IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Unmanned Systems and Vehicles
Vol. 20, No. 6, Nov/Dec 1990, pp. 1437-1443.

Tele-autonomous Guidance for Mobile Robots [*]

by

J. Borenstein, Member, IEEE, and **Y. Koren**, Senior Member, IEEE

Department of Mechanical Engineering and Applied Mechanics

The University of Michigan - Ann Arbor

Ann Arbor, MI 48109

ABSTRACT

A new technique for the remote guidance of fast mobile robots has been developed and implemented. With this method, the mobile robot follows the general direction prescribed by an operator. However, if the robot encounters an obstacle, it autonomously avoids collision with that obstacle while trying to match the prescribed direction as closely as possible. This novel implementation of *shared control* is completely transparent and transfers control between tele-operation and autonomous obstacle avoidance *gradually*. Our method, called *tele-autonomous* operation, allows the operator to steer vehicles and robots at high speeds and in cluttered environments, even without visual contact.

Tele-autonomous operation is based on the *Virtual Force Field* (VFF) method, which was developed earlier for autonomous obstacle avoidance. The VFF method is especially suited to the accommodation of inaccurate sensor data (such as that produced by ultrasonic sensors) and sensor fusion, and allows the mobile robot to travel quickly without stopping for obstacles.

---

[*] This work was sponsored by the Department of Energy Grant DE-FG02-86NE37969, and by CAMRSS (NASA Center) Grant 025400-6

[**] The material in this paper was partially presented at the *3rd Topical Meeting on Robotics and Remote Systems*, Charleston, South Carolina, March 13-16, 1989

1. Introduction

Conventional tele-operated vehicles or mobile robots rely on visual contact with the operator, either directly or through video transmissions, as shown in Fig. 1a. Guiding such a vehicle is a formidable task, often complicated by the limited view from the TV camera. Under such conditions, a human tele-operator must exercise extreme care, especially in obstacle-cluttered environments. Consequently, the actual traveling speed of the vehicle might be very slow. When dust, smoke, or steam inhibit vision-based guidance, conventional tele-operated activity is ruled out altogether.

To overcome this problem, we have developed a system that combines *autonomous* obstacle avoidance with *tele-operation* into what we call a *tele-autonomous system*. In this system, the tele-operator can guide the mobile robot even without any visual contact: The mobile robot follows the general direction prescribed by the operator; however, when the robot encounters an obstacle, it autonomously avoids collision with that obstacle, trying to match the operator's prescribed direction as close as possible.

As shown in Fig. 1b, the steering command to the mobile robot is based on the *vectorial sum* of two vectors: (1) the operator's reference command $\mathbf{F}_t$, and (2) the obstacle avoidance feedback $\mathbf{F}_r$ generated by the autonomous obstacle avoidance algorithm. The steering of the robot is aligned with the direction of the resultant vector $\mathbf{F}_r + \mathbf{F}_t$ and yields continuous and smooth motion. In the absence of obstacles, $\mathbf{F}_r$

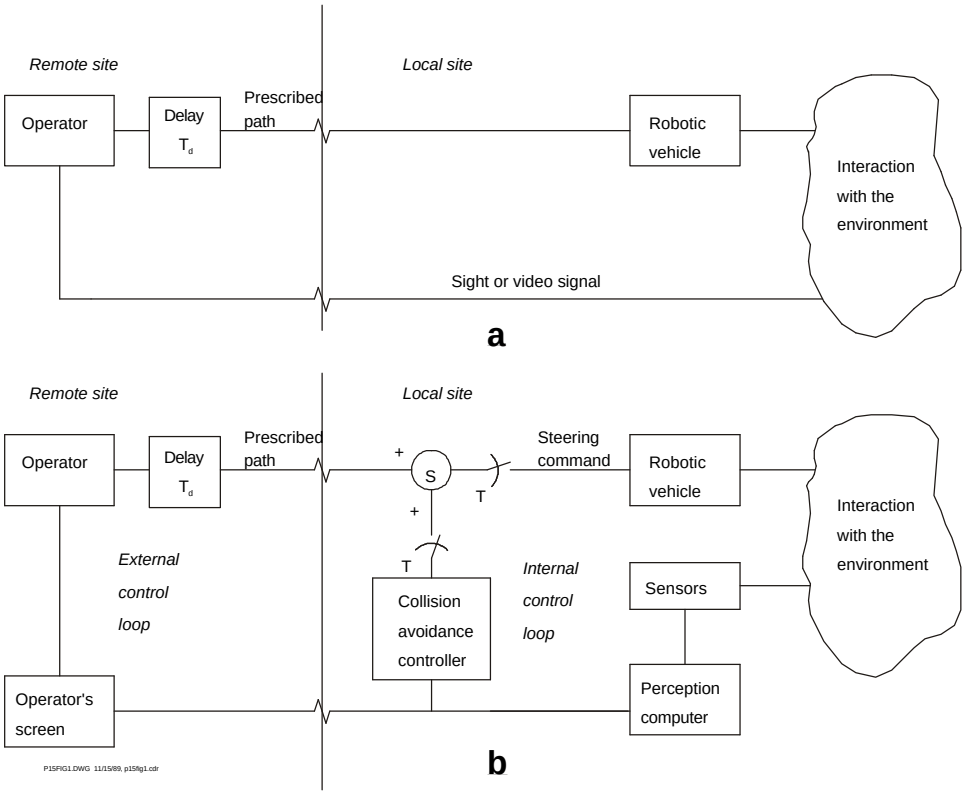

Figure 1: (a) Block diagram of conventional tele-operation. (b) Block diagram of the tele-autonomous system.

= 0, and the robot follows the operator's directions. If the robot approaches an obstacle, $\mathbf{F}_r$ (usually pointing away from the object) gradually increases in magnitude and thus causes a progressive avoidance maneuver. This gradual shift in control is completely transparent.

The *tele-autonomous* system has a *shared control* architecture. In shared control systems, a complex task is divided between the human operator and the machine. In our system, the global control task of guiding the robot to a target is performed by the operator in the external loop, as shown in Fig. 1b. The local control task of guiding the mobile robot around unexpected obstacles is performed in the internal loop by the robot's onboard computer. The control tasks are not switched in a *bang-bang* control fashion. Rather, these tasks are smoothly integrated, and the level of control shifts gradually from the operator to the machine, and vice versa.

As seen in Fig. 1b, the internal loop is of a sampled-data type and operates at a constant sampling frequency $f = 1/T$. Compared to conventional tele-operation, the internal loop causes only a minor delay in the entire operation of our system if the sampling period is small. On the other hand, the performance of the system would be degraded if T was large. Using a version of Shannon's sampling theorem, we may define the boundary condition for *tele-autonomous* operation as follows:

Tele-autonomous operation can be implemented when the sampling frequency of the autonomous loop is greater than $1/T_d$, where T_d is the sum of the operator's delay (due to decision making and response time) and the communication delay (e.g., in space missions).

The minimum sampling period T is dictated by the computation cycle of the internal loop. Therefore, a fast collision avoidance algorithm is a necessary condition for successful tele-autonomous operation. Our experience shows that in practice, the relationship $T < 5T_d$ should be satisfied in order to take full advantage of the integrated autonomous feature. In our system, $T = 26$ ms, which is at least one order of magnitude smaller than the typical response time of an operator.

Information about the robot's environment is derived from its onboard sensors and is shown on the operator's screen, which closes the external control loop. The visual information enables the operator to steer the robot out of difficult trap situations and to the designated target. In our system, ultrasonic sensors are used, although the shared control architecture would work with vision or other sensors as well (provided the constraint on the sampling period T is met).

A comparable approach to tele-autonomous operation has been introduced by Boissiere and Harrigan [1]. This approach, named *telerobotics*, is designed for remotely operated manipulators and was tested on a PUMA robot. The telerobotics concept also couples human commands with computer reasoning in a shared control architecture. The operator commands are communicated to a map and a sensor-based constraint analyzer. Based on knowledge of the environment and the manipulator properties, the analyzer examines the consequences of the operator commands and determines appropriate perturbations, which are communicated to the robot controller.

In the implementation of the telerobotics system, the position of objects in the environment was established by a vision system before the robot commenced its motion. Although real-time sensing of obstacles was not included, some sampling time problems were reported: "New robot position updates are not always available during the 28 ms cycle time of the robot controller." This again shows the importance of a short sampling period in the internal loop of this type of system.

Tele-autonomous operation can compensate for a variety of adverse operating conditions that would not allow conventional tele-operation. Examples of such conditions are:

a. Limited visual contact with the remotely operated vehicle.

b. Emergency situations requiring guidance of vehicles or mobile robots in cluttered environments. In such situations, a human operator may have to perform under considerable psychological stress while speed may be critical.

c. Sensor-motorically impaired operators e. g., users of electric wheelchairs with additional handicaps.

d. Poor communication conditions such as electromagnetic interferences that temporarily disturb transmissions (e.g., radio-frequency interferences) and time delays (e.g., space missions). Under such conditions, the external control loop may be temporarily disabled, while the internal control loop remains intact and protects the robot from collisions.

Tele-autonomous operation is also advantageous for:

a. The simultaneous operation of multiple mobile robots by one operator (provided the robots pursue similar tasks).

b. Tele-operation of multi-link robotic arms equipped with range sensors. The enhanced autonomy and self-protection mechanism provided by *t*ele-autonomous operation makes this approach ideally suited to remotely controlled arms operating under poor visual conditions or with significant time delays [1], [10], [11], [24].

A central element in tele-autonomous operation is the autonomous obstacle avoidance algorithm. We have developed an autonomous obstacle avoidance method for fast mobile robots, called the *Virtual Force Field* (VFF) method [6]. The VFF method is especially suited to the accommodation of inaccurate sensor data (such as that produced by ultrasonic sensors), as well as sensor fusion, and enables continuous motion of the robot without stopping for obstacles. Because of its central role in tele-autonomous operation, the VFF method will be reviewed in Section 3; Section 2 presents a brief review of related methods.

2. Review of Obstacle Avoidance Methods

Tele-autonomous operation requires the integration of real-time obstacle avoidance algorithms with tele-operation. In this section, some of the more popular approaches are reviewed and discussed in view of their applicability to tele-autonomous operation.

2.1 Edge Detection Methods

In a common method for obstacle avoidance the algorithm tries to determine the position of the two farthest vertical edges of the obstacle and consequently attempts to steer the robot around either one of them. The line connecting the two edges is considered to represent one of the boundaries of the obstacle. This method was used in our own previous research [3], [5], as well as in several other investigations, [12], [18], [26]. A disadvantage with edge-detection methods is that the robot is required to stop in front of an obstacle to allow for more accurate measurements.

In *tele-autonomous* operation the robot can not stop to scan its environment for obstacles since this would slow down the operation and violate the necessary condition $T < T_d$. Also, periodic, machine-

generated stops do not conform to the required gradual, transparent shared control architecture. Thus, the described edge-detection methods are unsuitable for tele-autonomous systems.

2.2 The Certainty Grid for Obstacle Representation

A method for probabilistic representation of obstacles in a grid-type world model has been developed at Carnegie-Mellon University (CMU) [14], [19], [20]. The resulting world model, called *certainty grid*, is especially suited to the unified representation of data from different sensors such as ultrasonic, vision, and proximity sensors [21], as well as the accommodation of inaccurate sensor data, which includes the range measurements from ultrasonic sensors.

With the certainty grid world model, the robot's work area is represented by a two-dimensional array of square elements, denoted as cells. Each cell contains a *certainty value* (CV) that indicates the measure of confidence that an obstacle exists within the cell area. With the CMU method, CVs are updated by a probabilistic function C_x. This function is empirically formulated to take into account the characteristics of a given type of sensor.

Ultrasonic sensors, for example, have a conical field of view. A typical ultrasonic sensor [22] returns a radial measure of distance to the nearest object within the cone, yet does not specify the angular location of the object. (Fig. 2 shows the area A in which an object must be located to result in a distance measurement d). It can be shown that if an object is detected by an ultrasonic sensor, it is more likely that this object is closer to the acoustic axis of the sensor than to the periphery of the conical field of view [5]. For this reason, the probabilistic function C_x increases CVs in cells close to the acoustic axis more than in cells at the periphery.

In CMU's implementation of this method [21], the mobile robot remains stationary while taking a panoramic scan with its 24 ultrasonic sensors. After scanning, the certainty grid is updated with the probabilistic function C_x, which is applied for each one of the 24 range readings. The robot then moves to a new location, stops, and the procedure is repeated. After traversing a room in this manner, the resulting certainty grid represents a fairly accurate map of the room. A global path-planning method is then employed for off-line calculations of subsequent robot paths.

This method is unsuitable for tele-autonomous operation, since it requires the robot to stop for scanning, which violates the requirement for gradual, transparent shared control.

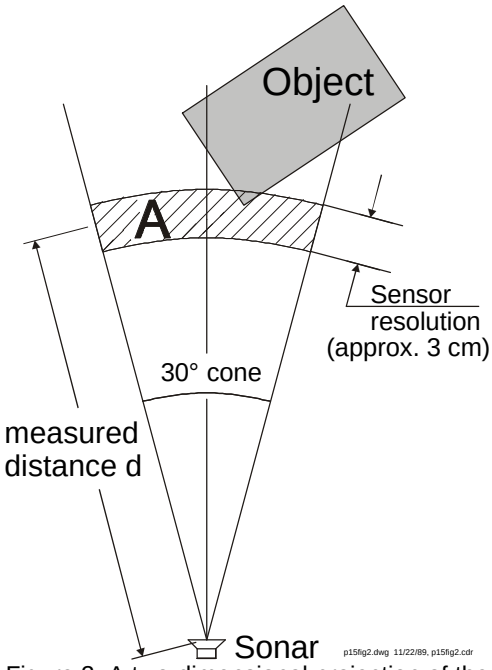

Figure 2: A two-dimensional projection of the conical field of view of an ultrasonic sensor. A range reading d indicates the existence of an object somewhere within the shaded region A.

2.3 Potential Field Methods

The idea of obstacles exerting virtual repelling forces toward a robot, while the target generates a virtual attractive force, has been suggested by Khatib [15]. Krogh [16] uses a similar concept that takes into

consideration the robot's velocity in the vicinity of obstacles. Thorpe [25] applies this potential fields method to off-line path planning, and Krogh and Thorpe [17] later suggest a combined method for global and local path planning that uses Krogh's *generalized potential field* approach.

Common to these methods is the assumption of a known and prescribed world model in which simple, predefined geometric shapes represent simulated obstacles. A robot path is generated off-line through the following steps:

a. Each obstacle generates repulsive forces (usually the normals to well-defined obstacle boundaries), while the goal generates an attractive force.

b. The resultant force vector **R** is calculated for a given robot position. **R** is the vectorial sum of all repulsive and attractive forces.

c. With **R** as the accelerating force acting on the robot, the robot's new position for a given time interval is calculated, and the algorithm returns to step a.

While each of the potential field methods described above features valuable refinements, none of them has been implemented on a mobile robot that uses real sensory data or in simulations that assume unknown environments. Only Brooks [8], [9] uses a force field method in an experimental mobile robot equipped with ultrasonic sensors. Brooks' implementation treats each ultrasonic range reading as a repulsive force vector. If the magnitude of the sum of the repulsive forces exceeds a certain threshold, the robot *stops*, turns into the direction of the resultant force vector, and moves on. This method also requires the robot to stop and thus is not suited for tele-autonomous operation.

3. The VFF Method for Real-time Obstacle Avoidance

The tele-autonomous operation of mobile robots is based on our virtual force field (VFF) method for real-time obstacle avoidance [6]. The VFF method allows the controlled vehicle to travel smoothly among densely cluttered and unexpected obstacles. Since a VFF-controlled vehicle does not stop in front of obstacles, this method is suitable for tele-autonomous operation. In fact, the vehicle slows down only when approaching an obstacle head-on or for dynamic reasons (e.g., if an obstacle forces the vehicle into a sharp turn).

The VFF method combines elements of previous methods (described in Section 2), as well as a number of novel ideas and fast algorithms. This section discusses and summarizes the features peculiar to the VFF method.

3.1 The Histogram Grid for Obstacle Representation

The VFF method uses *histogramic in-motion mapping* (HIMM), a new method for real-time map building with a mobile robot in motion. HIMM represents data in a two-dimensional array (called a *histogram grid*) that is updated through rapid, continuous sampling of the onboard range sensors during motion [7]. Rapid, in-motion-sampling results in a probabilistic map that is well-suited to modeling inaccurate and noisy range data (such as that produced by ultrasonic sensors) and requires minimal computational overhead.

The histogram grid is derived from the certainty grid concept described in Section 2.2. Like the certainty grid, each cell in the histogram grid holds a *certainty value* (CV) that represents the confidence in the existence of an obstacle at that location. The histogram grid differs from the certainty grid in the way it is built and updated. With CMU's method, the certainty grid is updated by projecting a probability profile onto each of those cells that are affected by a range reading. This procedure is computationally intensive and would impose a heavy time-penalty if real-time execution was attempted by an onboard computer.

Our method, on the other hand, creates a probability distribution with little computational overhead. This effect is achieved by incrementing only one cell in the histogram grid for each range reading. For ultrasonic sensors, this cell corresponds to the measured distance *d* (see Fig. 3a) and lies on the acoustic axis of the sensor. While this approach may seem to be an oversimplification, a probabilistic distribution is actually obtained by continuously and rapidly sampling each sensor while the vehicle is moving. Thus, the same cell and its neighboring cells are repeatedly incremented, as shown in Fig. 3b. This results in a histogramic probability distribution in which high CVs are obtained in cells close to the actual location of the obstacle. Our experiments with both methods show [23] that actual rapid sampling from the moving robot (as is the case with HIMM) is more accurate than methods using an assumed probability function.

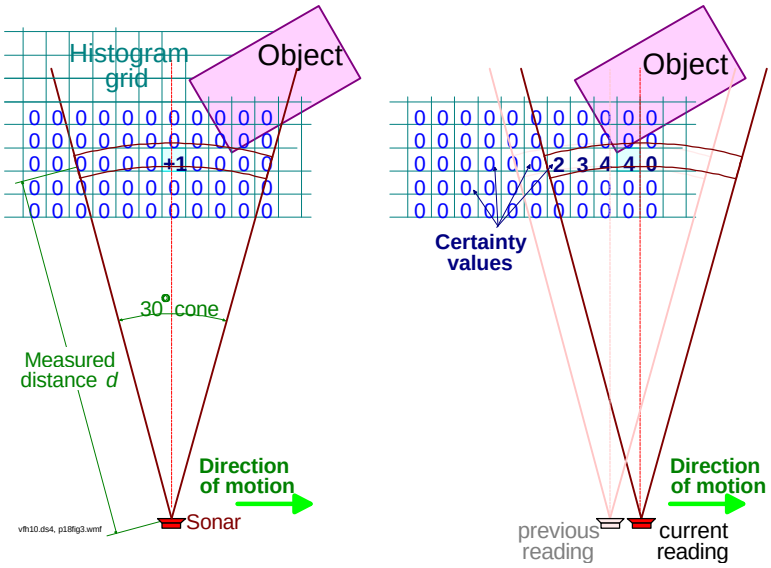

Figure 3: (a) Only one cell is incremented for each range reading. With ultrasonic sensors, this is the cell that lies on the acoustic axis and corresponds to the measured distance *d*. (b) A histogramic probability distribution is obtained by continuous and rapid sampling of the sensors while the vehicle is moving.

Another important feature of HIMM is the *cleaning function*. This function works as follows: Whenever a certain ultrasonic sensor provides a range measurement *d*, the algorithm assumes that the sector between *S* and *A* (see Fig. 3a) is empty and decrements CVs of cells in this sector. This approach was first used in CMU's certainty grid method. However, CMU's method computes and projects a negative probability function for all cells in the sector, while we decrement only those cells that are located on the line connecting center cell C_c and origin cell C_o (i.e., the acoustic axis, see Fig. 3a).

The cleaning function provides two important benefits:

a. Most mobile robots use dead reckoning as the basic means of determining the robot's current position. This method rapidly accumulates position errors. Onboard sensors, however, measure the location of objects relative to the momentary position of the robot; if this position is inaccurate, the position of the object in the histogram grid will also be inaccurate. As the robot traverses a certain area again and again, it will repeatedly encounter the same object. However, due to the accumulating position error of the robot, the location of the object in the histogram grid will shift. A typical mobile robot may accumulate a position error of several meters within just a few hundred meters of accumulated travel [2], [4]. Thus, if the robot passes by an object repeatedly, the object may be "stretched" in the histogram grid over several meters. Consequently, passages may seem blocked even when they are actually wide enough to permit the robot to pass through; therefore, the tele-autonomous system might fail to operate properly.

   The cleaning function remedies this problem by cleaning older images of an object from the histogram grid. Thus, objects in the histogram grid are always shown “correctly” relative to the robot’s assumed new position, and the robot will correctly perform the avoidance maneuver.

b. The cleaning function allows the HIMM algorithm to distinguish between stationary and moving objects. While HIMM does not explicitly identify a moving object as such, the changing location of the object is temporarily noted in the histogram grid, and the robot can avoid the obstacle. As the object moves beyond the robot's “field of view,’ subsequent range readings detect the empty space and decrement the temporarily high CVs accordingly. The resulting histogram grid usually shows only a faint trace of the path of the object. Subsequent robot travel through the path of the object erases the trail completely.

The important benefit of this feature is safe operation in a dynamic environment, e. g., the successful avoidance of slow moving objects[1]. The changing location of the moving object is *temporarily* noted in the histogram grid (high CVs) and the robot can avoid the object. A few seconds after the object has moved on, the cleaning function has reduced the CVs representing the object's previous positions, and the histogram grid shows only a faint trace of the path of the object (low CVs). Thus, the robot can subsequently travel through the path of the object.

Other advantages of HIMM are (see [7] for a detailed discussion):

a. Mapping and obstacle avoidance tasks are integrated – The VFF algorithm can immediately use the map for real-time obstacle avoidance.

b. HIMM builds high CVs with only a few samples - A special *growth-rate operator* increments CVs more rapidly when they belong to a cluster of filled cells (indicating the existence of a real object).

---

[1] *Slow* in this context means that a sufficient number of range readings "show" the object in a certain area. This is a function of the sampling frequency of the sensors, the number of sensors, the size of the moving object, and, of course, the velocity of the object. In our system, a person crossing the robot’s path at walking speed is successfully avoided, even if the robot travels at its maximum speed 0.78 m/s.

In contrast, CVs of isolated cells (probably resulting from noise) are incremented at a slower rate. This feature is important for quick real-time reaction to suddenly appearing obstacles.

3.2 Virtual Force Fields and the Histogram Grid

By applying the potential field idea to the histogram grid world model, the probabilistic sensor information can be used efficiently to control the vehicle. The application of this concept is discussed below, and Fig. 4 illustrates how this algorithm works.

a. A virtual window moves with the vehicle and overlays a square region of the histogram grid. We call this the *active window*, and cells that are momentarily covered by the active window are called active cells. In our current implementation, the active window covers an area of 33×33 cells in the histogram grid.

b. Each *active cell* applies a *virtual repulsive force* $\underline{\mathbf{F}}_{i,j}$ toward the vehicle. The magnitude of this force is proportional to the CV of the cell, and inversely proportional to d^x, where d is the distance between the cell and the vehicle, and x is a positive real number. We will assume $x = 2$ in the following discussion.

c. Next, all virtual repulsive force vectors $\mathbf{F}_{i,j}$ from the active cells are added up to yield the resultant repulsive force vector $\mathbf{F}_r$.

d. A constant-magnitude virtual attractive force $\mathbf{F}_t$ is applied to the vehicle by the target. The summation of $\mathbf{F}_r$ and $\mathbf{F}_t$ yields the resulting force vector **R**.

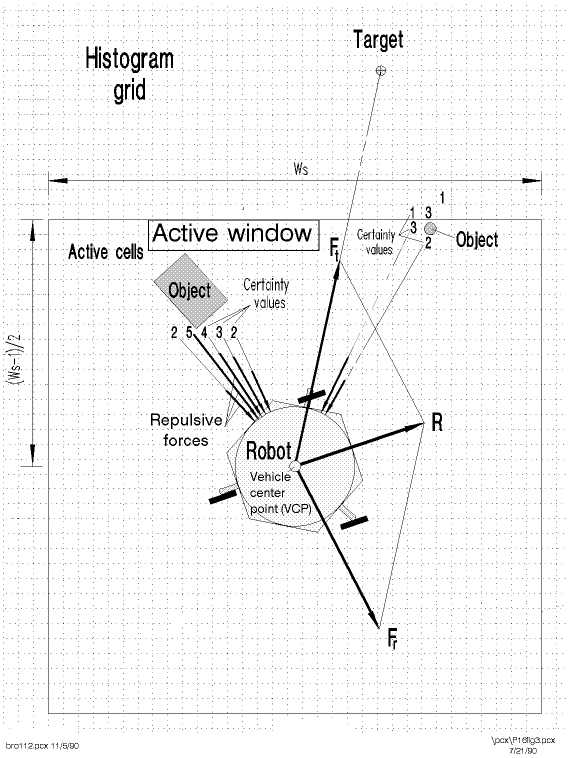

Figure 4: The Virtual Force Field concept: Occupied cells exert repulsive forces toward the mobile robot while the target exerts an attractive force. The robot is steered in the direction of the resultant force R.

In order to compute **R**, the individual repulsive force vectors $\mathbf{F}_{i,j}$ must be computed and accumulated vectorially for each non-zero cell inside the window (i.e., up to 33×33 = 1089 force vectors in our current application). The computational heart of the VFF algorithm is therefore a specially developed algorithm for fast computation and summation of the repulsive force vectors. Note that steps (a through d) are performed simultaneously and asynchronously with the sensor readings and updates of the histogram grid.

One particular advantage of the VFF method lies in the immediate influence of sensor data on the robot's control. In practice, each range reading is inserted into the histogram grid as soon as it becomes available. Therefore, subsequent calculations of **R** will take this data-point into account. In other words, every incremental change in the robot's environment experienced is reflected immediately in the next control command. This feature enables the vehicle to quickly respond to suddenly appearing obstacles, which is imperative when traveling at high speeds.

4. Tele-autonomous Operation

Tele-autonomous operation can be implemented by modifying the VFF algorithm. This is done by replacing the constant target-directed force $\mathbf{F}_t$ with a virtual force in which the direction and magnitude are determined by the operator-controlled joystick. In addition, the operator can control the maximum speed of the vehicle with another input device (e.g., a foot pedal), while the tele-autonomous algorithm adjusts the actual speed to the environmental conditions (see [6] for a comprehensive discussion on autonomous speed adjustments).

As the robot moves, range readings are taken and projected onto the histogram grid, while the algorithm scans the active window as follows: Each active cell exerts a virtual repulsive force $\mathbf{F}_{i,j}$ toward the robot, "pushing' the robot away from the cell. The vectorial sum of all individual repulsive forces (from active cells) $\mathbf{F}_r$ is given by

$$\mathbf{F}_r = \sum_{i,j} \mathbf{F}_{i,j} \tag{1}$$

Each individual repulsive force is defined by

$$\mathbf{F}_{i,j} = \frac{F_{cr} C_{i,j}}{d^2(i,j)} \left[\frac{x_i - x_0}{d(i,j)} \hat{x} + \frac{y_i - y_0}{d(i,j)} \hat{y} \right] \tag{2}$$

where

F_{cr} Force constant (repelling).

$d(i,j)$ Distance between *active cell* (i,j) and the robot.

$C_{i,j}$ Certainty value of *active cell* (i,j).

x_0, y_0 Robot's present coordinates.

x_i, y_j Coordinates of *active cell* (i,j).

Note that:

a. The force constant in Eq. (2) is divided by d^2. Therefore, occupied cells exert strong repulsive forces when they are in the immediate vicinity of the robot, and weak forces when they are further away.

b. The notation chosen in Eq. (2) does not require any trigonometric expressions. Furthermore, almost all of Eq. (2) can be calculated off-line and stored in a look-up table of moderate size. This way, very short computation times are achieved.

The position of the user's joystick prescribes the magnitude and direction of the command vector, $\underline{\mathbf{F}}_t$:

$$\mathbf{F}_t = F_{cj}(\Delta x \hat{x} + \Delta y \hat{y}) \tag{3}$$

where $F_{cj}$is the joystick force constant and Δx, Δy are joystick deflection in x- and y-direction.

Next, $\mathbf{F}_t$ is added vectorially to $\mathbf{F}_r$, producing the resultant force vector $\mathbf{R}$.

$$\mathbf{R} = \mathbf{F}_t + \mathbf{F}_r \quad (4)$$

The direction of **R**, $\delta = \mathbf{R}/|\mathbf{R}|$ (in degrees), is used as the reference for the robot's steering-rate command Ω:

$$\Omega = K_s[\delta - \theta] \quad (5)$$

where K_s is the proportional constant for steering (in sec^{-1}) and θ is the current direction of travel (in degrees).

(The actual implementation of Eq. (5) is more complicated and uses an algorithm that calculates the *shortest rotational difference* between δ and θ.)

Equation 4 is the core of the shared control architecture of the tele-autonomous system. This architecture allows for the gradual real-time transfer of control between the fully autonomous and tele-autonomous modes of operation while the vehicle is in motion.

An indication of the real-time performance of our algorithm is the sampling time T (i.e., the rate at which speed and steer commands for the low-level controller are issued). In our system $T = 26$ ms, using an Intel 80386-based PC-compatible computer running at 20 MHz. The following computations are performed during T:

a. Read sonar information (from the sensor-control computer).

b. Update the histogram grid.

c. Calculate and add (up to) $33 \times 33 = 1089$ individual force vectors inside the active window (Eqs. 1 and 2).

d. Compute the resultant **R** and the steering direction (Eqs. 3 and 4).

e. Calculate the speed command.

f. Communicate with the low-level motion controller (send speed and steer commands and receive position update).

5. Experimental Verification

We have implemented and tested the VFF method on our mobile robot CARMEL (*C*omputer-*A*ided *R*obotics for *M*aintenance, *E*mergency, and *L*ife support). CARMEL is based on a commercially available mobile platform [13] that has a maximum travel speed of $V_{max} = 0.78$ m/s and a maximum steering rate of $V = 120$ deg/s; and weighs (in its current configuration) about 125 kg. The platform has a unique 3-wheel drive (synchro-drive) that permits omnidirectional steering. A Z-80 onboard computer serves as the vehicle's low-level controller.

CARMEL is equipped with a ring of 24 ultrasonic sensors [22] and two additional computers: a PC-compatible single-board computer that controls the sensors, and a 20 MHz, 80386-based AT-compatible computer that performs the computations for tele-autonomous operation.

In extensive tests, we ran CARMEL through difficult obstacle courses under tele-autonomous control. The obstacles consist of unmarked, everyday objects such as chairs, partitions, and bookshelves. In most experiments, the vehicle runs at its maximum speed (0.78 m/sec). This speed is only reduced when an obstacle is approached head-on or if required by the dynamic damping function [6]. The robot successfully avoids unexpected obstacles as small as a vertical pole 3/4 inch in diameter.

A typical experimental setup is depicted in Fig. 5, where obstacles were placed randomly between the starting point S and the target T. In this experiment, the operator's view was completely blocked so that he or she could not see the scene. Only the starting point S and target location T were marked on the operator's screen. Figure 6 shows the screen after the vehicle had been guided successfully from S to T through the obstacle course in Fig. 5. The vehicle's path is shown, along with the shape of the obstacles the vehicle encountered during the run. Each blob in Fig. 6 represents one cell in the histogram grid. In our current implementation, CVs range only from 0 to 3. When CV = 0, no sensor reading was projected onto the cell during the run (and, consequently, no blob is shown). When CV = 1 (or CV = 2), one (or two) readings were projected onto the cell; this is shown in Fig. 6 as blobs comprising of one (or two) pixels. When CV = 3, three or more readings were projected onto the same cell, which is represented by a 4-pixel blob in Fig. 6.

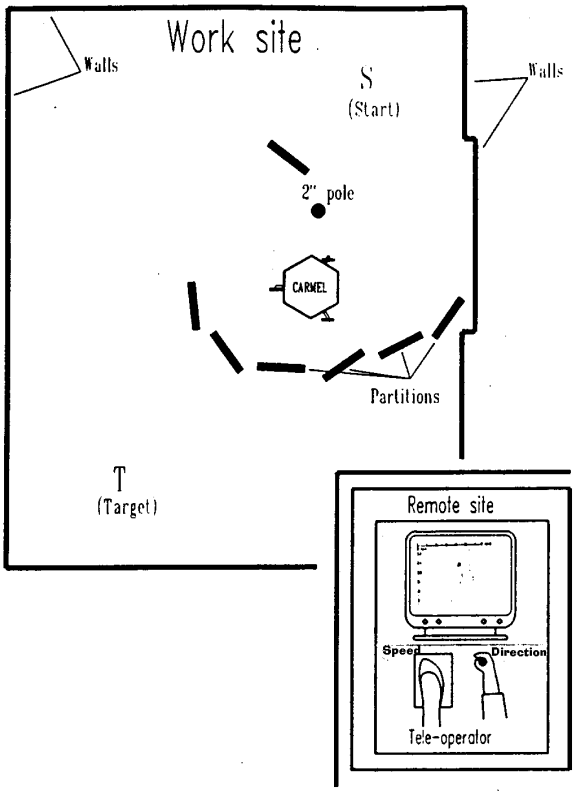

Figure 5: The tele-autonomous concept: An operator guides the mobile robot without direct visual contact, while the robot protects itself from collisions with unexpected obstacles.

We would like to emphasize that in this experiment, the vehicle was guided under tele-autonomous control by an operator who had no visual contact with the vehicle. The vehicle ran at a maximum speed of 0.78 m/sec, but the damping feature [6] reduced the maximum speed as a function of the obstacles present; therefore, the average speed was only 0.5 m/sec.

This experiment also demonstrates the self-protection function of tele-autonomous operation. At point A (in Fig. 6), the operator was directing the robot straight toward T, which would have caused a collision. Under tele-autonomous control, however, the robot diverted from the prescribed direction to avoid the collision. At point B, the shape of the obstacles was sufficiently clear on the operator's screen, allowing him to steer the robot out of the particular obstacle configuration.

6. Summary

Tele-autonomous operation, a new technique for tele-operated guidance of mobile robots, has been introduced. This new technique for the remote guidance of mobile robots uses a shared control architecture, in which the robot's on-site sensing and reflex capacity is combined with human reasoning, analyzing, and decision making. Under tele-autonomous control, the environment conditions and the instantaneous direction of the vehicle dictate whether the operator or the vehicle takes the leading role in directing the vehicle to the target, and to what degree.

Tele-autonomous operation was implemented and tested successfully on an experimental system. Our implementation is based on the following principles:

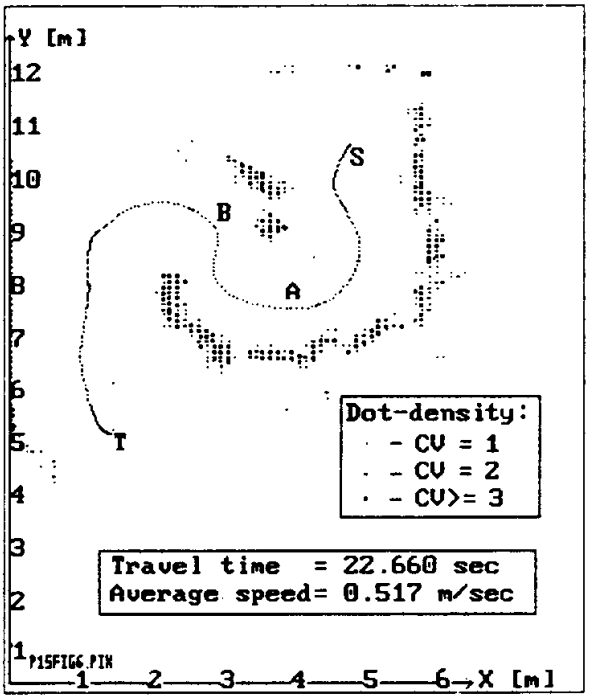

Figure 6: Operator's screen after successful completion of a tele-autonomous task. Obstacles correspond to the setup in Fig. 5.

a. A histogram grid representation of obstacles is obtained through fast sampling during motion.

b. Based on this grid, a field of virtual repulsive forces is created. All repulsive force vectors within an active window are added, resulting in the virtual repulsive force vector $\mathbf{F}_r$.

c. The vectorial summation of $\mathbf{F}_r$ and the virtual force $\mathbf{F}_t$ in the direction of the operator's joystick yields a resultant force **R**, which points to a direction considered free of obstacles.

d. The robot steering is aligned with the direction of **R**.

While ultrasonic sensors are currently used in our mobile robot, practically any other range-finding sensor could be employed instead (or in addition), because the histogram grid representation of obstacles lends itself easily to the integration of data from similar sensors, as well as different types of sensors (such as vision, touch, and proximity).

7. References

1. P. T. Boissiere and R. W. Harrigan, "Telerobotic Operation of Conventional Robot Manipulators." *Proceedings of the IEEE Conference on Robotics and Automation*, Philadelphia, April 25, 1988, pp. 576-583.

2. J. Borenstein, and Y. Koren, "A Mobile Platform For Nursing Robots." *IEEE Transactions on Industrial Electronics*, Vol. 32, No. 2, 1985, pp. 158-165.

3. J. Borenstein and Y. Koren, "Hierarchical Computer System for Autonomous Vehicle." *Proceedings of the 8th Israeli Convention on CAD/CAM and Robotics*, Tel-Aviv, Israel, December 2-4, 1986.

4. J. Borenstein and Y. Koren "Motion Control Analysis of a Mobile Robot." *Transactions of ASME, Journal of Dynamics, Measurement and Control*, Vol. 109, No. 2, 1987, pp. 73-79.

5. J. Borenstein and Y. Koren, "Obstacle Avoidance with Ultrasonic Sensors." *IEEE Journal of Robotics and Automation*. Vol. RA-4, No. 2, 1988, pp. 213-218.

6. J. Borenstein and Y. Koren, "Real-time Obstacle Avoidance for Fast Mobile Robots." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 5, 1989, pp. 1179-1187.

7. J. Borenstein and Y. Koren, "Histogramic In-motion Mapping for Mobile Robot Obstacle Avoidance." *IEEE Journal of Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 535-539.

8. R. A. Brooks "A Robust Layered Control System for a Mobile Robot." *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, 1986, pp. 14-23.

9. R. A. Brooks and J. H. Connell, "Asynchronous Distributed Control System for a Mobile Robot." *Proceedings of the SPIE*, Vol. 727, 1987, pp. 77-84.

10. L. Conway, R. Volz, and M. Walker, "New Concepts in Tele-autonomous Systems." *Second AIAA/NASA/USAF Symposium on Automation, Robotics, and Advanced Computation for the National Space Program*, March 11, 1987.

11. L. Conway, R. Volz, and M. Walker, "Tele-Autonomous Systems: Methods and Architectures for Intermingling Autonomous and Telerobotic Technology." *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 30, 1987.

12. J. L. Crowley, "Dynamic World Modeling for an Intelligent Mobile Robot." *IEEE Seventh International Conference on Pattern Recognition, Proceedings*, Montreal, Canada, July 30 - August 2, 1984, pp. 207-210.

13. Cybermation, 1987, "K2A Mobile Platform." *Commercial Offer*, 5457 JAE Valley Road, Roanoke, VA 24014.

14. A. Elfes, "Sonar-based Real-World Mapping and Navigation." *IEEE Journal of Robotics and Automation*, Vol. RA-3, No 3, 1987, pp. 249-265.

15. O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots." *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, St. Louis, March 25-28, 1985, pp. 500-505.

16. B. H. Krogh, "A Generalized Potential Field Approach to Obstacle Avoidance Control." *International Robotics Research Conference*, Bethlehem, PA, August 1984.

17. B. H. Krogh and C. E. Thorpe, "Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles." *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 7-10, 1986, pp. 1664-1669.

18. R. Kuc and B. Barshan, "Navigating Vehicles through an Unstructured Environment with Sonar." *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, Arizona, May 14-19, 1989, pp. 1422-1426.

19. H. P. Moravec and A. Elfes, "High Resolution Maps from Wide Angle Sonar." *Proceedings of the IEEE Conference on Robotics and Automation*, Washington, D.C., 1985, pp. 116-121.

20. H. P. Moravec, “Certainty Grids for Mobile Robots.” Preprint of *Carnegie-Mellon University, The Robotics Institute*, Technical Report, 1986.

21. H. P. Moravec, “Sensor Fusion in Certainty Grids for Mobile Robots.” *AI Magazine*, Summer 1988, pp. 61-74.

22. Polaroid Corporation, “Ultrasonic Ranging System.” Ultrasonic Ranging Marketing, 1 Upland Road, Norwood, MA 02062.

23. U. Raschke and J. Borenstein, “A Comparison of Grid-type Map-building Techniques by Index of Performance.” *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, 1990, pp. 1828-1832.

24. T. B. Sheridan, “Human Supervisory Control of Robot Systems,” *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, Vol. 2, 1986, pp. 808-812.

25. C. F. Thorpe, “Path Relaxation: Path Planning for a Mobile Robot.” Carnegie-Mellon University, The Robotics Institute, Mobile Robots Laboratory, Autonomous Mobile Robots, Annual Report, 1985, pp. 39-42.

26. C. R. Weisbin, G. de Saussure, and D. Kammer, “SELF-CONTROLLED. A Real-time Expert System for an Autonomous Mobile Robot.” *Computers in Mechanical Engineering*, September, 1986, pp. 12-19.