of different workpiece joints (Fig. 16) (Kremers and co-workers, 1983).

Modular Printer-Carriage Assembly. A nine-part printer carriage was assembled by a modular-assembly station consisting of two PUMA arms with visual and force sensors, respectively, attached to their wrists, a binary vision module, and general-purpose part feeder and assembly fixture on two tables, respectively (Fig. 17). Binary visual sensing was used to locate plastic rockers and, subsequently, force and click sensing were used to verify that they have been snapped properly into a shaft (Smith and Nitzan, 1983).

Issues

Some issues entailed in adaptive robot control are as follows.

Basic Theory. Developing a general analysis that guarantees stability and high speed in controlling nonlinear, stochastic systems with unknown and varying parameters.

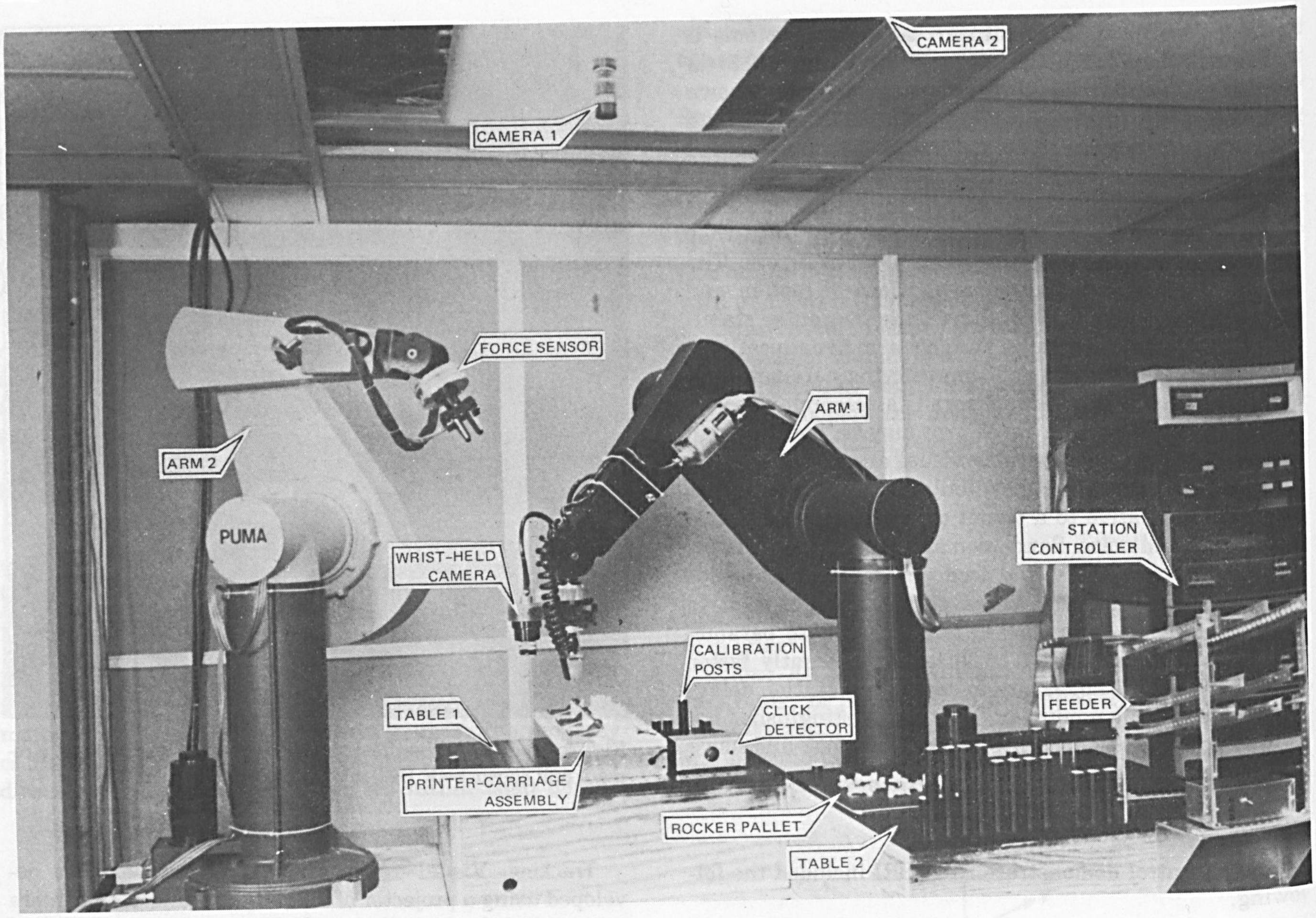

Figure 17. Modular assembly station.

Sensor Integration. Selecting the best sensors for a given task and integrating their outputs into the robot control system.

Branching. Planning cost-effective conditional branches for robot actions, depending on the sensor signals.

Execution Speed. Obtaining and matching high speed sensory data processing, arm motion, and arm control.

ROBOT PROGRAMMING

Training today's industrial robot arm is usually done by a human operator who, using a teach box with push buttons, leads that arm through its task steps and records the location or action of each step. Although popular, this teaching-by-doing method may be wasteful (eg, no useful work during training for batch production), tedious (eg, training for many computable locations), or inadequate (eg, training for sensor-guided task steps). These limitations can be partially overcome by using a computer-based textual robot-programming language. Furthermore, a robot-programming language (or an equivalent computer program) is essential for programming a robot system, working alone or with other machines, that consists of manipulators, sensors, and auxiliary devices and is controlled by a hierarchy of distributed computers.

A robot programming-language is applicable in two programming modes: on-line programming and off-line programming. On-line programming is performed by a programmer who, although sensing and handling real equipment and workpieces, generates a program text that may include manipulator locations (either taught by doing or typed in), robot sensing (to overcome uncertainties), and logic or control statements (which usually constitute the bulk of the program). Off-line programming is similar to on-line programming except that the programmer deals with simulated objects (workpieces, sensors, manipulators, and other equipment) rather than real objects.

Robot-Programming Languages

The main goal of using a robot-programming language is to facilitate the programming of a robot system for a new task or modification of an old one. To achieve this goal, a robot-programming language provides the user with high level programming capabilities. These capabilities are implemented by means of a language processor and a robot controller: the processor accepts and checks the user statements and translates them into commands for the controller, the controller then generates lower level commands for the corresponding device (eg, the trajectories, joint values, and servo commands for the arm joints).

Current Robot-Programming-Language Capabilities. Eight commercially available U.S. robot-programming languages have been compared by Gruver and co-workers (1983).

AL, Stanford University (Mujtaba and co-workers, 1981).

AML, IBM Corp. (Taylor and co-workers, 1982).

HELP, General Electric Co. (1982).

JARS, Jet Propulsion Laboratory (Craig, 1980).

MCL, McDonnell Douglas Corp. (Wood and Fugelso, 1983).

RAIL, Automatix, Inc. (1982).

RPS, SRI International (Parks, 1983).

VAL, Unimation, Inc. (1980, 1981).

The combined features of these robot languages are classified below into general and robotic programming capabilities (see the above references for individual robot-language capabilities).

General Programming Capabilities. General programming capabilities include the following high level language features:

Data Types. Integer, real, character string, label, and aggregate (an ordered set of data types) (Taylor and co-workers, 1982).

Operations. Arithmetic, relational, logical, assignment, etc.

Control Expressions. Block structure (BEGIN_END), branching (GOTO), conditional branching (IF_THEN; IF_THEN_ELSE_), continuing (WHILE_DO_), and looping (DO_UNTIL_).

Subroutines and Functions. Library (compiled) and user-generated (interpreted).

Interactive Support Modules. Text editing, hot editing (run, stop, edit, and continue to run a program), compiling, interpreting, graphic simulation, etc.

Debugging Features. Break points, tracing, and single-stepping.

Robotic Programming Capabilities. Robotic programming capabilities, invoked by declarations or commands, are usually built into the robot language; in a few cases, however, they are incorporated into the language by interfacing with external modules (eg, RPS interfaces with the PUMA VAL (Unimation, Inc.) controller and the SRI and MIC vision modules). These capabilities are classified below according to the nature of the robot functions:

Geometric Data Types. Vector, displacement, rotation, frame, transform, and path of points.

Motion of Arm End Effector. Specified joint(s), interpolated joints (between two points), straight line (to a given destination), straight line via a given point, continuous path (through given points), sawtooth weaving superimposed on a continuous path, specification of speed and acceleration or deceleration, departure and approach, etc.

Vision. Picture taking, binary feature extraction, silhouette-based object recognition and location, adjustment of thresholds and windows, and gray-level feature extraction (eg, histograms).

Servoing with Sensory Feedback. Visual sensing, limit switch, and force/torque sensing.

Multiprocessing. Simultaneous control of multiple arms, sensors, machines, and other devices in a manufacturing cell.

Limitations and Issues. Current robot-programming languages are handicapped by the following limitations and issues (Soroka, 1983; Goldman, 1985).

System Integration. Robot-programming languages, currently capable of controlling a single arm, should be able to concurrently control a flexible workcell comprised of two or more arms, machines, vision modules, other sensors, and auxiliary equipment, and be integrable into the factory CAD–CAM system.

Task-Level Commands. Robot-programming languages, currently including only manipulator-level commands, should also include higher, task-level commands to simplify workcell programming for more complex tasks.

Flexibility and Intelligence. Robot-programming languages should be flexible and intelligent enough to handle unpredictable situations, eg, be able to verify that specified trajectories are collision-free.

Hardware Dependency. Robot-programming-language software is closely tied to specific computer hardware; although achieving higher speed, this dependency hinders computer portability.

Language Standardization. A commercial robot-programming language is proprietary with its vendor and depends heavily on the kinematics of the arm it controls; hence, users of different arms must program them in several languages. This problem could be solved by having a standard, arm-independent language processor that interfaces with each arm-specific controller that servos the arm joints.

Marketing Considerations. Robot-arm marketing appears to be the major incentive for developing a robot programming language. This factor may oppose efforts to standardize robot programming languages unless one robot manufacturer dominates the market.

User–Language Compatibility. The spectrum of users who match the complexity of a given robot-programming language is narrow. Developing robot-programming languages that match a wide spectrum of users, from a factory worker to a computer scientist, is an issue. Along this line, the utility of a robot-programming language that is restrictive when used by a sophisticated programmer who requires programming flexibility may be questioned. An alternative approach is to utilize the general programming capabilities of a high level language (such as C, Pascal, or Ada) and, with an appropriate computer operating system, develop hierarchies of robot-function subroutines that constitute robot-programming capabilities. This approach could reduce the cost and enhance the portability of the robot software as well as improve programming flexibility.

Off-Line Programming

Off-line programming is performed by a programmer who generates a program text that may include simulated manipulator locations, simulated robot sensing, and logic or control statements.

Advantages and Disadvantages. Compared with on-line programming, off-line programming has the following advantages.

No Production Stoppage. Production need not be stopped to make real robotic equipment available for programming. This is a major cost-reduction factor.

Safety. The danger of harm to the programmer, the equipment, and the workpieces during programming is eliminated.

Early Programming. Programming can begin before workpieces, robots, machines, fixtures, and other equipment arrive and, hence, is not susceptible to delays caused by late deliveries.

Product Redesign. Any mistake in the design of a workpiece or a product can be detected and corrected before it is produced.

Sensitivity Verification. The generated program can be verified under exhaustive conditions that can be simulated but not realized experimentally. For the same reason, determining the sensitivity of operation to variation of parameters must be based on simulation.

On the other hand, the disadvantages of off-line programming are described below.

Model Limitations. The accuracy of simulating a robot system depends on the authenticity of its model; complex phenomena that are hard to model precisely will result in simulation errors.

Modeling Cost. Unlike on-line programming, simulation in off-line programming entails modeling cost.

Three-Dimensional Geometric Models. Three types of geometric model of three-dimensional objects are distinguished:

1. *Wire-Frame Models.* Represent objects by their vertices and edges; these models are simple and fast but are hard to visualize (especially if hidden lines are not eliminated) are useless for collision detection, and may be ambiguous.
2. *Surface Models.* Represent objects by their surfaces; the above wire-frame model drawbacks are eliminated by using surface models, but they are slow and provide no information about the mass, inertia, and stability of objects.
3. *Solid Models.* Represent objects by their volumes (including surfaces); most of the above drawbacks are eliminated by using solid models, but they are slow.

Simulation. A graphic simulation system uses a three-dimensional model to generate static or dynamic displays of objects and workcell layout. Such a system enables a programmer to view workcell objects from different directions (using perspective transformations) with variable magnification (eg, zooming), as if the programmer were a "flying eye" observing a real workcell. Using this facility, an off-line programmer interactively writes and debugs the program steps for a given task, observes the animation of the operation, and when satisfied with the results, downloads the program on to the (real) workcell controller and runs it. For example, Figure 18 shows a display (generated on a Silicon Graphics IRIS 2400 system) of a simulated PUMA arm with a two-finger hand mounted on a wrist force sensor, a table, and a cylindrical part on it. A simple task was programmed off-line and executed by a real, calibrated workcell: a vision module (simulated but not shown in Fig. 18) located the part and the arm, guided by this information, picked up the part and moved it to its destination.

CAD–CAM Database. Ultimately, the geometric and nongeometric representations of all the equipment and workpieces in a factory will be stored in a computer, and these databases will be applied to CAD and computer-aided manufacturing (CAM) (see MANUFACTURING, AI IN). Currently, however, a limited amount of such a database

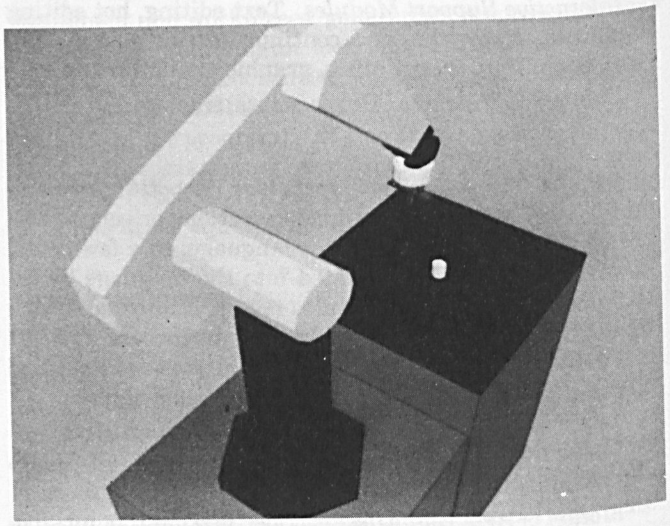

Figure 18. Graphic simulation in off-line programming of a robot system. Courtesy of Smith, SRI International.

exists in certain industries (eg, some aerospace and automotive industries), and its use is limited to static and dynamic graphic display in the design stage and to programming numerically controlled (NC) part machining in the manufacturing stage.

Research Issues. Among the research issues related to off-line programming are the following ones:

Three-Dimensional Object Modeling. Increasing the speed of processing surface and solid models both algorithmically and by hardware, modeling flexible objects, locational uncertainty.

CAD–CAM. Tolerances, path planning and collision avoidance, task-level robot functions acting on workpieces under a range of constraints, near optimal robot selection and workcell-layout design, nongeometric workpiece and equipment representations, standardizing robot tools and types, rules for manufacturing processes (fabrication, handling inspection, and assembly), integration of different CAD–CAM systems.

Simulation. Efficient user interface: robot sensing simulation, robot reach, robot dynamics, static and dynamic process simulation.

MANUFACTURING PROCESS PLANNING

Manual robot-system programming entails a major effort even if facilitated by a robot-programming language or done off-line. Graphic simulation may help specify robot locations, but not logic or control statements, which usually constitute most of the program. Developing a task-oriented programming language will mitigate this problem, but a long-term solution is automatic robot-system planning and programming.

Consider a programmable assembly workcell that is controlled by a hierarchy of distributed computers and includes manipulators, sensors, part feeders, and auxiliary devices. If instructed to assemble a particular product, the workcell must be provided with a workable plan for the assembly process. Consider how such a plan can be generated cost-effectively in a future factory equipped with robotic CAD–CAM systems.

The output of the CAD system for a given product will include two major components: the design of the product parts and their attachment sequence. This information is used to plan the robotic assembly operations, each of which includes actions and verifications of equipment, parts, and the actions themselves. The CAD output can be generated interactively by a human designer without much difficulty. In contrast, planning the details of the actions and, especially, the verification of each assembly operation is very tedious and prone to human errors. Hence, there is a need to automate planning of robotic assembly processes (and, for similar reasons, other manufacturing processes).

Process planning (qv) depends on the state of the process world, which describes the equipment, workpieces, and actions vs time, and is part of the process knowledge base. Few automatic planners using AI techniques have been developed to date, and these planners [eg, DEVISER (Vere, 1981)] are based on the assumption that the state of the process world is known exactly at the planning time. In an unstructured world, where this assumption is invalid, the state of the process world should be determined by robot sensors.

Development of a sensor-based planner and execution monitor using AI techniques was begun by Cheeseman at SRI (Nitzan, 1983). The planner will include a knowledge base ("expert system"), which consists of a world model (effectors, sensors, and other devices), process rules, and sensor-selection rules, as well as a plan generator which plans the action and checking steps for a given robot process. The generated plan will be sent to the execution controller, which will command and coordinate the process execution. As the process world changes, its state will be updated by the robot sensors; this information is required to execute and monitor the plan actions. The highlights of the sensor-based planner are described below.

The plan will call for the use of sensors at run time to fill in unknown information of two types: enabling execution of actions in the plan, eg, visually servoing an arm, and monitoring the plan execution, ie, verifying its actions. The planner will select the best sensor(s) for a given task on the basis of rules and estimated locational errors of objects, effectors, and sensors. Conditional planning will provide alternative branches for all possible sensor outputs, followed by immediate rejoining of the branches to simplify the plan (ie, prevent branch bushiness).

Multiple sensors will be used to combine sensory information if a single sensor is not sufficient. Fixed sensors (eg, mounted on the ceiling) will be distinguished from manipulated ones (eg, mounted on robot end effectors) because the latter entail actions, in addition to information gathering, which must be planned. Error detection and correction will be implemented. If a failure is detected, the process will be halted and, if possible, replanned locally to minimize production down time.

Research issues in planning include the following: a formal theory and representation of knowledge (eg, geometry, uncertainty, temporal relationships, and condition monitoring); planning collision-free paths, fine-motion sensing, part mating, error detection and correction, and multiple robot cooperation; real-time planning and replanning; planning for multiple agents (distributed AI); and planner learning. These research issues are extremely difficult to solve. Despite the importance of automatic planning, its implementation will take many years. Eventually, however, automatic robot-system planning will be an essential component of the factory.

BIBLIOGRAPHY

G. J. Agin, "Real Time Control of a Robot with a Mobile Camera," in *Proceedings of the Ninth International Symposium and Exposition on Industrial Robots,* Washington, D.C., Mar. 1979, pp. 233–246.

G. J. Agin and T. O. Binford. "Computer Description of Curved

Objects," in *Proceedings of the Third IJCAI,* Stanford, Calif., Morgan-Kaufmann, San Mateo, Calif., 1973, pp. 629–640.

M. D. Altschuler and co-workers, "The Numerical Stereo Camera," in *Proceedings of the Society for Photo-Optical Instrumentation Engineers Conference on 3-D Machine Perception,* Vol. 283, SPIE, Bellingham, Wash., 1981, pp. 15–24.

Automatix, *RAIL Software Reference Manual* (*ROBOVISION and CYBERVISION*), Rev. 3.0, MN-RB-07, Automatix, Burlington, Mass., Jan. 1982.

R. U. Ayres and S. M. Miller, *Robotics: Applications and Social Implications,* Ballinger Publishing Co., Cambridge, Mass., 1983.

D. H. Ballard and C. M. Brown, *Computer Vision,* Prentice-Hall., Inc., Englewood Cliffs, N.J., 1982.

S. Barnard, "Automated Inspection Using Gray-Scale Statistics," in *Proceedings of the First National Conference on Artificial Intelligence,* Stanford, Calif., AAAI, Menlo Park, Calif., 1980, pp. 49–52.

T. G. Bartholet, "The First 'Functionoid' Developed by Odetics, Inc.," in *Proceedings of the IEEE International Conference on Advanced Robotics,* Tokyo, Japan, Sept. 12–13, 1983, pp. 293–298.

R. A. Boie, "Capacitive Impedance Readout Tactile Image Sensor," in *Proceedings of the IEEE International Conference on Robotics,* Atlanta, Ga., Mar. 13–15, 1984, pp. 370–378.

R. C. Bolles and R. A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus-Method," *Int. J. Robot. Res.* 1(3), 57–82 (Fall 1982).

R. C. Bolles and P. Horaud, "3DPO: A Three-Dimensional Part Orientation System," *Int. J. Robot. Res.* **5**(3) (1986).

J. J. Craig, *JARS: JPL Autonomous Robot System,* Jet Propulsion Laboratory, Pasadena, Calif., 1980.

J. J. Craig, "An Adaptive Algorithm for the Computed Torque Method of Manipulator Control," in D. Nitzan and R. C. Bolles, eds., *Workshop on Intelligent Robots: Achievements and Issues,* SRI International, Menlo Park, Calif., 1985, pp. 171–184.

P. Dario and D. De Rossi, "Tactile Sensors and the Gripping Challenge," *Proc. IEEE Spectr.* **22**(8), 46–52 (Aug. 1985).

J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *J. Appl. Mechan.* (*ASME*), 215–221 (June 1955).

S. Drake, *Using Compliance in Lieu of Sensory Feedback for Automatic Assembly,* Report T-657, Charles Stark Draper Laboratory, Cambridge, Mass., Sept. 1977.

R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis,* John Wiley & Sons, Inc., New York, 1973.

R. O. Duda, D. Nitzan, and P. Barret, "Use of Range and Reflectance Data to Find Planar Surface Regions," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-1**(3), 259–271 (July 1979),

J. F. Engelberger, *Robotics in Practice,* Kogan Page, London, 1980.

R. Featherstone, "The Calculation of Robot Dynamics Using Articulated-Body Inertias," *Int. J. Robot. Res.* **2**(1), 13–30 (Spring 1983).

General Electric, *Allegro Operator's Manual, A12 Assembly Robot,* General Electric Co., Bridgeport, Conn., 1982.

G. J. Gleason and G. J. Agin, "A Modular Vision System for Sensor-Controlled Manipulation and Inspection," in *Proceedings of the Ninth International Symposium and Exposition on Industrial Robots,* Washington, D.C., 1979, pp. 57–70.

R. Goldman, "Programming Languages for Robots," in D. Nitzan and R. C. Bolles, eds., *Workshop on Intelligent Robots: Achievements and Issues,* SRI International, Menlo Park, Calif., 1985, pp. 231–236.

F. E. Goodwin, *Coherent Laser Radar 3-D Vision Sensor,* SME Technical Paper MS85-1005, Society of Manufacturing Engineers, Dearborn, Mich., 1985.

W. E. L. Grimson and T. Lozano-Perez, "Model-Based Recognition and Location from Tactile Data," in *Proceedings of the 1984 IEEE International Conference on Robotics,* Atlanta, GA, 1984, pp. 248–255.

W. E. L. Grimson and T. Lozano-Perez, "Recognition and Location of Overlapping Parts from Sparse Data in Two and Three Dimensions," in *Proceedings of the IEEE International Conference on Robotics and Automation,* St. Louis, Mo., Mar. 25–28, 1985, pp. 61–66.

W. A. Gruver, B. I. Soroka, J. J. Craig, and T. L. Turner, "Evaluation of Commercially Available Robot Programming Languages," *Proceedings of the Thirteenth International Symposium on Industrial Robots and Robots,* Vol. 7, Chicago, Ill., Apr. 18–21, 1983, pp. 12.58–12.68.

W. Hahn, *Theory and Application of Liapunov's Direct Method,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

L. D. Harmon, "Automated Tactile Sensing," *Int. J. Robot. Res.* 1(2), 3–32 (Summer 1982).

E. C. Hildreth, "Edge Detection for Computer Vision System," *Mechan. Eng.,* 48–53 (Aug. 1982).

E. C. Hildreth, *The Measurement of Visual Motion,* MIT Press, Cambridge, Mass., 1983.

W. D. Hillis, "A High-Resolution Image Touch Sensor," *Int. J. Robot. Res.* 1(2), 33–44 (Summer 1982).

J. M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Trans. System Man Cybernetics* 10(11), 730–736 (1980).

J. M. Hollerbach, "Tactile Sensors and Interpretation of Contact Features," in D. Nitzan and R. C. Bolles, eds., *Workshop on Intelligent Robots: Achievements and Issues,* SRI International, Menlo Park, Calif., 1985, pp. 143–152.

B. K. P. Horn, "Shape from Shading," in P. H. Winston, ed., *The Psychology of Computer Vision,* McGraw-Hill Book Co., Inc., New York, 1975.

B. K. P. Horn, *Robot Vision,* MIT Press, Cambridge, Mass., 1986.

T. Iwamoto, H. Yamamoto, and K. Honma, "Transformable Crawler Mechanism with Adaptability to Terrain Variations," in Bartholet, 1983, pp. 285–291.

J. Jablonowski and J. W. Posey, "Robotics Terminology," in S. Y. Nof, ed., *Handbook of Industrial Robotics,* John Wiley & Sons, Inc., New York, pp. 1271–1303, 1985.

R. A. Jarvis, "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-2**, 122–139 (Mar. 1983).

A. R. Johnston, *Infrared Laser Rangerfinder,* Report NPO-13460, Jet Propulsion Laboratory, Pasadena, Calif., Aug. 1973.

T. R. Kane and D. A. Levinson, "The Use of Kane's Dynamical Equations in Robotics," *Int. J. Robot. Res.* **2**(3), 3–21 (Fall 1983).

J. H. Kremers and co-workers, "Development of a Machine-Vision Based Robotic Arc-Welding System," in *Proceedings of the Thirteenth International Symposium on Industrial Robots and Robots,* Vol. 7, Chicago, 1983, pp. 14.19–14.33.

J. Y. S. Luh and co-workers, "On-Line Computational Scheme for Mechanical Manipulators," *Trans. ASME, J. Dyn. Sys. Meas. Contr.* **102**(2), 69–76 (1980).

R. B. McGhee, "Vehicular Legged Locomotion," in G. N. Saridis, ed., *Advances in Automation and Robotics,* JAI, Greenwich, Conn., 1983.

D. F. McGhie and J. W. Hill, "Vision Controlled MS78-685 Subassembly Station," in *SME Robots III Conference,* Technical Paper MS78-685, Chicago, Ill., Nov. 7–9, 1978.

M. Miyagawa and co-workers, "Flexible Vision System "Multi-Window" and This Application," in *Proceedings of the IEEE International Conference on Advanced Robotics,* Tokyo, Japan, 1983, pp. 171–178.

S. Mujtaba and R. Goldman, *AL User's Manual,* 3rd ed., Report No. STAN-CS-81-889, Stanford University, Stanford, Calif., Dec. 1981.

P. G. Mulgaonkar, D. Nitzan, C. K. Cowan, and A. Bavarsky, Inspection of Three-Dimensional Objects," *Proceedings of the Twelfth NSF Conference on Production Research and Technology,* University of Wisconsin, Madison, Wis., May 1985, pp. 431–435.

J. L. Mundy, "The Limits of Accuracy for Range Sensing," in D. Nitzan and R. C. Bolles, eds., *Workshop on Intelligent Robots: Achievements and Issues,* SRI International, Menlo Park, Calif., 1985, pp. 109–133.

D. Nitzan, *Scene Analysis Using Range Data,* AI Center Technical Note 69, Stanford Research Institute, Menlo Park, Calif., Aug. 1972.

D. Nitzan, "Assessment of Robotic Sensors," in *Proceedings of the First International Conference on Robot Vision and Sensory Controls,* Stratford-Upon-Avon, UK, Apr. 1981, pp. 1–11.

D. Nitzan, C. Bavrouil, P. C. Cheeseman, and R. C. Smith, "Use of Sensors in Robot Systems," in *Proceedings of the IEEE International Conference on Advanced Robotics,* Tokyo, Japan, 1983, pp. 123–132.

D. Nitzan, A. Brain, and R. Duda, "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis," *Proc. IEEE* **65**(2), 206–220 (Feb. 1977).

R. Nevatia and T. O. Binford, "Structured Descriptions of Complex Objects," in *Proceedings of the Third International Joint Conference on Artificial Intelligence,* Morgan-Kaufmann, San Mateo, Calif., 1973, pp. 641–647.

D. Nitzan, R. C. Bolles, J. H. Kremers, and P. G. Mulgaonkar, "3-D Vision for Robot Applications," in *Proceedings of the NATO Workshop on Knowledge Engineering for Robotic Applications,* Maratea, Italy, May 12–16, 1986.

D. Nitzan and C. A. Rosen, "Programmable Industrial Automation," *IEEE Trans. Comput.* **25**(10), 1259–1270 (Dec. 1976).

T. Okada and S. Tsuchiya, "On a Versatile Finger System," in *Proceedings of the Seventh International Symposium on Industrial Robots,* Tokyo, Japan, Oct. 19–21, 1977, pp. 345–352.

W. T. Parks, "The SRI Robot Programming System (RPS)," in *Proceedings of the Thirteenth International Symposium on Industrial Robots and Robots,* Vol. 7, Chicago, 1983, pp. 12.21–12.41.

R. P. Paul, "Manipulator Path Control," in *Proceedings of the International Conference on Cybernetics and Society,* San Francisco, Calif., Sept. 1975, pp. 147–152.

R. P. Paul, *Robot Manipulation: Mathematics, Programming, and Control,* MIT Press, Cambridge, Mass., 1981.

M. H. Raibert, "An All Digital VLSI Tactile Array Sensor," in *Proceedings of the IEEE International Conference on Robotics,* Atlanta, Ga., 1984, pp. 314–319.

M. H. Raibert and I. E. Sutherland, "Machines That Walk," *Sci. Am.* **248**(2), 44–53 (Jan. 1983).

M. H. Raibert and J. E. Tanner, "Design and Implementation of a VLSI Tactile Sensing Computer," *Int. J. Robot. Res.* 1(3), 3–18 (Fall 1982).

B. Raphael, *The Thinking Computer: Mind Insider Matter,* W. H. Freeman, San Francisco, Calif., 1976.

C. A. Rosen and co-workers, *Exploratory Research in Advanced Automation,* 2nd report, NSF Grant GI-38100X1, Stanford Research Institute, Menlo Park, Calif., Aug. 1974.

C. A. Rosen and co-workers, *Machine Intelligence Research Applied to Industrial Automation,* 6th report, NSF Grant APR75-13074, Stanford Research Institute, Menlo Park, Calif., Nov. 1976.

A. Rosenfeld and A. C. Kak, *Digital Picture Processing,* Academic Press, Inc., New York, 1976.

K. Sadamoto, *Robots in the Japanese Economy,* Survey Japan, Tokyo, 1981.

J. K. Salisbury and J. J. Craig, "Articulated Hands: Force Control and Kinematic Issues," *Int. J. Robot. Res.* 1(1), 4–17 (Spring 1982).

A. C. Sanderson and L. E. Weiss, "Adaptive Visual Servo Control of Robots," in A. Pugh, ed., *Robot Vision,* IFS Publications, UK, 1983.

A. C. Sanderson and L. E. Weiss, "Adaptive Control of Sensor-Based Robotic Systems," in D. Nitzan and R. C. Bolles, eds., *Workshop on Intelligent Robots: Achievements and Issues,* SRI International, Menlo Park, Calif., 1985, pp. 185–205.

W. M. Silver, "On the Equivalence of Lagrangian and Newton–Euler Dynamics for Manipulators," *Int. J. Robot. Res.* 1(2), 60–70 (Summer 1982).

R. C. Smith and D. Nitzan, "A Modular Programmable Assembly Station," in *Proceedings of the Thirteenth International Symposium on Industrial Robots and Robots,* Vol. 7, Chicago, 1983, pp. 5.53–5.75.

B. I. Soroka, "What Can't Robot Languages Do?" in *Proceedings of the Thirteenth International Symposium on Industrial Robots and Robots,* Vol. 7, Chicago, 1983, pp. 12.1–12.8.

R. H. Taylor, "Planning and Execution of Straight Line Manipulator Trajectories," *IBM J. Res. Dev.* **23**(4), 253–264 (July 1979).

R. H. Taylor, P. D. Summers, and J. M. Meyer, "AML: A Manufacturing Language," *Int. J. Robot. Res.* 1(3), 19–41 (Fall 1982).

Y. Umetani and K. Yonemoto, "Japan Robotics Research for the Next Generation," in *Proceedings of the IEEE International Conference on Advanced Robotics,* Tokyo, 1983, pp. 3–20.

Unimation, *User's Guide to VAL,* Version 12, No. 398-H2A, Unimation Inc., Danbury, Conn., June 1980.

Unimation, *VAL Univision Supplement,* Version 13 (VSN), 2nd ed., Unimation Inc., Danbury, Conn., July 1981.

S. A. Vere, *Planning in Time: Windows and Durations for Activities and Goals,* Jet Propulsion Laboratory Report, California Institute of Technology, Pasadena, Calif., Nov. 1981.

B. O. Wood and M. A. Fugelso, "MCL, The Manufacturing Control Language," in *Proceedings of the Thirteenth International Symposium on Industrial Robots and Robots,* Vol. 7, Chicago, 1983, pp. 12.84–12.96.

General References

Books

J. K. Aggarwal, R. O. Duda, and A. Rosenfeld, eds., *Computer Methods in Image Analysis,* IEEE Press, New York, 1977.

I. Aleksander, *Artificial Vision for Robots,* Chapman & Hall, New York, 1984.

I. Aleksander, ed., *The World Yearbook of Robotics Research and Development,* Kogan Page, London, 1985.

B. G. Batchelor and co-workers, eds., *Automated Visual Inspection,* IFS Publications, UK, 1985.

G. Beni and S. Hackwood, eds., *Recent Advances in Robotics,* John Wiley & Sons, Inc., New York, 1985.

J. M. Brady, ed., *Computer Vision,* North-Holland, Amsterdam, The Netherlands, 1981.

J. M. Brady and co-workers, eds., *Robot Motion: Planning and Control,* MIT Press, Cambridge, Mass., 1982.

J. M. Brady and R. Paul, *Robotics Research: The First International Symposium,* MIT Press, Cambridge, Mass., 1984.

P. Coiffet, *Robot Technology,* 7 vols., Prentice-Hall, Inc., Englewood Cliffs, N.J. 1981.

J. J. Craig, *Introduction to Robotics: Mechanics and Control,* Addison-Wesley Publishing Co., Inc., Reading, Mass., 1986.

A. J. Critchlow, *Introduction to Robotics,* Macmillan, New York, 1985.

G. G. Dodd and L. Rossol, eds., *Computer Vision and Sensor-Based Robots,* Plenum Press, New York, 1979.

R. C. Dorf, *Robotics and Automated Manufacturing,* Reston Publishing Co., Reston, Va., 1983.

W. B. Gevartner, *Intelligent Machines—An Introductory Perspective of Artificial Intelligence and Robotics,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1985.

M. P. Groover and E. W. Zimmers, Jr., *CAD/CAM—Computer-Aided Design and Manufacturing,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1984.

A. R. Hansen and E. M. Riseman, *Computer Vision Systems,* Academic Press, Inc., New York, 1978.

W. B. Heginbotham, ed., *Programmable Assembly,* IFS Publications, UK, 1984.

V. D. Hunt, *Smart Robots,* Chapman & Hall, New York, 1985.

C. G. S. Lee, R. C. Gonzales, and K. S. Fu, *Tutorial on Robotics,* 2nd ed., Computer Society Press, Silver Springs, Md., 1986.

M. D. Levine, *Vision in Man and Machine,* McGraw-Hill Book Co., Inc., New York, 1985.

D. Marr, *Vision,* W. H. Freeman, San Francisco, 1982.

M. E. Mortenson, *Geometric Modeling,* John Wiley & Sons, Inc., New York, 1985.

R. Nevatia, *Machine Perception,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1982.

W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics,* McGraw-Hill Book Co., Inc., New York, 1979.

S. Y. Nof, ed., *Handbook of Industrial Robots,* John Wiley & Sons, Inc., New York, 1985.

Y. C. Pao, *Elements of Computer-Aided Design and Manufacturing,* John Wiley & Sons, Inc., New York, 1984.

T. Pavlidis, *Structural Pattern Recognition,* Springer-Verlag, New York, 1977.

W. Pratt, *Digital Image Processing,* John Wiley & Sons, Inc., New York, 1978.

A. Pugh, *Robot Vision,* IFS Publications, UK, 1983.

A. Pugh, *Robot Sensors, Vol. 1-Vision; Vol. 2-Tactile and Non-Vision,* IFS Publications, UK, 1986.

P. Ranky, *The Design and Operation of Flexible Manufacturing Systems,* IFS Publications, UK, 1983.

W. G. Rohm, "A Remote Promise," *Infosystems,* 52–56 (Sept. 1986).

A. Rosenfeld and A. C. Kak, *Digital Picture Processing,* Academic Press, New York, 1976.

W. E. Snyder, *Industrial Robots: Computer Interfacing and Control,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1985.

D. J. Todd, *Walking Machines—An Introduction to Legged Robots,* Chapman & Hall, New York, 1985.

J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles,* Addison-Wesley Publishing Co., Inc., Reading, Mass., 1974.

M. W. Thring, *Robots and Telechirs,* Halstead Press, a Division of John Wiley & Sons, Inc., New York, 1983.

Conference Proceedings

Proceedings of the International Symposium on Industrial Robots (ISIR), nos. 1–15, 1970–1985, Robotics International of the Society of Manufacturing Engineers, Dearborn, Mich.

Proceedings of the Annual Robots Conferences, Nos. 1–10, 1976–1986, Robotics International of the Society of Manufacturing Engineers, Dearborn, Mich.

Proceedings of the Annual IEEE International Conference on Robotics and Automation, 1984, 1985, 1986, IEEE Computer Society Press, Washington, D.C.

Proceedings of the First International Symposium on Robotics Research, J. M. Brady and R. Paul, eds., MIT Press, Cambridge, Mass., 1984.

Proceedings of the Second International Symposium on Robotics Research, H. Hanafusa and H. Inoue, eds., Kyoto, Japan, MIT Press, Cambridge, Mass., 1985.

Proceedings of the Annual IEEE Conferences on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Washington, D.C.

Proceedings of the First World Conference on Robotics Research, Bethlehem, Pa., Robotics International of SME, Dearborn, Mich., 1984.

Proceedings of the Second World Conference on Robotics Research, Scottsdale, Ariz., Robotics International of SME, Dearborn, Mich., Aug. 1986.

Digital Image Proceedings and Analysis, Vols. 1–2, R. Chellappa and A. W. Sawchuck, eds., IEEE Computer Society Press, Washington, D.C., 1985.

Journals

IEEE Journal of Robotics and Automation.

International Journal of Robotics Research.

International Journal of Robotics and Computer Integrated Manufacturing.

Robotics Today.

Robotics World.

The Industrial Robot.

Sensor Review.

International Journal of Advanced Manufacturing Technology.

International Journal of Robotics.

Journal of Robotic Systems.

D. Nitzan
SRI International

ROBOTS, LEGGED

Legged robots offer the possibility of vehicular travel where only animals can now go. They also offer a tool for the scientific study of legged locomotion in people and animals. This article surveys research on legged robots, starting with work in the nineteenth century and proceeding to 1990. Attention is given to the distinction between statically stable and actively balanced machines, and rough-terrain locomotion is discussed.

WHY STUDY LEGGED MACHINES?

Aside from the sheer thrill of creating machines that actually run and walk, there are two serious reasons for exploring legged robots. One reason is mobility: there is a need for vehicles that can travel on difficult terrain, where existing vehicles cannot go. Wheels excel on prepared surfaces such as rails and roads, but perform poorly where the terrain is soft or uneven. Only about half the earth's landmass is accessible to existing wheeled and tracked vehicles, whereas a much greater area can be reached by animals on foot. It should be possible to build legged vehicles that can go to the places that only animals can now reach. Such vehicles may be useful in industrial, agricultural, military, and space applications.

One reason legs provide better mobility than wheels on rough terrain is that they can use isolated footholds that optimize support and traction, whereas a wheel requires a continuous path of support. As a consequence, a legged system is free to choose among the best footholds in the reachable terrain wheres a wheel is forced to negotiate the worst terrain along the chosen path. A ladder illustrates this point: rungs provide footholds that enable legged systems to climb, but the spaces between the rungs would prevent a wheeled system from making progress. Another advantage of legs is that they provide an active suspension that decouples the path of the body from the paths of the feet. The payload is free to travel smoothly despite pronounced variations in the terrain. A legged system can also step or leap over obstacles. The performance of legged vehicles can, to a great extent, be independent of the detailed roughness of the ground.

A second reason for exploring legged machines is to understand how humans and animals use their legs for locomotion. People use a diverse and rich set of motions to carry, swing, toss, glide, and otherwise propel their bodies through space, maintaining orientation, balance, and speed as they go. Animals also demonstrate great mobility and agility. They use their legs to move quickly and reliably through forest, swamp, marsh, and jungle and from tree to tree. They move with great speed and efficiency. These systems, people and animals, are impressive from a mechanical engineering, sensory-motor integration, and computational point of view.

Despite skill in using limbs for locomotion, humans are still at a primitive stage in understanding the principles that underlie walking and running. What control mechanisms do animals use? The development of legged machines can lead to new ideas about animal locomotion. To the extent that an animal and a machine perform similar locomotion tasks, their control systems and mechanical structures must solve similar problems. Results in biology have already helped with the progress of legged robots. Perhaps results in robotic legged locomotion can provide insight into the principles and control algorithms underlying animal locomotion.

HISTORY OF LEGGED MACHINES

The scientific study of legged locomotion began just over a century ago when Leland Stanford, then governor of California, commissioned Eadward Muybridge to find out whether or not a trotting horse left the ground with all four feet at the same time. Stanford had wagered that it never did. After Muybridge proved him wrong with a set of stop-motion photographs that appeared in *Scientific American* in 1878, Muybridge went on to document the walking and running behavior of more than 40 mammals, including humans (Muybridge, 1955, 1957). Even after 100 years, his photographic data are of considerable value and beauty, and survive as a landmark in locomotion research.

The study of machines that walk also had its origin in Muybridge's time. An early walking model appeared in about 1870 (Lucas, 1894). It used a linkage to move the body along a straight horizontal path while the feet moved up and down to exchange support during stepping. The linkage was based on a design by Chebeyshev, the famous Russian mathematician, made some years earlier. During the 80 or 90 years that followed, workers viewed the task of building walking machines as the task of designing linkages that would generate stepping patterns when driven by a source of power. Many designs were proposed (U.S. Pat., 1893, 1926, 1928, 1928, 1947, 1949, 1958, 1959; Shigley, 1957; Morrison, 1968) (Fig. 1). The performance of such machines was limited by their fixed patterns of motion, since they could not adjust to variations in the terrain. By the late 1950s it had become clear that linkages providing fixed motion would not do the trick and that useful walking machines would need a more versatile form of control (Liston, 1970).

A second approach to providing control for legged locomotion was to harness a human. Mosher used this approach in a four-legged walking truck at General Electric in the mid-1960s (Liston and Mosher, 1968). The project was part of a decade-long campaign to build advanced teleoperators capable of providing better dexterity through high fidelity force feedback. The walking machine Mosher built stood 11 ft tall, weighed 3,000 lb, and was powered hydraulically (Fig. 2). A human driver controlled the motion. Each of the driver's limbs was connected to a handle or pedal that controlled one of the truck's four legs. Whenever the driver caused a truck leg to push against an obstacle, force feedback would cause the handle or pedal to push back on the human, letting the driver feel the obstacle as though it were his or her own arm or leg doing the pushing. After about 20 hours of training, Mosher was able to handle the machine with surprising agility. Films of the machine operating under his control show it ambling along at about 5 mph, climbing a stack of railroad ties, pushing a foundered jeep out of the mud, and maneuvering a large drum onto some hooks. Despite its dependence on a well-trained human for control, the GE Walking Truck was a milestone in legged technology.

A third approach to controlling legged locomotion became feasible in the 1970s: using a digital computer. McGhee (1983) and co-workers at the Ohio State University were the first to do so. In 1977, they built an insectlike hexapod that would walk with a number of gaits, turn, walk sideways, and negotiate simple obstacles. The computer's primary task was to solve kinematic equations in

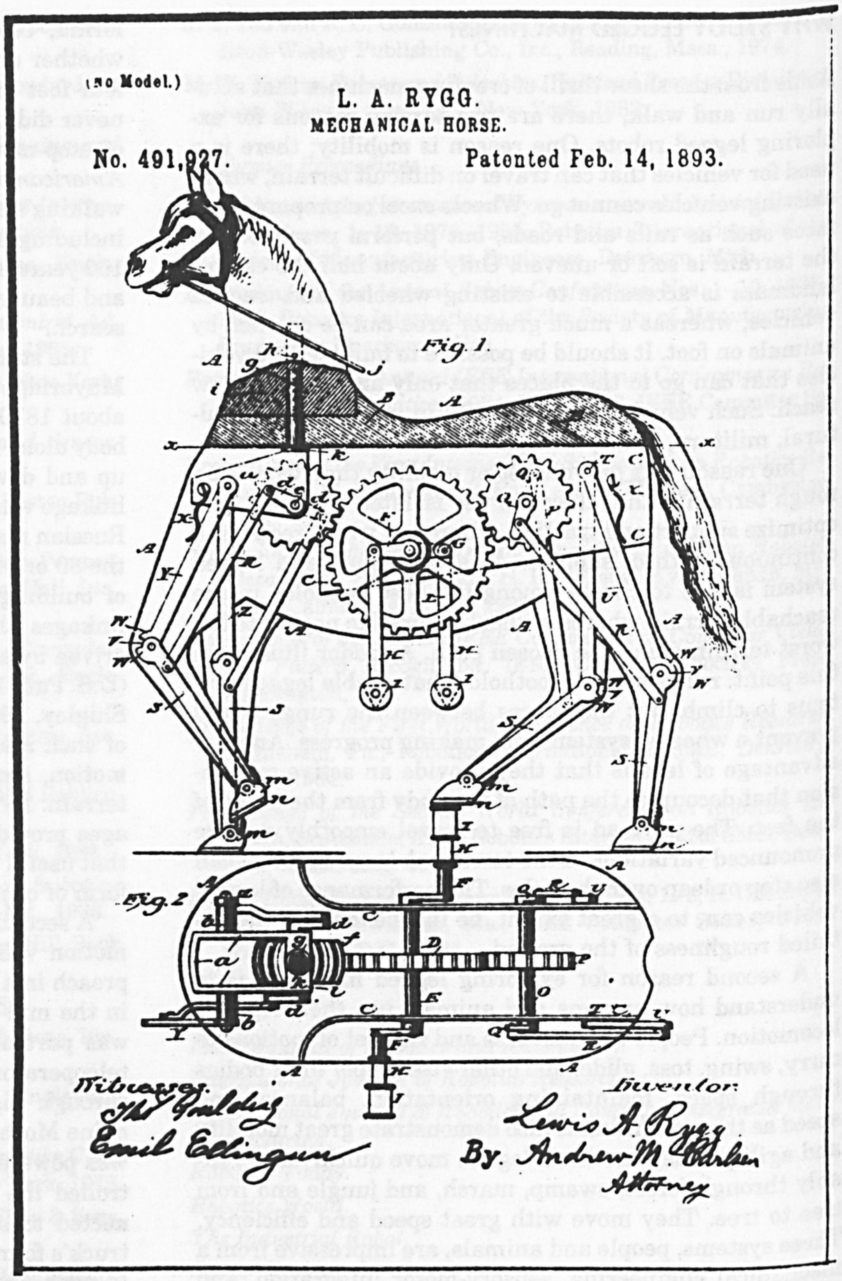

Figure 1. Mechanical horse (U.S. Pat., 1983). The stirrups double as pedals so the rider can power the stepping motions. The reins move the head and forelegs from side to side for steering.

order to coordinate the 18 electric motors driving the legs. The coordination ensured that the machine's center of mass stayed over the polygon of support provided by the supporting feet while allowing the legs to sequence through a gait (Fig. 3). The machine traveled quite slowly, covering several yards per minute. The hexapod provided McGee with an experimental means of pursuing his earlier theoretical findings on the combinatorics and selection of gait (McGhee, 1968; McGhee and Jain, 1972; Koozekanani and McGhee, 1973).

At about the same time, Gurfinkel and co-workers (1981) in the USSR built a machine with characteristics and performance quite similar to McGhee's (Okhotsimski and co-workers, 1977; Devjanin and co-workers, 1983). It used a hybrid computer for control, with analog computation aiding in kinematic calculations. The group at Ohio State subsequently built a much larger hexapod, called the Adaptive Suspension Vehicle (Fig. 4). It was designed for self-contained operation on natural terrain (Waldron and co-workers, 1984; Pugh and co-workers, 1990). It carries a gasoline engine for power, several computers and a human operator for control, and a laser range sensor for terrain preview. This machine walked at about 5 mph, negotiated simple obstacles on rough terrain, and pulled heavy loads.

Hirose realized that the three basic approaches to controlling legged locomotion, mechanical linkage, human teleoperation, and computer control, are not mutually exclusive. His experience with clever and unusual mechanisms (he had built seven kinds of mechanical snake) led to a leg with special mechanical structure that simplified the control of locomotion and could improve efficiency (Hirose and Umetani, 1980; Hirose and co-workers, 1984). The leg was a three-dimensional pantograph that trans-

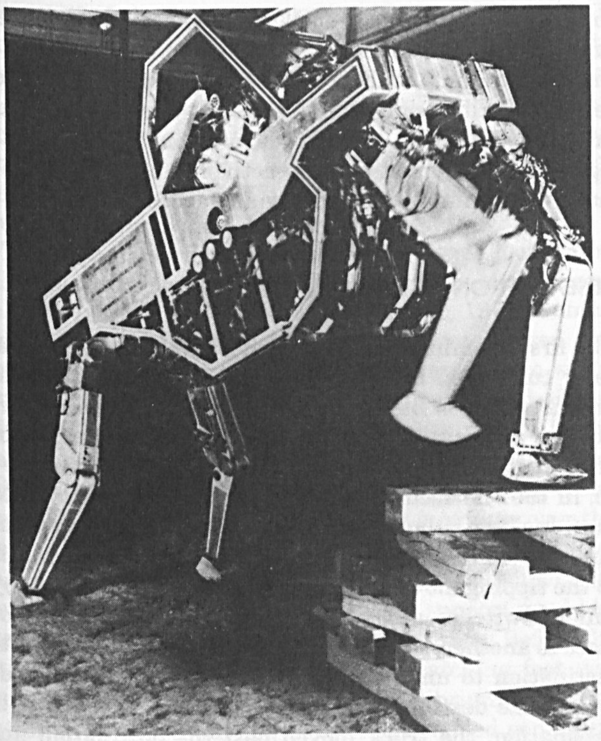

Figure 2. Walking truck developed by Mosher at General Electric in about 1968. The human driver controlled the machine with four handles and pedals that were connected to the four legs hydraulically. Courtesy of General Electric Research and Development Center.

lated the motion of each actuator into a pure Cartesian translation of the foot. With the ability to generate *x*, *y*, and *z* translations of each foot by merely choosing an actuator, the control computer was freed from the task of performing kinematic solutions. The mechanical linkage helped to perform the calculations needed for locomotion. The linkage was efficient because the actuators performed only positive work in moving the body forward and no work was done against gravity when moving on the level. Hirose used the pantograph leg in a small quadruped, about 3 ft long.

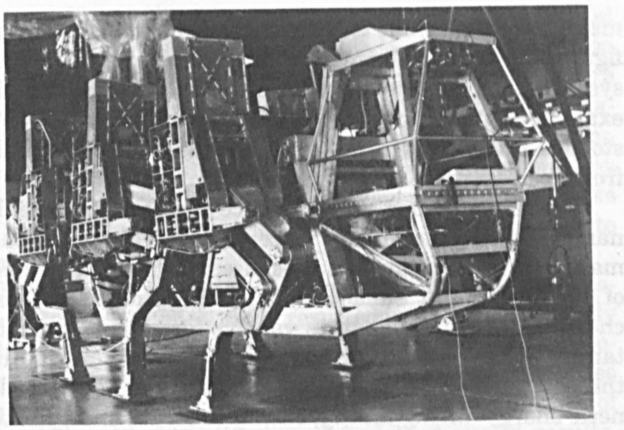

Figure 4. The hexapod walking machine developed at Ohio State University. It stands about 10 ft tall and 15 ft long and weighs 3 t. A 90 hp motorcycle engine provides power to 18 variable displacement hydraulic pumps that drive the joints. The legs use pantograph linkages to improve energy efficiency. The operator normally provides steering and speed commands while computers control the stepping motions of the legs.

These three walking machines, McGhee's, Gurfinkel's, and Hirose's, represent a class called static crawlers. Each differs in the details of construction and in the computing technology used for control, but shares a common approach to balance and stability. They all keep enough feet on the ground to guarantee a broad base of support at all times, and the body and legs move to keep the center of mass over this broad support base. The forward velocity is kept low enough so that kinetic energy can be ignored in the stability calculation. Other machines in this class have also been studied (Russell, 1983; Sutherland and Ullner, 1984; Ooka and co-workers, 1985; Carlton and Bartholet, 1987; Bares and Whittaker, 1990).

DYNAMICS AND BALANCE

The study of dynamic legged machines that balance actively will now be considered. These systems operate in a regime where the velocities and kinetic energies of the

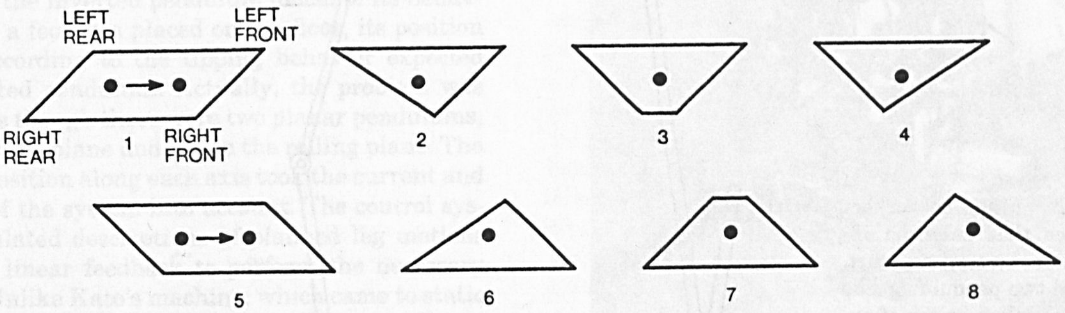

Figure 3. Statically stable gait. The diagram shows the sequence of support patterns provided by the feet of a quadruped walking with a crawling gait. The body and legs move to keep the projection of the center of mass within the polygon defined by the feet. A supporting foot is located at each vertex. The dot indicates the projection of the center of the mass. Adapted from McGhee and Frank (1968).

masses are important to the behavior. Geometry and configuration are not adequate in themselves to model such systems that move with substantial speed. Consider, for example, a fast-moving vehicle that might tip over if stopped suddenly with its center of mass too close to the front feet.

In order to predict and influence the behavior of a dynamic system, the energy stored in the velocity of each mass, in the elevation of each mass, and in the deflection of each complaint element must be considered. The exchange of energy among its various forms is also important in dynamic legged systems. For example, in running the body's potential energy of elevation changes into kinetic energy during falling, then into strain energy when parts of the leg deform elastically during rebound with the ground, then into kinetic energy again as the body accelerates upward, and finally back into potential energy of elevation. Energy is also exchanged in dynamic walking: the kinetic and gravitational potential energies oscillate out of phase throughout the cycle. Because these energies are out of phase, strain energy storage is modest during walking. This sort of dynamic exchange is central to legged locomotion.

Dynamics play a role in giving legged systems the ability to balance actively. A statically balanced system avoids tipping and the ensuing horizontal accelerations by keeping its center of mass over the polygon of support formed by the feet. In contrast, a dynamic legged system is always tipping. The control system avoids tipping too far by manipulating body and leg motions to ensure that each tipping interval is brief and that each tipping motion in one direction is compensated by a tipping motion in the opposite direction. An effective base of support is thus maintained over time. The ability of an actively balanced system to depart from static equilibrium relaxes the rules governing how legs can be used for support. For example, if a legged system can tolerate tipping, then it can position its feet away from the center of mass in order to use footholds that are widely separated or erratically placed. On the other hand, by keeping the feet near the centerline the system can travel where there is only a narrow path of good support. If a legged system can tolerate intermittent support, then it can move all its legs to new footholds at one time, to jump onto or over obstacles, and to use short periods of ballistic flight for increased speed. Animals routinely exploit active balance to travel on difficult terrain; legged vehicles must balance actively, too, if they are to move with animallike mobility and speed.

RESEARCH ON ACTIVE BALANCE

The first machines that balanced actively were automatically controlled inverted pendulums. A person can balance a broom on his or her finger with relative ease. Why not use automatic control to build a broom that can balance itself? Claude Shannon was probably the first to do so. In 1951 he used the parts from an erector set to build a machine that balanced an inverted pendulum atop a small powered truck. The truck drove back and forth in response to the tipping movements of the pendulum, as sensed by a pair of switches at its base. In order to move from one place to another, the truck first had to drive away from the destination to unbalance the pendulum, then proceeded toward the destination. In order to balance again at the destination, the truck moved past the destination until the pendulum was again upright with no forward velocity, then moved back to the destination.

At Shannon's urging, Cannon and his students at Stanford University set about demonstrating controllers that balanced two pendulums at once. In one case the pendulums were mounted side by side on the cart, and in the other they were mounted one on top of the other (Fig. 5). They also demonstrated balance for a flexible inverted pendulum (Schaefer, 1965; Schaefer and Cannon, 1966). Cannon's group was interested in the single-input multiple-output problem and in the limitations of achievable balance: how could the cart's motor be used to control the angles of two pendulums as well as the position of the cart? How far from balance could the system deviate before it was impossible to return to equilibrium, given such parameters of the mechanical system as the cart motor's

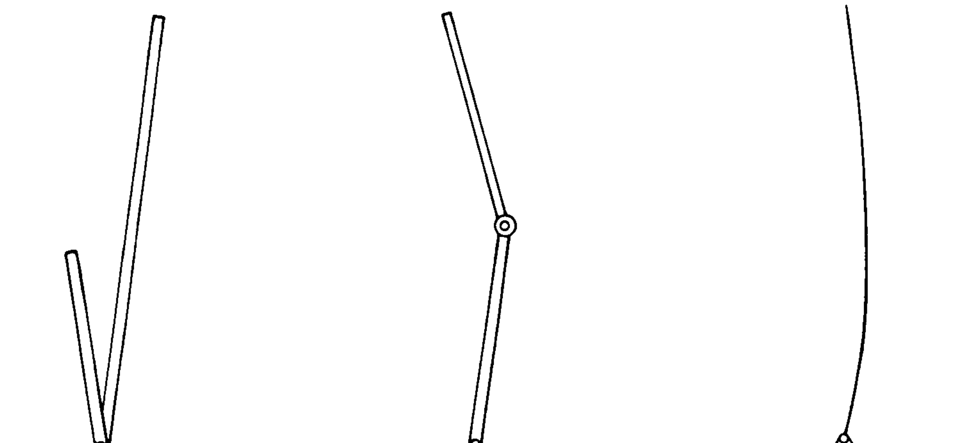

Figure 5. Cannon and co-workers built machines that balanced inverted pendulums on a moving cart. They balanced two pendulums side by side, one pendulum on top of another, and a long, limber inverted pendulum. Only one input, the force driving the cart horizontally, was available for control. Adapted from Schaefer and Canon (1966).

strength and the pendula lengths? These studies of balance for inverted pendulums were important precursors to later work on locomotion. The inverted pendulum model for walking would become a primary tool for studying balance in legged systems (Hemami and Weimer, 1974; Vukobratovic and Stepaneko, 1972; Vukobratovic, 1973; Hemami and Golliday, 1977; Kato and co-workers, 1983; Miura and Shimoyama, 1984).

The importance of active balance in legged locomotion was recognized for some years (Manter, 1938; McGhee and Kuhner, 1969; Frank, 1970; Vukobratovic, 1973; Gubina and co-workers, 1974), but the first dynamic legged systems did not appear until about 1980. Kato and co-workers (1983) built a biped that walked with a quasi-dynamic gait (Ogo and co-workers, 1980). The machine had 10 hydraulically powered degrees of freedom and two large feet. This machine was usually a static crawler, moving along a preplanned trajectory to keep the center of mass over the base of support provided by the large supporting foot. Once during each step, however, the machine temporarily destabilized itself to tip forward so that support would be transferred quickly from one foot to the other. Before the transfer took place on each step, the catching foot was positioned to return the machine to equilibrium passively. No active response was required. The inverted pendulum model was used to plan the tipping and catching motions. This machine walked with a quasi-dynamic gait, taking about a dozen 0.5-m steps per minute. Kato's approach represents an interesting way to achieve dynamic behavior. The system was not dynamic in the sense of reacting at run time to the progress of the motion. Instead, an off-line analysis of the dynamics of the system specified how to position the catching foot statically to get run-time dynamic behavior. Knowledge of dynamics of the system was "compiled" into a simple run-time strategy.

Miura and Shimoyama (1984) built a walking machine that may have been the first to balance itself actively. Their stilt biped was patterned after a human walking on stilts. Each foot provided only a point of support, and the machine had three actuators: one for each leg that moved the leg sideways and a third that separated the legs fore and aft. Because the legs did not change length, the hips were used to pick up the feet. This gave the machine a pronounced shuffling gait reminiscent of Charlie Chaplin's stiff-kneed walk. Control for the stilt biped relied, once again, on the inverted pendulum model of its behavior. Each time a foot was placed on the floor, its position was chosen according to the tipping behavior expected from an inverted pendulum. Actually, the problem was broken down as though there were two planar pendulums, one in the pitching plane and one in the rolling plane. The choice of foot position along each axis took the current and desired state of the system into account. The control system used tabulated descriptions of planned leg motions together with linear feedback to perform the necessary calculations. Unlike Kato's machine, which came to static equilibrium before and after each dynamic transfer, the stilt biped tipped all the time.

Furusho and Masubuchi (1987a, 1987b) used an inverted pendulum model in conjunction with an hierarchical control scheme. A low level servo moved each joint toward a desired position. An inverted pendulum model was derived from the dominant modes of the closed-loop system. The set points for the closed-loop system were selected to produce a walking motion. Their five-link planar biped started from standing, walked at 0.8 m/s, then returned to a standing position. Later machines built by Furusho and Sano (1990) with seven and nine links walked in three dimensions and used ankle torques to control forward speed.

Later work in Kato's laboratory (Takanishi and co-workers, 1985, 1989, 1990b) resulted in a series of actively balanced walking bipeds. They modeled the electric motors as point masses and the links as massless. Because the electric motors are so much heavier than the links of the robots, this simplified model yielded useful results. One of these machines has a large upper trunk that moves back and forth and side to side to stabilize the walking motion and to balance when pushed (Fig. 6).

McGeer (1989, 1990) took an unusual and elegant approach to dynamic walking, called passive dynamic walking. He was motivated by the stable behavior of toy animals that walk downhill by waddling from side to side and by work on ballistic walking, which suggested that humans swing their legs passively during the swing phase of walking (McMahon and Mochon, 1980, 1981). McGeer designed machines that travel downhill using no sensors, actuators, or computers. They rely on appropriate choices

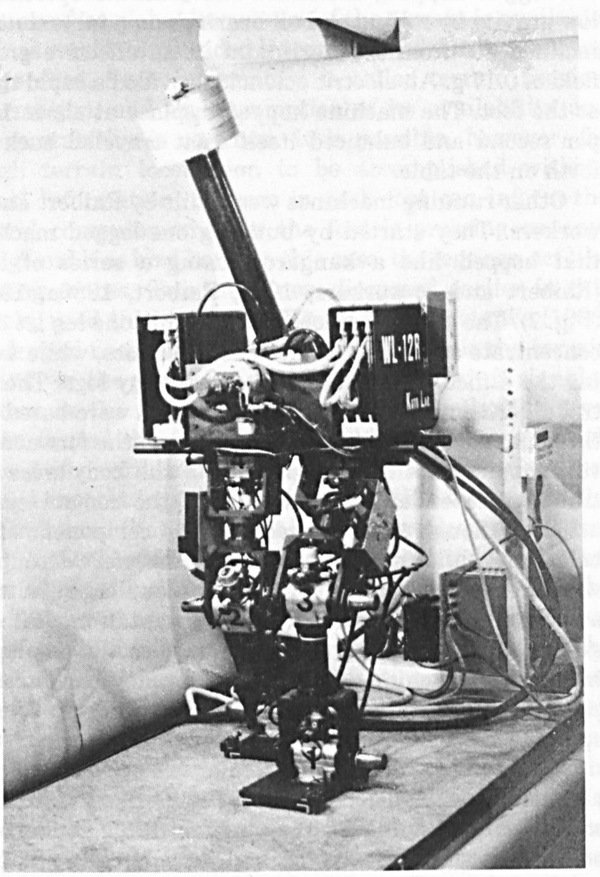

Figure 6. Dynamic walking biped built by Takinishi. The control system stabilizes the walking motion by moving the upper trunk back and forth.

of machine geometry (link lengths, link masses, joint damping, walking surface slope, and foot shape) to do the computing. McGeer's machines can be thought of as mechanical computers whose algorithms are embedded in the form of the structure. These machines are remarkable because they obtained stable dynamic walking from such a parsimonious design. Future legged machines will probably combine the passive characteristics of McGeer's design with the active control found in most other robots.

RUNNING MACHINES

Running is a form of legged locomotion in which all feet leave the ground at some point in the cycle. In animals, running is usually associated with travel at high speed. Matsuoka (1980) was the first to build a machine that ran; the goal was to model repetitive hopping in humans. He formulated a model with a body and one massless leg and simplified the problem by assuming that the duration of the support phase was short compared with the ballistic flight phase. This extreme form of running, in which nearly the entire cycle is spent in flight, minimizes the influence of tipping during support. This model permitted Matsuoka to derive a time-optimal state feedback controller that provided stability for hopping in place and for low speed translations.

To test the method for control, Matsuoka built a planar one-legged hopping machine. The machine operated at low gravity by rolling on ball bearings on a table that was inclined 10° from the horizontal in an effective gravity field of 0.17 g. An electric solenoid provided a rapid thrust at the foot. The machine hopped in place at about 1 hop per second and balanced itself as it traveled back and forth on the table.

Other running machines were built by Raibert and coworkers. They started by building one-legged machines that hopped like a kangaroo, using a series of leaps (Raibert and co-workers, 1984; Raibert, 1986a, 1986b) (Fig. 7). The study of machines with only one leg let them concentrate on active balance and dynamics, while avoiding the difficult task of coordinating many legs. The control algorithms for the hopping machines were based on a three-part decomposition of the problem: the forward travel, hopping motion, and posture of the body were controlled independently. Each part of the control system acted as though it influenced just one component of the behavior, while interactions due to imperfect decoupling were treated as disturbances. This decoupling resulted in a particularly simple control implementation. Raibert's group extended the control algorithms used for one-legged hopping to machines that ran on two legs and trotted, paced, and bounded on four. These machines traveled at specified speeds, changed gait during running, ran fast (13 mph), jumped, maintained balance when disturbed, climbed stairs, and performed rudimentary gymnastic maneuvers (Hodgins and co-workers, 1985; Raibert and co-workers, 1986; Koechling and Raibert, 1988; Hodgins and Raibert, 1990; Raibert, 1990).

An important aspect of these running machines was the mechanical springiness built into their legs. To a large

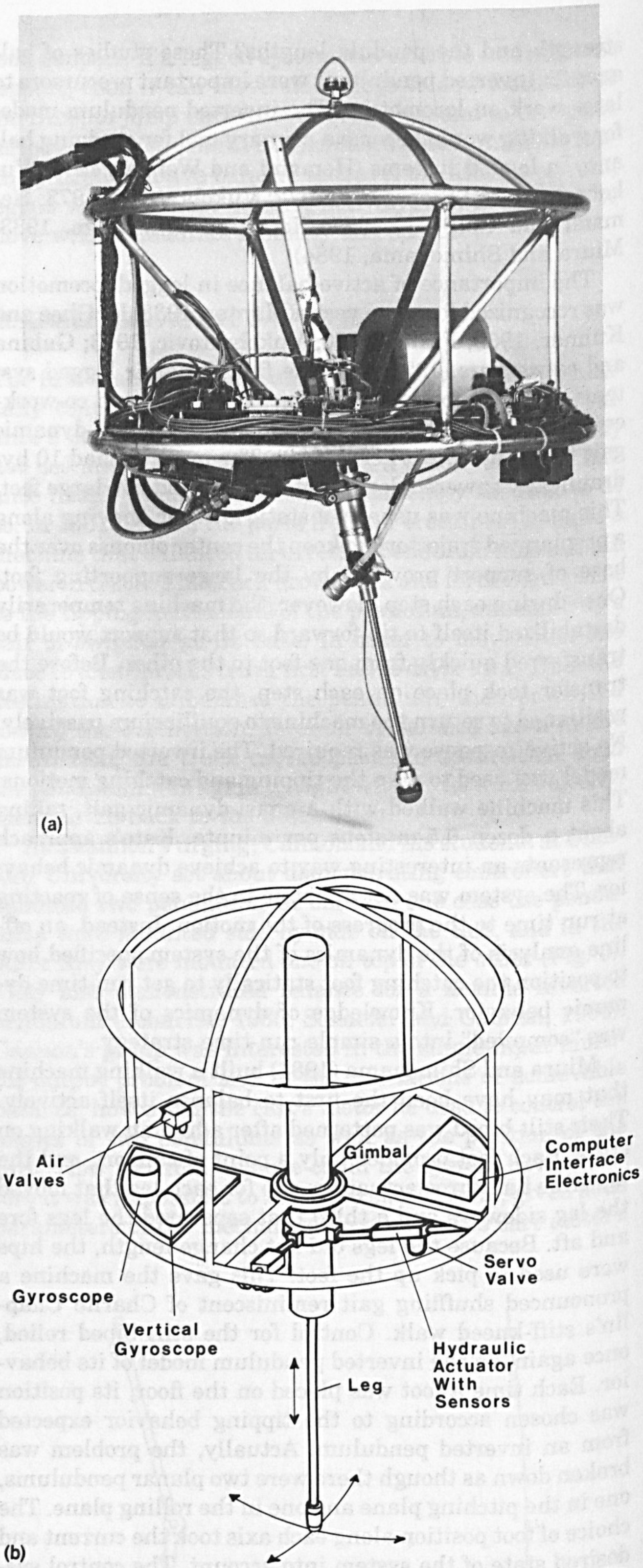

Figure 7. Three dimensional hopping machine used for experiments. The control system operates to regulated hopping height, forward velocity, and body posture. Top recorded running speed was about 2.2 m/s (4.8 mph).

degree, the hopping motion was a passive elastic rebound that occurred when the downward motion of the body was reversed by the springy leg. The control algorithm responsible for hopping excited and modulated these bouncing motion, but did not specify the details of the trajectory. The control system and the mechanical system worked together to produce the hopping behavior. Synergy between mechanism and control is a theme gaining importance in robotics.

LOCOMOTION ON ROUGH TERRAIN

The practical value of legged machines is tied to the promise of mobility on rough terrain. Static and dynamic locomotion systems can both play useful roles in this context. Because they generally move slowly, statically stable systems can acquire data from terrain sensors before they move, interpret the sensor information, and find sequences of feasible footholds. Dynamically stable systems can use their kinetic energy to bridge large gaps that may separate one foothold from the next, or to jump over obstacles. These approaches to rough terrain are discussed in this section.

There are several ways that characteristics can make terrain rough and, therefore, difficult to negotiate.

- The terrain may not be level.
- Traction may be limited (slippery).
- There may be areas of poor or nonexistent support (holes).
- Vertical variations may cause uneven altitude of footholds; or may include large obstacles between footholds (poles).
- The terrain may offer only intricate footholds (eg, rungs of a ladder).

For a statically stable system to travel on rough terrain, the control must decide which locations on the terrain provide suitable footholds and then place the feet on the chosen footholds. A suitable foothold is one that provides adequate support, allows the system to keep its balance, and permits progress toward the goal. It is assumed that the task of placing a foot on a desired foothold, once chosen, is easy for a statically stable system. Okhotsimski and Platonov (1973, 1975) devised such an algorithm for a simulated six-legged walker. Information from a range finder provided terrain information to identify feasible footholds. Their algorithm took the physical structure of the walking machine into account so as to choose footholds that minimized the maximum force exerted by any leg and to keep each foot's reaction force as nearly vertical as possible. Their simulations traveled on three-dimensional poles and holes terrain using the sequence of support polygons found by the foothold selection algorithms.

Several workers have developed algorithms for statically stable rough terrain locomotion. These algorithms first choose a desired path along which the body is to move, followed by heuristic selection of reachable footholds along the path. This approach is called the bodymotion-then-footholds paradigm. McGhee and Iswandhi (1979) used such a paradigm for six-legged walking. After specifying the body motion, their algorithm lifted the legs with the least kinematic travel available in the direction of travel, while putting legs into support with the largest available travel. This heuristic extended each support state to increase the probability that it would overlap with the next support state. Avoidance of deadlock was emphasized over stability by maximizing the number of legs in the air.

Hirose (1984) applied the bodymotion-then-footholds paradigm to quadruped rough-terrain locomotion. He developed an hierarchical algorithm with one level providing gait control and another level providing basic motion regulation. The motion regulation performed such functions as controlling the pitch and height of the body and preventing collisions between the legs. In Hirose's computer simulations, a quadruped walked across terrain with holes, crossed a river, and made local modifications to the motion trace to avoid a large hole.

Okhotsimski and co-workers built a six-legged walking machine that could climb up onto a small ledge (Devjanin and co-workers, 1983). The machine climbed by raising its body and then placing each foot up on the ledge. Care was taken to keep the body level during climbing. They also implemented algorithms to control the forces in the legs during locomotion on soft soil (Gorinevsky and Shneider, 1990).

The work on rough-terrain locomotion described above assumes that global knowledge of the terrain is available, such as would be provided by computer vision or scanning range data. Such information is called terrain preview information, and the assumption of its availability and interpretability is common. It is possible, however, for rough terrain locomotion to be accomplished without global information. Sensors in each foot can inform the control computer whether the foot is pressing on anything, and if so how hard. In this case, the legs can act like sensing probes. Two machines, discussed earlier in this article, used this approach. Hirose and Umetani's (1980) quadruped used switches on the feet to probe the terrain directly in the path of the robot. If a foot switch signaled contact as the foot advanced forward, a reflexlike algorithm caused the foot to be pulled back, lifted, then advanced forward again. Another reflexlike algorithm caused support legs to push downward if a load cell in the foot indicated that it was not bearing an adequate vertical load. A third reflex caused the relative altitude of the feet to be adjusted so the body remained level, as indicated by an oil-damped pendulum. Hirose's quadruped used these reflexes to climb up and down steps without a model of the terrain and without human intervention. The Ohio State University Adaptive Suspension Vehicle could also travel on rough terrain without global terrain information (Pugh and co-workers, 1990). It used force sensors in the leg actuators and a force distribution control algorithm to accomodate variations in the terrain, without foothold selection or planning. They made convincing demonstrations of their machine's ability to travel on gentle slopes, through a muddy cornfield, and over railroad ties, all without a visual sensor or human foothold selection.

So far, we have considered rough terrain in the context of static slow-moving systems. Dynamic legged systems should be able to traverse more difficult terrain than static systems of comparable size and reach. In principle, dynamic legged systems should be able to use balance to travel where available footholds provide only a narrow base of support, to use kinetic energy to travel where the available footholds form an erratic pattern of support, and to use ballistic flight phases to leap over regions of terrain that offer no good support at all. Generally, a dynamic legged system can use its kinetic energy as a bridge from one foothold to the next. These potential advantages of dynamic legged systems are obtained at the expense of more complicated control for placing the feet. Foot placement is straightforward in statically stable systems, once a reachable foothold has been chosen. In dynamic legged systems, however, the act of positioning the feet with respect to available footholds interacts with the need for stability. Each placement of a foot on the ground causes the body to accelerate, and influences the forward speed and direction of travel. The algorithm responsible for placing the feet must manipulate the dynamic parameters of the system to simultaneously balance the machine and keep it moving as desired.

Takanishi and co-workers (1990a) built several robots that walk dynamically on simple rough terrain. Their most recent machine walks up and down 0.1 m stairs and on ±10° slopes. The control algorithms are based on a generalization of zero moment point control, as introduced by Vukobratovic and Stepaneko (1972). When the terrain is not smooth the zero moment point moves along a surface connecting the point of contact of the feet rather than along the surface of the ground. Given knowledge of the terrain, trajectories for the feet, and the zero moment point, the control system precomputes trunk and waist motions that execute the planned trajectories and cause the machine to walk.

Hodgins studied algorithms for running on rough terrain (Hodgins and Raibert, 1991). Three methods were implemented for adjusting the robot's stride to place the feet on desired footholds while maintaining balance. Each method adjusted a different parameter of the running cycle, the forward running speed, the running height, and the duration of ground contact. All three control methods were successful in manipulating step length in laboratory experiments on a planar biped running machine, but the method that adjusted forward speed provided the widest range of step lengths with accurate control of step length. Hodgins used these control methods to demonstrate a robot placing its feet on a target, leaping over an obstacle, and running up and down a short flight of stairs (Fig. 8).

DEVELOPMENT OF USEFUL LEGGED ROBOTS

Most of the legged machines that have been built are not yet useful vehicles or even prototypes for such vehicles. They are experimental apparatus used in the laboratory to explore ideas about legged locomotion. Each was designed to isolate and examine a few specific locomotion problems, while postponing or ignoring many other problems. A summary of the development of legged robots is given in Table 1. It is important to ask what problems remain to be solved before legged robots are transformed into practical machines that do useful work.

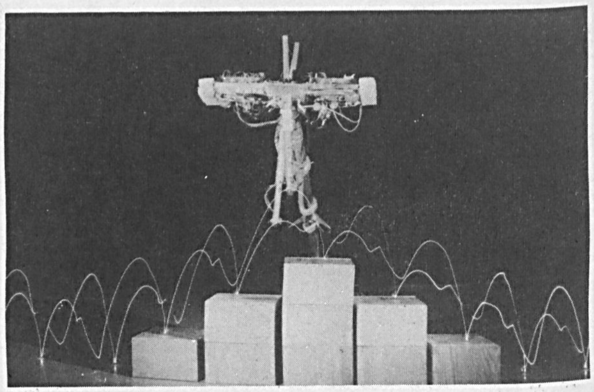

Figure 8. Planar biped running up and down a flight of three stairs. The control system adjusts the length of the machine's steps so that the feet land approximately in the center of each stair. The machine is shown running from right to left at about 0.5 m/s. Light sources indicate the paths of the feet. Each stair is 0.18 m high and 0.3 m deep.

Terrain Sensing

Perhaps the most severe problem limiting current walking machines as well as other forms of autonomous vehicle, is their inability to perceive the shape and structure of their immediate surroundings. Humans and animals use their eyes to locate good footholds, to avoid obstacles, to measure their own rate and direction of progress, and to navigate with respect to visible landmarks. The problem of giving machines the ability to see has received intensive and consistent attention for the last 25 or 30 years. There has been steady progress during that period. Current machines can see well enough to operate in well-structured and partially structured environments, but it is difficult to predict when legged machines will be able to see well enough to operate autonomously in rough outdoor terrain. Such autonomous machine behavior will probably not be seen for at least ten years.

Sensors simpler than vision may be able to provide solutions to certain parts of the problem under certain circumstances. For instance, sonar and laser range data may be used to detect and avoid nearby obstacles. Motion data may be used for measuring speed and direction of travel with techniques that are substantially simpler than those needed to perceive shape in three dimensions.

Travel on Rough Terrain

Complete knowledge of the geometry of the terrain, as might be supplied by vision or other senses would not in itself solve the problem of locomotion on rough terrain. A system traveling on rough terrain needs to know or figure out what terrain shapes provide good footholds, which sequence of footholds would permit traversal of the terrain, and how to move so as to place the feet on the available

Table 1. Milestones in the Development of Legged Robots

1836	Weber and Weber	Measurements in human corpses show that natural frequency of leg as compound pendulum is similar to cadence in walking
1850	Chebyshev	Designs linkage used in early walking mechanism (Lucas, 1894)
1872	Muybridge	Develops stop-motion photography to document running animals
1893	Rygg	Patents human-powered mechanical horse
1945	Wallace	Patents hopping tank with reaction wheels that provide stability
1961	Space General	Eight-legged linkage machine walks in outdoor terrain (Morrison, 1968)
1963	Cannon, Higdon and Schaefer	Control system balances single, double, and limber inverted pendulums
1968	Frank and McGhee	Simple digital logic controls walking of Phony Pony
1968	Mosher	GE quadruped truck climbs railroad ties under control of human driver
1969	Bucyrus-Erie Co.	Big Muskie, a 15,000 t walking dragline is used for strip mining; it moves in soft terrain at a speed of 900 ft/h (Sitek, 1976)
1977	McGhee	Hexapod robot walks with wave gait using digital computer to coordinate leg motion
1977	Gurfinkel	Hybrid computer controls hexapod walker in USSR
1977	McMahon and Greene	Human runners set new speed records on tuned track at Harvard; its compliance is adjusted to mechanics of human leg
1980	Hirose and Umetani	Quadruped machine climbs stairs and climbs over obstacles using simple sensors and reflexlike control; the leg mechanism simplifies control
1980	Kato	Hydraulic biped walks with quasidynamic gait
1980	Matsuoka	Mechanism balances in the plane while hopping on one leg
1981	Miura and Shimoyama	Biped balances actively while walking in three-dimensional space
1983	Sutherland	Self-contained hexapod carries human rider; computer, hydraulics, and human share computing task
1983	Odetics	Self-contained hexapod lifts and moves back end of pickup truck (Russell, 1983)
1983	Raibert	One-legged machine hops in place, travels at a specified rate, keeps its balance when disturbed, and jumps over small obstacles
1984	Furusho	Planar five-link biped starts walking from a standing position, and travels at 0.8 m/s
1987	Waldron and McGhee	Three-ton self-contained hexapod carrying human driver travels at 5 mph, travels in irregular terrain, and pulls a load
1988	Hodgins and Koechling	Planar biped climbs short stairway, jumps over obstacles, and runs with top speed of 13.1 mph.
1989	Raibert	Quadruped runs with trotting, pacing, and bounding gaits, and changes between gaits
1990	McGeer	Planar biped with knees walks passively down sloping surface

footholds. It will be necessary to coordinate the dynamics of the vehicle with the dynamics of the terrain.

The techniques that will allow legged systems to operate on rough terrain will involve the mechanics of locomotion, kinematics, dynamics, geometric representation, spatial reasoning, and planning. Although course- and medium-grain knowledge of the terrain will be important, it is expected that techniques that make legged systems inherently insensitive to fine-grain terrain variations will play an important role too. Ignoring the hard sensing issues mentioned earlier, it is believed that the perception and control mechanisms required for legged systems to travel on rough terrain will require a substantial research effort, but can be solved within the next 10 years if they are pursued vigorously.

Mechanical Design and System Integration

When these sensing and control problems are solved, it will remain to develop mechanical designs that function with efficiency and reliability. Useful vehicles must carry their own power, control computers, and a payload. A host of interesting problems present themselves including such matters as energy efficiency, structural design, strength and weight of materials, and efficient control. For instance, the development of materials and structures for efficient storage and recovery of elastic energy will be particularly important for legged vehicles. It is expected that initial useful legged vehicles can be built with existing mechanical and aerospace techniques.

BIBLIOGRAPHY

J. E. Bares and W. L. Whittaker, "Walking Robot with a Circulating Gait," in *IEEE International Workshop on Intelligent Robots and Systems*, 1990, pp. 809–816.

V. V. Biletskii, "Biped Locomotion Dynamics I," *Izv. AN SSSR Mekhanika Tverdogo Tela* 10(3), 3–14 (1975).

R. E. Carlton and S. J. Bartholet, "The Evolution of the Application of Mobile Robotics to Nuclear Facility Operations and Maintenance," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Raleigh, N.C., 1987, pp. 720–726.

E. A. Devjanin, V. S. Gurfinkel, E. V. Gurfinkel, V. A. Kartashev, A. V. Lensky, A. Y. Schneider, and L. G. Shtilman, "The Six-Legged Walking Robot Capable of Terrain Adaptation," *Mechanisms Machine Theory* **18**, 257–260 (1983).

A. A. Frank, "An Approach to the Dynamic Analysis and Synthesis of Biped Locomotion Machines," *Med. Bio. Eng.* **8**, 465–476 (1970).

J. Furusho and M. Masubuchi, "A Theoretically Motivated Reduced Order Model for the Control of Synamic Biped Locomotion," *J. Dynam. Sys. Contr.* **109**, 155–163 (1987a).

J. Furusho and M. Masubuchi, "Control of a Dynamical Biped

Locomotion System for Steady Walking," in H. Miura and Shimoyama, eds., *Study on Mechanisms and Control of Bipeds,* University of Tokyo, Tokyo, 1987b, pp. 116–127.

J. Furusho and A. Sano, "Sensor-Based Control of a Nine-Link Biped," *Int. J. Robot. Res.* **9**(2), 83–98 (1990).

D. M. Gorinevsky and A. Y. Shneider, "Force Control in Locomotion of Legged Vehicles over Rigid and Soft Surfaces," *Int. J. Robot. Res.* **9**(2), 4–23 (1990).

F. Gubina, H. Hemami, and R. B. McGhee, "On the Dynamic Stability of Biped Locomotion," *IEEE Trans. Biomed. Eng.* **21**, 102–108 (1974).

V. S. Gurfinkel, E. V. Gurfinkel, A. Y. Shneider, E. A. Devjanin, A. V. Lensky, and L. A. Shitilman, "Walking Robot with Supervisory Control," *Mechanism Machine Theory* **16**, 31–36 (1981).

H. Hemami and C. L. Golliday, Jr., "The Inverted Pendulum and Biped Stability," *Math Biosci.* **14**, 95–110 (1977).

H. Hemami and F. C. Weimer, "Further Considerations of the Inverted Pendulum," in *Proceedings of the Fourth Iranian Conference on Electrical Engineering,* Shiraz, Iran, 1974, pp. 697–708.

S. Hirose, "A Study of Design and Control of a Quadruped Walking Vehicle," *Int. J. Robot. Res.* **3**, 113–133 (1984).

S. Hirose, M. Nose, H. Kikuchi, and Y. Umentani, "Adaptive Gait Control of a Quadruped Walking Vehicle," *Int. J. Robot. Res.* **1**, 253–277 (1984).

S. Hirose and Y. Umetani, "The Basic Motion Regulation System for a Quadruped Walking Vehicle," in *ASME Conference on Mechanisms,* 1980.

J. Hodgins, J. Koechling, and M. H. Raibert, "Running Experiments with a Planar Biped," in *Third International Symposium on Robotics Research,* MIT Press, Cambridge, Mass., 1985.

J. Hodgins and M. H. Railbert, "Biped Gymnastics," *Int. J. Robot. Res.* **9**(2), 115–132 (1990).

J. Hodgins and M. H. Railbert, "Adjusting Step Length for Rough Terrain Locomotion," *IEEE J. Robot. Automat.* **7**(3), 289–298 (1991).

T. Kato, A. Takanishi, H. Jichikawa, and I. Kato, "The Realization of the Quasi-Dynamic Walking by the Biped Walking Machine," in A. Morecki, G. Bianchi, and K. Kedzior, eds., *Fourth Symposium on Theory and Practice of Robots and Manipulators,* Polish Scientific Publishers, Warsaw, 1983, pp. 341–351.

J. Koechling and M. H. Raibert, "How Fast Can a Legged Robot Run?" in K. Youcef-Toumi and H. Kazerooni, eds., *Symposium in Robotics,* DSC Vol. 11, American Society of Mechanical Engineers, Boston, 1988.

S. H. Koozekanani and R. B. McGhee, "Occupancy Problems with Pairwise Exclusion Constraints—An Aspect of Gait Enumeration," *J. Cybernet.* **2**, 14–26 (1973).

R. A. Liston and R. S. Mosher, "A Versatile Walking Truck," in *Proceedings of the Transportation Engineering Conference,* Institution of Civil Engineers, London, 1968.

E. Lucas, "Huitieme recreation—La machine a marcher," *Recreat. Math.* **4**, 198–204 (1894).

T. McGeer, "Powered Flight, Child's Play, Silly Wheels, and Walking Machines," in *IEEE Conference on Robotics and Automation,* Scottsdale, Ariz., 1989.

T. McGeer, "Passive Dynamic Walking," *Int. J. Robot. Res.* **9**(2), 62–82 (1990).

R. B. McGhee, "Some Finite State Aspects of Legged Locomotion," *Math. Biosci.* **2**, 67–84 (1968).

R. B. McGhee, "Vehicular Legged Locomotion," in G. N. Saridis, ed., *Advances in Automation and Robotics,* JAI Press, 1983.

R. B. McGhee and A. A. Frank, "On the Stability Properties of Quadruped Creeping Gairs," *Math. Biosci.* **3**, 331–351 (1968).

R. B. McGhee and G. I. Iswandhi, "Adaptive Locomotion of a Multilegged Robot over Rough Terrain," *IEEE Trans. Sys. Man Cybernet.* **9**, 176–182 (1979).

R. B. McGhee and A. K. Jain, "Some Properties of Regularly Realizable Gait Matrices," *Math. Biosci.* **13**, 179–193 (1972).

R. B. McGhee and M. B. Kuhner, "On the Dynamic Stability of Legged Locomotion Systems," in M. M. Gavrilovic and A. B. Wilson, Jr., eds., *Advances in External Control of Human Extremities,* Yugoslav Committee for Electronics and Automation, Belgrade, Yugoslavia, 1969, pp. 431–442.

J. Manter, "Dynamics of Quadrrupedal Walking," *J. Exper. Biol.* **15**, 522–539 (1938).

K. Matsuoka, "A Mechanical Model of Repetitive Hopping Movements," *Biomechan.* **5**, 251–258 (1980).

H. Miura and I. Shimoyama, "Dynamic Walk of a Biped," *Int. J. Robot. Res.* **3**, 60–74 (1984).

S. Mochon and T. A. McMahon, "Ballistic Walking," *J. Biomechan.* **13**, 49–57 (1980).

S. Mochon and T. A. McMahon, "Ballistic Walking: An Improved Model," *Math. Biosci.* **52**, 241–260 (1981).

E. Muybridge, *The Human Figure in Motion,* Dover Publications, New York, 1955. Originally published by Chapman and Hall, Ltd., London, 1901.

E. Muybridge, *Animals in Motion,* Dover Publications, New York, 1957. Originally published by Chapman and Hall, Ltd., London, 1899.

K. Ogo, A. Ganse, and I. Kato, "Dynamic Walking of Biped Walking Machine Aiming at Completion of Steady Walking," in A. Morecki, G. Bianchi, and K. Kedior, eds., *Third Symposium on Theory and Practice of Robots and Manipulators,* Elsevier, Amsterdam, The Netherlands, 1980.

D. E. Okjotsimski, V. A. Gurfinkel, E. A. Devyanin, and A. K. Platonoc, "Integrated Walking Robot Development," in *Conference on Cybernetic Models of the Human Neuromuscular System,* Engineering Foundation, 1977.

D. E. Okjotsimski and A. K. Platonov, "Control Algorithm of the Walker Climbing over Obstacles," in *Proceedings of the Third IJCAI,* Stanford, Calif., Morgan-Kaufmann, San Mateo, Calif., 1973.

D. E. Okjotsimski and A. K. Platonov, "Perceptive Robot Moving in 3D World," in *Proceedings of the Fourth IJCAI,* Tbilisi, Ga., Morgan-Kaufmann, San Mateo, Calif., 1975.

A. Ooka, K., Agi, Y. Wada, Y. Kida, A. Takemoto, K. Okamoto, and K. Yoshida, "Intelligent Robot System II," in *Second International Symposium on Robotics Research,* MIT Press, Cambridge, Mass., 1985, pp. 341–347.

D. R. Pugh, E. A. Ribble, V. J. Vohnout, T. E. Bihari, T. M. Walliser, M. R. Patterson, and K. J. Waldron, "A Technical Description of the Adaptive Suspension Vehicle," *Int. J. Robot. Res.* **9**, 24–42 (1990).

M. H. Raibert, *Legged Robots That Balance,* MIT Press, Cambridge, Mass., 1986a.

M. H. Raibert, "Symmetry in Running," *Science* **231**, 1292–1294 (1986b).

M. H. Raibert, "Trotting, Pacing, and Bounding by a Quadruped Robot," *J. Biomechan.* **23** (Suppl. 1), 79–98 (1990).

M. H. Raibert, H. B. Brown, Jr., and M. Chepponis, "Experiments in Balance with a 3D One-Legged Hopping Machine," *Int. J. Robot. Res.* **3**, 75–92 (1984).

M. H. Raibert, M. Chepponis, and H. B. Brown, Jr., "Running on Four Legs as Though They Were One," *IEEE J. Robot. Automat.* **2,** 70–82 (1986).

M. Russell, Jr., "Odex I: The First Functionoid," *Robot. Age* **5,** 12–18 (1983).

J. F. Schaefer, *On the Bounded Control of Some Unstable Mechanical Systems,* Ph.D. dissertation, Stanford University, Stanford, Calif., 1965.

J. F. Schaefer and R. H. Cannon, Jr., "On the Control of Unstable Mechanical Systems," in *International Federation of Automatic Control,* London, 1966, pp. 6c.1–6c.13.

R. Shigley, *The Mechanics of Walking Vehicles,* Report **7,** Land Locomotion Laboratory, Detroit, Mich., 1957.

G. Sitek, "Big Muskie," *Heavy Duty Equipment Maintenance* **4,** 16–13 (1976).

I. E. Sutherland and M. K. Ullner, "Footprints in the Asphalt," *Int. J. Robot. Res.* **3,** 29–36 (1984).

A. Takanishi, H. Lim, M. Tsuda, and I. Kato, "Realization of Dynamic Biped Walking Stabilized by Trunk Motion on a Sagittally Uneven Surface," in *IEEE International Workshop on Intelligent Robots and Systems,* 1990a, pp. 323–330.

A. Takanishi, G. Naito, M. Ishida, and I. Kato, "Realization of Plane Walking by a Biped Walking Robot WL-10R," in A. Morecki, G. Bianchi, and K. Kedzior, eds., *Fifth Symposium on Theory and Practice of Robots and Manipulators,* MIT Press, Cambridge, Mass., 1985, pp. 383–394.

A. Takanishi, T. Takeya, H. Karaki, and I. Kato, "A Control Method for Dynamic Biped Walking under Known External Force," *IEEE International Workshop on Intelligent Robots and Systems,* 1990b, pp. 795–801.

A. Takanishi, M. Tochizawa, H. Karaki, and I. Kato, "Dynamic Biped Walking Stabilized with Optimal Trunk and Waist Motion," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems,* 1989, pp. 187–192.

U.S. Pat. 491,927 (1893), L. A. Rygg.

U.S. Pat. 1,574,679 (1926), F. A. Nilson.

U.S. Pat. 1,691,233 (1928), A. Ehrlich.

U.S. Pat. 2,430,537 (1947), E. Snell.

U.S. Pat. 2,491,064 (1949), W. E. Urschel.

U.S. Pat. 2,822,878 (1958), P. E. Corson.

U.S. Pat. 2,918,738 (1959), I. R. Bair.

M. Vukobratovic, "Dynamics and Control of Anthropomorphic Active Mechanisms," in A. Morecki, G. Bianchi, and K. Kedzior, eds., *Theory and Practice of Robots and Manipulator Systems, Proceedings of RoManSy'73,* Elsevier, Amsterdam, The Netherlands, 1973, pp. 313–332.

M. Vukobratovic and Y. Stepaneko, "On the Stability of Anthropomorphic Systems," *Math. Biosci.* **14,** 1–38 (1972).

K. J. Waldron, V. J. Vohnout, A. Pery, and R. B. McGhee, "Configuration Design of the Adaptive Suspension Vehicle," *Int. J. Robot. Res.* **3,** 37–48 (1984).

General References

M. G. Bekker, "Is the Wheel the Last Word in Land Locomotion?" *New Sci.* **17,** 406–410 (1961).

D. T. Higdon, *Automatic Control of Inherently Unstable Systems with Bounded Control Inputs,* Ph.D. dissertation, Stanford University, Stanford, Calif., 1963.

D. T. Higdon and R. H. Cannon, Jr., "On the Control of Unstable Multiple-Output Mechanical Systems," in *ASME Winter Annual Meeting,* 1963.

R. A. Liston, "Walking Machine," *J. Terramechanics* **3,** 18–31 (1964).

R. A. Liston, "Increasing Vehicle Agility by Legs: The Quadruped Transporter," paper presented at the Thirty-eighth National Meeting of the Operations Research Society of America, 1970.

R. B. McGhee, "Robot Locomotion with Active Terrain Accommodation," in *Proceedings of the National Science Foundation Robotics Research Workshop,* University of Rhode Island, Providence, 1980.

S. Song and K. J. Waldron, *Machines That Walk,* MIT Press, Cambridge, Mass., 1989.

U.S. Pat. 1,669,906 (1928), E. A. Kinch.

U.S. Pat. 2,371,368 (1942), H. W. Wallace.

W. Weber, *Mechanik der Menschlichen Gehwerkzeuge,* Dieterich'sche Buchhandlung, Goettingen, German, 1836.

Marc H. Raibert
MIT

Jessica K. Hodgins
IBM T. J. Watson Research Center

This research was supported by a grant from the System Development Foundation and a contract from the Defense Advanced Research Projects Agency.

ROBOTS, MOBILE

Mobile robots are vehicles that move autonomously through an unstructured environment, performing a useful mission. Related devices share some of these capabilities, but each differs in at least one important aspect. Teleoperated vehicles perform similar tasks in similar environments, but replace autonomous operation with direct human control. Automatic guided vehicles (AGV), operating in factories, typically follow guide wires embedded in the floor and thus do not work in an unstructured world. Even wind-up toys (or Simon's famous ant on the beach) can move autonomously in unstructured environments and can generate an appearance of sophistication, but they usually do not accomplish anything useful or purposeful. In contrast, real mobile robots operate in buildings with unmapped clutter and unpredictable moving objects or in the rugged outdoor world. They operate for extended periods without direct human supervision. They are designed to perform tasks requiring purposeful mobility, ranging from planetary exploration to materials transport to excavation to planting and harvesting.

The research challenge in building autonomous mobile robots is focused on handling unstructured environments. There are several approaches:

- Impose structure on the environment, such as deploying beacons for positioning.
- Sense the environment, and build internal structures, by image understanding techniques.
- Plan and react to the environment, both before and during moving, to bypass obstacles.

- Engineer mechanical solutions to overcome limitations of the environment.
- Integrate a robot architecture that combines several of the above approaches into a capable system.

POSITION SENSING

Mobile robots need to measure their position and motion. Relative position measurement, which tells how far the vehicle has moved from a previous vehicle position, is important for local maneuvering (has that obstacle been cleared yet?) and for perception (what is the effective baseline for motion stereo?). Absolute positioning, in a global coordinate framework, is often important for mission execution (dig at position *X*) and for reporting results (unexpected obstacle detected at position *Y*). Inferring positions from landmarks or motion from visual tracking is still, in general, a difficult problem, although sensing offers some solutions. It is increasingly possible and desirable to find technological solutions for direct position measurement.

Beacons. Both acoustic and electromagnetic beacons are available in a large variety of forms. The simplest beacons are continuously active and are used by the vehicle to measure bearing to the beacon. A single beacon is only useful for homing to that beacon. If the vehicle has a compass, then knowing its bearing to a beacon constrains its position to a line in space. Measuring the angle between two visible beacons constrains the vehicle position to lie on a circle (or sphere, in three dimensions). Three beacons, or two beacons and a compass, will give the exact vehicle location and heading.

Transponders are beacons that require active transmission both by the vehicle and by the beacons. The vehicle emits a pulse, which is received by the transponder. The transponder responds with another pulse, often at a different frequency to reduce confusion. By knowing the speed of sound or light and knowing the delay in the transponder, the total time until the vehicle hears the return pulse gives the range to the beacon. A single transponder, used for range and bearing, plus an onboard compass, is sufficient to give position on the plane.

Passive beacons are reflectors, often used with a laser on the vehicle. Typically the reflectors are all mounted at the same height above a flat floor, with the laser spinning in the same plane. When a vehicle-mounted sensor sees a bright reflection, the angle of the laser gives the bearing to the beacon.

Global positioning system (GPS) is an increasingly popular beacon system based on satellites. Each satellite continuously broadcasts an accurate time reference signal. The receiver compares the time skew between signals from pairs of satellites, which, multiplied by the speed of light, gives the difference in distance from the vehicle to each satellite. Observing four satellites is sufficient to give three-dimensional position. Certain noise effects of GPS are randomly distributed with zero mean, so filtering the position estimates over time will improve accuracy. Other effects, including ionospheric disturbances, introduce a bias into the measurement. Differential GPS uses a second receiver at a known, fixed, nearby location to calculate the current systematic errors and correct for them.

Inertial. Inertial navigation in its pure form uses three single-axis gyros and three accelerometers, mounted orthogonally, to measure three degrees of freedom of orientation and acceleration. Integrating acceleration gives velocity; a second integration gives position. Sources of error range from thermal drift of the gyros to local variation in the earth's gravitational field, which corrupts the accelerometer readings. Because the raw data are doubly integrated, inertial navigation position errors can build up rapidly over time. There are several commonly used solutions for error reduction. If the vehicle can periodically come to a complete stop, the accelerometers can be reset and any accumulated error can be reduced. Other schemes use external measurements, from beacons, landmarks, or shaft encoders, for long-term stability and use inertial sensing for short-term accuracy. The best methods use filtering schemes to combine the various sources of information (Maybeck, 1979).

Dead Reckoning. Dead reackoning is the simplest and least expensive means of position sensing, but also the least accurate. The robot senses its motion by shaft encoders on its wheels and integrates its motion to estimate position. Vehicle heading may be sensed directly by compass, or inferred by the difference in distance traveled between right and left wheels, or calculated by distance traveled and angle of steered wheels. Dead reckoning errors come from the assumption that a revolution of the axle means a fixed distance traveled by the wheel. Several factors make this assumption inaccurate: wheel slip on the ground; low tire pressure, which reduces effective wheel radius; tire squirm when concerning, which causes the vehicle to move slightly tangentially to the tire direction; etc. Approaches to reducing dead reckoning error include instrumenting nondriven wheels to reduce the likelihood of slip or measuring the motion of all wheels and trying to infer actual vehicle motion.

SENSING

Sensing for mobile robots subsumes most of image understanding (qv). Many of the tasks for mobile robots involve straightforward applications of techniques developed for other purposes, such as object recognition. The distinctive sensor processing required for mobile robots involves obstacle detection, map building, and object tracking. Typical sensors include both active (sonar and laser range finders) and passive (video cameras) (see SENSORS AND SENSOR FUSION; see also VISUAL MOTION ANALYSIS; NEURAL NETWORKS).

Sonar

The most common sensor for indoor mobile robots is sonar, due to its low cost and compact size. Typical sonars have a range of about 30 ft, a coverage of approximately 30°, and an accuracy of approximately 2 in. For each pulse, only the distance to the first echo is reported, which could be anywhere within the 30° cone of sensitivity. For use as a soft bumper, this wide scan has the great advantage of low data rate and thus fast and cheap processing. If the sonar is to be used for building maps, however, it requires much more detailed processing to localize the source of the echo. One approach (Elfes, 1987) models the sonar return statistically. Each distance measurement is used to update occupancy probabilities in the sensed area. The area inside the wedge scanned by the sonar does not generate an echo and is, therefore, most likely unoccupied. Occupancy probabilities for that area are reduced. Somewhere along the rim of the wedge, at a distance given by the reading, some object must have generated the echo. The region along the rim is more likely to be occupied, and can have its probability increased. During an actual run, the world is tiled with cells, each initially unknown. As the vehicle moves, takes sonar readings, and updates the occupancy probability of each cell, the outlines of objects gradually emerge and the free space around the robot becomes better defined.

Another set of approaches to sonar interpretation model the geometry of the objects that caused the echo (Leonard and co-workers, 1990; Kuc and Siegel, 1987). Sound waves reflect specularly (ie, at a mirrorlike angle), so echos must come either from a perpendicular surface, from a rounded object, or from the interior of a corner where sound reflects off of both walls. These different geometries will have different effects as the vehicle moves. If the reflection comes from a wall, for instance, the echo location will appear to move with the vehicle, whereas reflections from a corner remain fixed. This sort of reasoning about geometries and physics of the sonar can provide an accurate map, especially from an uncluttered environment where corners and curved objects are relatively rare.

Laser Range Finders

Mobile robots often need much more accurate range data than are available from sonar, especially for terrain mapping outdoors. The current state of the art is scanning laser range finders, (Besl and Jain, 1985; Besl, 1988). For a typical amplitude-modulated system, the output of a laser is modulated with a sine wave, the laser is pointed at an object, and the returned energy is measured. The phase shift between outgoing and returned signals gives the elapsed time for the laser, which is used to measure the distance to the object. The beam is typically directed at a pair of mirrors, first a multifaceted spinning mirror that generates horizontal scans, followed by a nodding mirror that moves those scans vertically down the scene. The resulting “image” is an array of range values. The performance of these systems is limited by a low laser power for eye safety and requires balancing range resolution, number of pixels, scanning time, and maximum range. The performances of a typical first-generation scanner (built by ERIM) and a second-generation scanner (from Perceptron) are given in Table 1. Other range sensing technology uses pulsed lasers and directly measures time of flight or separates the source and the detector and measures range from triangulation.

Range data can be used at several levels of detail. If the robot's surroundings are flat, with only isolated vertical obstacles, thresholding the elevation value is adequate for obstacle detection. In rolling terrain, it is necessary to look through the elevation data for areas that are too steep, too rough, or unreachable from the current vehicle location, and treat those as obstacles. In really rough terrain, it is necessary to reason about vehicle geometry, and search for paths for individual wheels, in order to find feasible paths.

Individual frames of range data do not always provide adequate information. Data from multiple frames must often be combined to fill in gaps in coverage and to provide increased resolution. If the vehicle has a very accurate inertial navigation system, fusing range data can be as simple as averaging the elevations reported at each location from the different scans. For vehicles with less precise relative positioning, the elevation data from different view points must be matched to find the most likely vehicle motion before combining the data. If the range data contains easily identifiable features, such as isolated objects or clean range edges, it is easy to do symbolic feature matching from frame to frame. In more chaotic terrain, it is necessary to do iconic matches, comparing elevation data in the various frames or to use variants of optical

Table 1. Relative Performance of Example Range Scanners

Characteristic	ERIM	Perceptron
Eye safe	yes (?)	yes
Field of view (h × v)	80 × 30	60 × 60 (programmable tilt)
Pixels	256 × 64	256 × 256
Ambiguity interval	20 m	40 m
Depth	8 bits (8 cm)	12 bits (1 cm)
Intensity	8 bits	8 bits
Max range	40 m (?)	50 m
Scan rate	2 frames/s	2 frames/s
Scan direction	top to bottom	programmable
Interface	VME to Sun	VME to Sun
Temperature	narrow range	“Pittsburgh”
Construction	wire wrap	printed circuit
Components	all custom	most off the shelf
Size (w × h × d)	90 × 35 × 45 (cm)	45 × 35 × 35 (cm)
Weight	50 kg	<25 kg
Power	26 VDC	110 VAC

flow tailored for range data (Herbert and co-workers, 1990; Olin and co-workers, 1987). A particularly interesting variation is the TraX system (Bobick and Bolles, 1989), which tracks objects as they are approached by the vehicle. As the object's size in the image increases, TraX generates more detailed segmentations and interpretations. A bush, for instance, might start out as a spherical blob and end up segmented into individual branches.

Passive Ranging

Instead of sensing range directly with range finders, range can also be inferred from images, using stereo or motion processing. Shape from motion methods are particularly popular on mobile robots (Bhanu, 1989; Dutta and Snyder, 1990; Graefe, 1990; Matthies and Shafer, 1987; Matthies and co-workers, 1988). In feature matching methods, individual points or lines are tracked from image to image, and their ranges are calculated by triangulation based on vehicle motion between frames. In optical flow methods, the apparent brightness "flow" of the image is used to directly calculate a depth field, without going through the intermediate step of finding and tracking features. Shape from motion is limited by not being able to calculate depth directly ahead of the vehicle. The point toward which the camera is moving is the focus of expansion (FOE), the image point from which objects seem to expand in optical flow. Objects directly at the FOE will not appear to move as the vehicle approaches, and shape from motion cannot calculate their depth. This is obviously a disadvantage for obstacle detection algorithms. Objects near the FOE will have inaccurate calculated depths, depending on how accurately the true vehicle motion is known. Several methods have been used to increase accuracy of motion methods, including tracking the FOE from the images, or using Kalman filters to update depth estimates.

Monocular Vision

If the environment in which a robot operates is mapped, the vehicle can use visual cues as landmarks to determine its location and motion (Faugeras, 1987; Fennema and co-workers, 1989). This is particularly useful in man-made environments, where many objects have sharp edges and other easy to track features. Systems for indoor navigation typically look for vertical edges, fit lines to the detected edge points, and match those lines to wire-frame models of the building interior. Once lines have been matched, triangulation gives the vehicle's current location. Variations on this method track the lines over many frames to filter the position estimates, or find features in outdoor images, or track horizontal as well as vertical edges.

Example: Road Following

A common task for outdoor mobile robots is following roads using video data, either color or monochrome. There are three general types of system: feature tracking, pattern classification, and neural networks. Feature tracking systems follow the lines, edges, and stripes of main roads (Dickmanns and Zapp, 1986; Kenue, 1989a, 1989b; Kluge and Thorpe, 1990). They usually consist of simple image processing operators, such as edge detectors or template matching systems, managed by a controller process. The controller examines the output of the image processing feature trackers, decides which detected points and lines are valid and which are outliers, updates its model of the current road geometry and passes it off to the steering program, and positions windows for the feature trackers in the next image. Controllers are based on control theory, on robust statistics, or on Hough transforms, all looking for consistent data. More sophisticated programs can examine the discarded features, decide the reason for failure, and update road models or image processing parameters as the road shape changes (eg, at an intersection) or illumination conditions vary.

Unimproved roads, which do not have lines or clean edges to track, are usually followed by pattern classification methods (Crisman and Thorpe, 1990; Turk and co-workers, 1988). These methods examine all the pixels in an image, classify each one according to color (sometimes adding texture or other features), and create an output image with each pixel labeled road or nonroad, perhaps with a confidence value. They then search the image for the most likely road location, fit a road model to the detected location, and update their color models for the next round of classification. Using the entire image, instead of isolated features, and using probabilities, instead of binary classifications, add to the ability of these algorithms to follow roads even in ill-structured conditions of shadows, dirt, broken edges, and changing road appearance.

Recent research in applying neural networks to road following has achieved some impressive results. The Autonomous land vehicle in a neural net (ALVINN) (Pomerleau, 1990) is a fully connected three-layer back-propagation network. It is trained by driving the robot by hand, giving the network the video images as input, and the human driver's steering wheel position as model output. After seeing approximately 200 training images, the network settles and is ready to take control. Typically, running on an unimproved road, ALVINN's hidden units learn large matched filters that look for the entire road or for road edges. Running on roads with lines and stripes, the network tends to settle on representations that look for those features. The advantage of this kind of system is that it is easy to retrain for many different kinds of road. The main disadvantage is that it has no symbolic representation of what it is looking for, so it is difficult to modify its representation without retraining. There is no clean way, for instance, to modify the network to run on a road that is twice as wide as the one for which it has been trained.

PLANNING

The most difficult aspects of robot planning (qv) arise in the lowest levels. High level planning for robots includes all the usual AI planning problems, including scheduling, partitioning plans among multiple intelligent agents, constraint satisfaction, and so forth. Although some of these are difficult problems, they are not unique to mobile robots, they involve symbolic reasoning, and they can often be solved off-line. Most problems of planning routes at large scales are pure geometric problems and, again, can be solved off-line, often by overlaying a grid on the world map and searching for suitable paths along the grid. But at the lowest level, the reasoning has to involve the interface with the real world. Robot trajectory planners, which plan how the vehicle should negotiate the next few meters of terrain, must get beyond symbolic and geometric planning to consider issues of precision and control.

Graphs. Indoor robots typically abstract space into occupied regions (often represented by sets of convex polygons) and free space (Brooks, 1986; Lozano-Perez and Wesley, 1979). This representation is further abstracted into a graph of possible robot paths, either skirting the edge of obstacles (for obstacle-based representations) or running down the middle of corridors of free space (for free-space methods). Additional nodes are added to the graph for the current vehicle position and the goal position, and paths are added to link the start and goal nodes into the graph. The graph is then searched by A^* or similar search methods to find the shortest path to the goal.

Potentials. A different abstraction for the path-planning problem is to use potential field models. Obstacles near the robot generate repulsive forces, whereas the goal attracts the vehicle. Repulsive potentials increase as the robot nears an object, guaranteeing safe clearance. Variants on potential field methods add potentials toward subgoals (Pauton, 1986) or make the obstacle potentials dependent not just on distance but also on closing speed (Krugh, 1984). Potential field methods have the advantage of quick, local calculations and are, therefore, suitable for reflexes during a robot run. Their local evaluation functions, however, allow them to be trapped in local minima, such as inside a U-shaped object, without reaching the goal. Hybrid methods use initial global search to find and eliminate local minima, or to select a rough global path, then use potential methods for fine-scale adjustment (Thorpe, 1984).

Vehicle Models. Typical cross-country trajectory planners cannot classify the world into obstacles and free space, but must instead examine the paths followed by individual wheels, and their interactions with the terrain, to determine traversability. Simple planners look for areas that are too steep, have too large a vertical step, or have vertical spikes that would entrap the vehicle's undercarriage (Kiersey and co-workers, 1988). More advanced cross country planners, in addition consider accuracy constraints (for both perception and vehicle motion) and sensor positioning (Stentz, 1990). Two distant objects may appear to be far enough apart to allow safe passage, but the planner may not be able to guarantee that the vehicle can move to and past the objects accurately enough to miss them. In that case, the planner must generate a path that moves closer to the objects, orient the vehicle so that sensors can see the objects and update their relative positions, and then replan to go through. The planner must also reason about nonholonomic motion constraints (limited turning radius) and variation of traversibility with vehicle orientation (both because of wheel orientation and vehicle shape). So, for instance, a particular ditch may be traversable if the vehicle approaches perpendicularly, but would entrap a wheel if the vehicle's path were nearly parallel to the ditch.

Example: Navlab Planner

The cross-country trajectory planner developed for the Navlab mobile robot incorporates the above concerns (Stentz, 1990). The search starts at the current vehicle configuration and expands, following the constraints of turning radius, to reach the goal. Each path is fattened by maximum expected error, which turns a planned ray in configuration space into a cone of possible trajectories. In order for a path to succeed, all paths within that cone must arrive within the goal configuration envelope without encountering obstacles. If that is impossible, the planner must select an intermediate goal and replan. The search is made efficient by an oct-tree representation of obstacles and free space, and by considering various pruning strategies.

MECHANISMS

Specialized mechanical designs, some specific to robots, facilitate navigation in difficult circumstances. The main designs serve two functions: increasing maneuverability, to help in tight situations with limited clearance, and increasing terrainability, to navigate rugged terrain. (See Robots, legged.)

Steering. Maneuvering in tight locations with a conventionally steered vehicle requires detailed sensing and planning. Even humans occasionally have a difficult time parallel parking. Many indoor robots avoid this problem by using omnidirectional steering. A commonly used design has three wheels, all steered and driven together, to keep the body orientation constant while the vehicle moves in any direction. Most such platforms have a circular cross section, which simplifies path planning, and a rotating turret, so cameras and other sensors can be pointed independently of the direction of the motion.

More complex omnidirectional vehicles allow independent steering of each wheel. This requires careful attention to coordination. A four-wheeled vehicle has eight control parameters (steering angle and wheel velocity for each wheel), but only three degrees of freedom in motion. In order to move without actuator conflict, all wheels must instantaneously be moving along concentric circles, with their axles pointed toward the center of the circle and their speeds proportional to their effective radii.

Rugged Terrain. Designs for rugged terrain range from tracks and wheels to multisegment bodies to walking machines. Many of these have been inspired by planetary exploration, especially the soft sands and rugged, eroded terrain of Mars. Bulldozer-style tracks provide great ter-

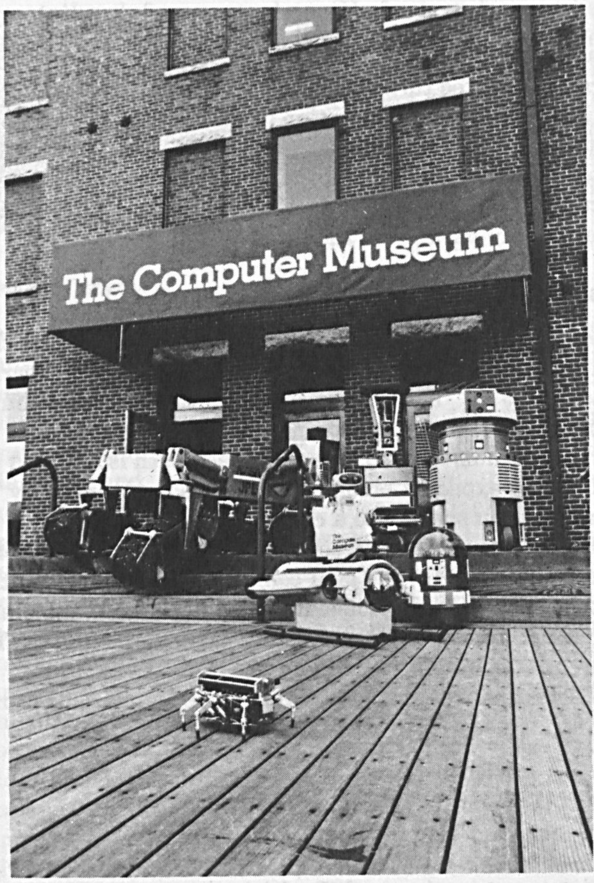

Figure 1. An assortment of mobile robots, real and fanciful. Rear left: the JPL loop-wheel vehicle, designed for planetary exploration. Rear center: Shakey, the pioneering AI-based robot, from SRI. Rear right: a Denning Sentry commercial mobile robot. On steps: Omnibot, a toy robot. Left in front of steps: Sea Rover, a highly mobile underwater vehicle, used primarily as a flying eyeball. Right in front of steps: RB5X, designed for educational use. Foreground: a robot prop from the movie *Runaway*. Courtesy of the Computer Museum, Boston.

rainability, but require impractical amounts of power, both for skid steering and because small rocks get caught and crushed between track segments. An improved design, the JPL loop-wheel vehicle, uses four stiff circular bands that deflect more than a tire but are not as flexible as treads. Using four smaller bands makes conventional steering feasible, and using one-piece loops instead of steel segments decreases rock entrapment.

A different approach is to build multisegment vehicles, with each segment having one or two pairs of wheels. Designs include segments connected with a flexible backbone, segments connected with active articulation that can lift the body over steep steps, and even robots with passive wheels and active articulation that move like a snake. Specialized mechanisms include a family of pipe climbers (Fukuda and co-workers, 1990) and a family of wall climbers (Nishi and co-workers, 1990). The first of the wall robots at Waseda University used a suction fan to hold the robot body to the wall and used wheels for propulsion. The second is a biped walker, with suction cup feet, that is more capable of surmounting ridges and ledges. The latest in the series uses a small aircraft propeller to hold the robot against the wall and to propel it at speeds up to 5 m/s.

Legs. Several families of walking machines have been developed, including dynamically stable hoppers (Raibert, 1989) and the Ohio State Adaptive Suspension Vehicle (Song and Waldron, 1989). Walking robots provide high terrainability and are much more efficient than wheeled vehicles in rough and soft terrain (see ROBOTS, LEGGED).

Example: AMBLER

The largest and perhaps most unusual walking land robot today is the AMBLER (Bares and co-workers, 1989). It has six orthogonal legs, each with a rotational shoulder and prismatic elbow operating in the horizontal plane, and a prismatic vertical outer link. On rough terrain, the AMBLER adjusts its vertical axes to keep the body horizontal, allowing efficient translational motion. Whenever possible, the body is kept at a constant elevation, saving the energy that a wheeled vehicle would expend climbing over every rock it encountered. The AMBLER's legs are arranged in two sets, with the three legs on each side sharing a common shoulder axis. The legs move in a novel "circulating" gait. Each leg in turn is placed ahead of the body, then moves past the outer edge of the body during its propulsion phase. Legs are recovered (brought forward to begin the walking cycle) by swinging them through a cavity in the middle of the body. This continuous rotation of the shoulders allows each leg to be placed only once while the body moves up to three meters, rather than requiring each leg to pick up and shuffle forward several times to follow the leg ahead of it, as in most walking machines.

Walking machines pose special planning problems. High speed walking machines must first select an appropriate gait, which defines the order in which legs should be recovered and the number of legs in the air at any one time. Slower machines, such as the AMBLER, move only a single leg at a time. Deciding which leg to move, and where to place it, requires considering terrain, body advance, and vehicle stability constraints. The center of gravity of a statically stable walking machine must be kept over its support polygon, the bounding polygon of the contact points of all feet touching the ground. The ABMLER uses a further constraint, the conservative support polygon (CSP). The CSP is formed by intersecting all of the support polygons formed by considering each set of $N - 1$ supporting feet. If the center of gravity of the AMBLER is held above the CSP, any single foot can fail without the vehicle tipping over. Part of the planning problem, then, consists of placing feet so that the sequence of CSP lines up with the desired direction of motion and allows continuous safe body advances (Wettergreen, 1990).

ARCHITECTURES

The separate parts of a mobile robot's software (sensing, thinking, and acting) must be organized by an architectural framework. An architecture for a complex robot must handle multiple sensors, potentially conflicting subgoals, planning at different levels, and execution monitoring. The architecture must also define interfaces and provide tools to allow many programmers to build the system and to allow multiprocessor execution.

Centralized. The most conventional architectures separate robot software into separate modules for sensing, thinking, and control. This has the advantage of giving one module control of the vehicle, another control of all sensors, and a third control of modeling and planning. This decomposition groups design tasks in the likely areas of expertise of separate research groups. The drawback of this approach is that it does not allow for high speed special-purpose reflexes, that must do sensing, thinking, and control all in one tightly integrated module.

Subsumption. The opposite approach is typified by Brooks (1986) in his subsumption architecture. In his robots, each module covers the complete range from sensory input to control output. He divides his modules into a hierarchy of functions, each subsuming the lower levels. The first module watches sensor data and moves the vehicle away from obstacles. The next layer moves the vehicle randomly, unless the lowest layer takes over to avoid hitting an object. Higher layers add purpose to the wandering (eg, toward open doorways), look for objects of interest, and so forth. Each layer is relatively simple to build, and at least in principle mostly decoupled from adjacent layers. But with no central world model, it takes careful design to ensure that various modules are not working at cross purposes. Related ideas include reactive or reflexive planning, which emphasize quick response rather than careful preplanning, and behaviors, which package sensing and control modes appropriate for specific situations (Payton, 1986).

Hybrid. Several attempts have been made to build architectures that combine the best of both approaches. These systems typically propose a hierarchy, in which sensor interpretation at each level feeds into both planning at the same level and higher level sensor interpretation (Albus and co-workers, 1987). Plans at each level are decomposed into lower level steps, and given to the next lower level for execution. The hierarchies are often structured by time (quick reflexes at the low level, through slower processes at higher levels), data abstraction (raw signals to symbolic reasoning), and space (local effects to global databases). In trying to encompass all possible systems, these general-purpose architectures lose their prescriptive power. Their main contribution may instead be descriptive, providing a common vocabulary in which to discuss the differences between architectures.

Trade-Offs

For current robot systems, there appears to be no single best architecture. A robot such as the AMBLER, which moves very slowly to conserve energy, may be best served by an architecture that centralizes all decision making. The task control architecture (Bares and co-workers, 1989) on the AMBLER collects all control information in a central module, so the best choice can be made before spending energy on moving. Smaller robots with limited computing may be better off with subsumption-style architectures. These robots do not have central world models based on sensor fusion and may not always make optimal decisions, but can react quickly. In a sense they substitute physical search and backtracking, moving through the environment, for the computer-based exploration of search spaces typical of an AI system.

SUMMARY

Mobile robots have evolved extensively from early systems such as Shakey (qv) (Nilsson, 1984) and the Stanford Cart (Moravec, 1983). VaMoRs now drives the autobahn at 100 km/h (Dickmanns and Zapps, 1986). Denning Sentries patrol warehouses at night. Raibert's (1989) machines hop, skip, and jump. EAVE-East looks for underwater objects (Blidberg, 1989). Yet very few truly autonomous vehicles are actually at work. This is partly due to issues of cost and hardware reliability. But it is still largely because of the remaining challenges in dealing with unstructured environments. Developing general-purpose practical robots will require further advances in position sensing, image understanding, planning, mechanical design, and system architectures. Additional open issues remain in coordinating multiple vehicles, sensing and handling moving obstacles, error detection and recovery, integrating vehicle motion and manipulation, and human interfaces.

But even though there are still deep research issues in each of those areas, there is beginning to be more emphasis on bringing together the components that already do work, in the limited domains in which they are reliable, to build complete prototype systems. New robot development and application initiatives, from hazardous waste cleanup to cooperating reconnaissance vehicles to intelligent vehicle highway systems, are all just getting under way. Mobile robots have already made the transition from science fiction to laboratory research. They are now poised for the next transition, to becoming practical working machines.

BIBLIOGRAPHY

J. Albus, H. McCain, and R. Lumia, *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*, Technical Report Technical Note 1235, National Bureau of Standards, Gaithersburg, Md., 1987.

J. Bares, M. Hebert, T. Kanabe, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker, "An Autonomous Rover for Exploring Mars," *IEEE Comput. Mag.* **22**(6) (June 1989).

P. Besl, *Range Imaging Sensors*, Technical Report GMR-6090, General Motors Research Laboratory, Warren, Mich., Mar. 1988.

P. Besl and R. Jain, "Three-Dimensional Object Recognition," *ACM Comput. Surveys* **17**(1) (Mar. 1985).

B. Bhanu, "Understanding Scene Dynamics," in *Proceedings of the Image Understanding Workshop*, Morgan-Kaufmann, San Mateo, Calif., 1989.

D. R. Bildberg, "Autonomous Underwater Vehicles: Current Activities and Research Opportunities," in *Proceedings of IAS-2*. IOS, Amsterdam, The Netherlands, Dec. 1989.

A. F. Bobick and R. C. Bolles, "Representation Space: An Approach to the Integration of Visual Information," in *Proceedings of CVPR*, IEEE, June 1989.

R. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE J. Robot. Automat.* **2**(1) (1986).

J. D. Crisman and C. E. Thorpe, "Color Vision for Road Following," in C. E. Thorpe, ed., *Vision and Navigation: The Carnegie Mellon Navlab*, Kluwer Academic Publishers, Boston, 1990, Chapt. 2.

E. Dickmanns and A. Zapp, "A Curvature-Based Scheme for Improving Road Vehicle Guidance by Computer Vision," in *Proceedings of the SPIE Conference 727 on Mobile Robots*, Cambridge, Mass., 1986.

R. Dutta and M. A. Snyder, "Robustness of Correspondence-Based Structure from Motion," in *DARPA Image Understanding Workshop*, Morgan-Kaufmann, san Mateo, Calif., Sept. 1990.

A. Elfes, "Sonar-Based Real-World Mapping and Navigation," *J. Robot. Automat.* **3** (1987).

O. Faugeras, F. Lustman, and G. Toscani, "Motion and Structure from Motion from Point and Line Matches," in *Proceedings of ICCV'87*, 1987.

C. Fennema, A. Hanson, and E. Riseman, "Towards Autonomous Mobile Robot Navigation," in *DARPA Image Understanding Workshop*, Morgan-Kaufmann, San Mateo, Calif., May 1989.

T. Fukuda, H. Hosokai, and N. Shimasaka, "Autonomous Plant Maintenance Robot," in *Proceedings of IROS 90*, IEEE Industrial Electronics Society, July 1990.

V. Graefe, "An Approach to Obstacle Recognition for Autonomous Mobile Robots, in *Proceedings of IROS 90*, IEEE Industrial Electronics Society, July 1990.

M. Hebert, I. Kweon, and T. Kanade, "3-D Vision Techniques for Autonomous Vehicles, in C. Thorpe, ed., *Vision and Navigation: The Carnegie Mellon Navlab*, Kluwer, Boston, 1990, Chapt. 8.

D. M. Keirsey, D. W. Payton, J. K. Rosenblatt, "Autonomous Navigation in Cross Country Terrain," in *Proceedings of the Image Understanding Workshop*, Morgan-Kaufmann, San Mateo, Calif., 1988.

S. K. Kenue, "Lanelok: Detection of Land Boundaries and Vehicle Tracking Using Image-Processing Techniques. Part I: Hough-Transform, Region-Tracing, and Correlation Algorithms," in *Mobile Robots IV*, SPIE, Nov. 1989a.

S. K. Kenue, "Lanelok: Detection of Land Boundaries and Vehicle Tracking Using Image-Processing Techniques. Part II: Template Matching Algorithms," in *Mobile Robots IV*, SPIE, Nov. 1989b.

K. Kluge and C. E. Thorpe, "Explicit Models for Robot Road Following," in C. Thorpe, ed., *Vision and Navigation: The Carnegie Mellon Navlab*, Kluwer, Boston, 1990, Chapt. 3.

B. Krogh, "A Generalized Potential Approach to Obstacle Avoidance Control," in *Robotics Research Conference Proceedings*, Bethlehem, Pa., Aug. 1984.

R. Kuc and M. W. Siegel, "Physically Based Simulation Model for Acoustic Sensor Robot Navigation," *PAMI* **9**(6) (Nov. 1987).

J. Leonard, H. Durrant-Whyte, and I. J. Cox, "Dynamic Map Building for an Autonomous Mobile Robot," in *Proceedings of IROS 90*, IEEE Industrial Electronics Society, July 1990.

T. Lozano-Perez and M. A. Wesley, "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles," *CACM* **22**(10) (Oct. 1979).

L. Matthies and S. Shafer, "Error Modeling in Stereo Navigation," *J. Robot. Automat.* **3** (1987).

L. Matthies, R. Szeliski, and T. Kanade, "Kalman Filter-Based Algorithms for Estimating Depth from Image Sequences," in *Proceedings of the Image Understanding Workshop*, Cambridge, Mass., 1988.

P. S. Maybeck, *Mathematics in Science and Engineering: Stochastic Models, Estimation, and Control*, Academic Press, Inc., New York, 1979.

H. P. Moravec, "The Stanford Cart and the CMU Rover," *Proc. IEEE* **71**(7) (July 1983).

N. J. Nilsson, *Shakey the Robot*, Artificial Intelligence Center Technical Note **323**, SRI International, Menlo Park, Calif., Apr. 1984.

A. Nishi, M. Ohkura, and H. Miyagi, "A Robot Capable of Moving on a Vertical Wall Using Thrust Force," in *Proceedings of IROS 90*, IEEE Industrial Electronics Society, July 1990.

K. Olin, F. Vilnrotter, M. Daily, and K. Reiser, "Developments in Knowledge-Based Vision for Obstacle Detection and Avoidance," in *Image Understanding Workshop*, Los Angeles, 1987.

D. W. Payton, "An Architecture For Reflexive Autonomous Vehicle Control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, 1986.

D. A. Pomerleau, "Neural Network Based Autonomous Navigation," in C. Thorpe, ed., *Vision and Navigation: The Carnegie Mellon Navlab*, Kluwer, Boston, 1990, Chapt. 5.

M. H. Raibert, *Legged Robots That Balance*, The MIT Press, Cambridge, Mass., 1989.

S. Song and K. J. Waldron, *Machines That Walk: The Adaptive Suspension Vehicle*, The MIT Press, Cambridge, Mass., 1989.

A. Stentz, "Multi-Resolution Constraint Modeling for Mobile Robot Planning," in C. Thorpe, ed., *Vision and Navigation: The Carnegie Mellon Navlab*, Kluwer, Boston, 1990, Chapt. 11.

C. E. Thorpe, "Path Relaxation: Path Planning for a Mobile Robot," in *Proceedings of the Fourth National Conference for Artificial Intelligence*, Austin, Tex., AAAI, Menlo Park, Calif., 1984.

M. Turk, D. Morgenthaler, K. Gremban, and M. Marra, "VITS — A Vision System for Autonomous Land Vehicle Navigation," *IEEE PAMI* **10**(3) (May 1988).

D. S. Wettergreen, H. J. Thomas, and C. Thorpe, "Planning Strategies for the AMBLER Walking Robot," in *Proceedings of the IEEE International Conference on Systems Engineering*, Aug. 1990.

Charles E. Thorpe
Carnegie Mellon University

ROSIE

A software tool to facilitate the building of expert systems (qv), ROSIE is a general-purpose, rule-based, procedure-oriented system. The system, when provided with a knowledge base, will become an expert system for the domain embodied by the knowledge base. It was developed around 1981 at Rand Corporation (see J. Fain and co-workers, The Rosie Language Reference Manual, Technical Note N-1647-ARPA, Rand Corp., 1981, and F. Hayes-Roth and co-workers, Rationale and Motivation for Rosie, Technical Note N-1648-ARPA, Rand Corp., 1981).

K. S. Arora
SUNY at Buffalo

ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE

This extensively revised and expanded *Second Edition* of the *Encyclopedia of Artificial Intelligence* defines the discipline by bringing together the core of knowledge from all fields encompassed by AI. It covers the latest developments in current AI topics such as neural networks, fuzzy logic, machine vision, natural language generation, and many more. Includes:

- Over 450 articles—all entries written expressly for the *Encyclopedia*
- Over 5,000 literature references; 454 illustrations and color photographs
- Over 50% new and revised material
- Exemplary indexing and cross-referencing for easy, complete information access to all topics

Praise for the *First Edition*...

"The *Encyclopedia* is a wonder of clarity and scope: surprisingly easy to read ... the clarity is an especially pleasant surprise, considering the articles were all written by AI experts ... It's a treasure house of easily accessible knowledge."

—*Language Technology*

"Excellent bibliographies are attached to most of the articles, and diagrams and sketches are clear and helpful. The indexing and cross-indexing are exemplary. As the editor points out, the reader will be led by the extensive cross-references to almost every other article ..."

—*Artificial Intelligence Reporter*

"The *Encyclopedia* is a first-class piece of work that will be an indispensable part of any AI library."

—*Computing Reviews*

"... A tour de force ... a truly fantastic encyclopedia which no one in the field of artificial intelligence can afford to be without."

—*Systems Research & Information*

WILEY-INTERSCIENCE
John Wiley & Sons, Inc.
Professional, Reference and Trade Group
605 Third Avenue, New York, N.Y. 10158-0012
New York • Chichester • Brisbane • Toronto • Singapore

ISBN 0 471-50307-X (Two-Volume Set)
ISBN 0 471-50305-3 (Vol. 1)
ISBN 0 471-50306-1 (Vol. 2)

ISBN 0-471-50306-1

90000

9 780471 503064