N88-16386

Task Path Planning, Scheduling, and Learning for Free-Ranging Robot Systems

G. Steve Wakefield
Research Assistant, Johnson Research Center
Research Institute, Room A-4A
University of Alabama in Huntsville
Huntsville, AL 35899
(205) 895-6217

ABSTRACT

The development of robotics applications for space operations is often restricted by the limited movement available to physically guided robots. Free-ranging robots can offer greater flexibility than physically guided robots in these applications. This paper presents an object oriented approach to path planning and task scheduling for free-ranging robots which allows the dynamic determination of paths based on the current environment. The system also provides task "learning" for repetitive jobs. This approach provides a basis for the design of free-ranging robot systems which are adaptable to various environments and tasks.

INTRODUCTION

The development of robotics applications for space operations is often restricted by the limited movement available to physically guided robots. Free-ranging robots can offer greater flexibility than physically guided robots in these applications. However, the use of free-ranging robots introduces two major complexities: path planning and task scheduling [5]. This paper presents an approach to path planning and task scheduling for free-ranging robots which allows the dynamic determination of paths based on the current environment, and provides task "learning" for repetitive jobs. This approach provides a basis for the design of free-ranging robot systems which are adaptable to various environments and tasks.

The system described is being developed for applications which are best accomplished through the use of free ranging robots. Some examples of such applications are operations in the various modules of the space station, work in hazardous environments, and industrial warehouse operations. Several goals are established for this research. The system should provide for:

PRECEDING PAGE BLANK NOT FILMED

- the ability for the robot to operate in different environments, dependent only upon the existing knowledge of each environment;

- the ability to execute requested tasks based on the knowledge of spare parts', tools', and work stations' locations;

- the ability to operate in a multi-robot environment;

- the requirement of a task definition only once; and,

- the use of previously defined paths whenever possible.

The system depends upon the use of a central control computer which contains knowledge of each operating environment, and controls all modifications to that environment.

CONTROLLED OPERATING ENVIRONMENT

The system is based on the concept that all dynamics in an environment are controlled by a central computer which uses a combination of artificial intelligence techniques and classical algorithms. In the current project, the environment is defined initially by the user; subsequent changes to the environment are performed by, and "remembered" by, the centrally controlled robots within the environment. All tasks and operations (including inputs to and outputs from the environment) are scheduled and planned before execution. The central computer defines the shortest path between the starting point and ending point of a task, monitors the location and use of all items required to complete a task, and coordinates the task with other robots within the environment.

The controlled environment also contains storage sites (for inventory, tools, spare parts, etc.), work stations (where the task is actually performed), and home bases (where the robot are be recharged when as necessary). The central computer maintains all necessary information and knowledge on storage sites, work stations and home bases. Such information includes location parameters, content status, idle/busy status, etc., and appropriate updates are made based on the dynamics of the environment.

The control system described in the paper has been developed on a Symbolics 3620 AI work station. The programs are written in Common Lisp but also utilize the Windows and Flavors packages available on the Symbolics, which provide an object oriented programming base. Each work station, storage site, robot, path, and task is represented as an object, with particular characteristics associated with each. In order to monitor robot operations within the environment, the current program graphically simulates the results of the path planning and scheduling routines performed when a task request is processed.

PATH PLANNING AND LEARNING

Several researchers have investigated path planning in various environments. Trivedi, Gonzalez, and Abidi worked on robotics control for hazardous environments [6]. Path planning for a manipulated arm robot has been researched by Jentsch [4]. Taghaboni and Tanchoco have investigated path planning of free ranging robots [5]. Their work is based on selecting the best path from known paths depending on the path providing the shortest time to completion [5]. The system developed by Kaiser and Hawkins deals with the control of free-flying robots, and selects a path based on the cost function [3]. Weisbin has described work in path planning, as well as learning concepts, for autonomous robots [7]. The system described here draws from the above mentioned research, but takes a somewhat different approach. Not only is collision avoidance used in path planning, but the knowledge of item locations allows the robot to choose where it needs to go to obtain tools and spare parts, and where the work is to be performed. Knowing this, the most efficient path for completing an entire task can be determined.

Tasks are given to the AI control center by the user. The user specified task information is stored in a relational data/knowledge-base. Tasks, therefore, need to be defined only once. Thereafter, user specification of the task name will allow the control system to access information and knowledge required for task completion. As more and more tasks are defined, the power of the system increases and the amount of task information required from the user decreases.

Once a task is defined, the central computer searches the data/knowledge bases for the required item information. The movements of the robot to acquire the items and perform the desired operations are then planned, including collision avoidance of static objects and other moving robots. The robot movement is scheduled via a recursive routine which locates any obstructions in the direct path between two points and determines the shortest path around those obstructions. It then checks for obstructions in that path, until a clear path between the two locations is found. Once the task has been planned and scheduled, the control sequence is sent to the robot for task performance. The control computer can also be used to define actions to be taken by the robot upon task completion.

When a path is determined, any other tasks requiring movement between the same points can use the previously determined path, without requiring recalculation by the recursive functions. However, if the environment has changed (objects added or moved) the recursive path finding routine is called and the new path subsequently stored for future use. Even when path modifications are made for a given task, the items and operations required to complete that task are "remembered" by the control computer from the initial definition of the task.

The pre-performance planning routine also allows for collision avoidance between robots in a multiple robot environment. Since all tasks and paths are determined prior to execution, the central computer knows where each robot is going to be at a given time and can use this information to determine the path for a new task. The path may include avoiding other robots in the environment by altering the path or simply waiting at a given location for another robot to pass.

LIMITATIONS AND ADVANTAGES

Among the advantages of this approach over other approaches being investigated are the increased flexibility of the system and the reduced overall CPU time. The increased flexibility is provided by: (1) shortest path determinations based on the current environment, (2) multi-robot operations within the environment, and (3) path modifications when required by the dynamics within the environment. The reduced CPU time results from not only the removal of continuous collision avoidance requirements but also the task and path "learning" capabilities of the control computer.

The major future enhancements to this approach should include: (1) developing the capability for a robot to assist in defining the initial environment via sensory enhancements; (2) providing the robot with the ability to search for a clear path through or around closely grouped objects; and (3) integrating classical algorithms for the performance of routine tasks once a robot has been placed at a work station. Each of these areas are being examined by current research and development efforts [1][2][6].

CONCLUSIONS

This paper presents an alternative solution to the problems of path planning and task scheduling for free-ranging robots. The approach is not limited to pre-defined paths and intersection nodes, but rather defines each individual path based on the current environment. The use of artificial intelligence techniques allows the control computer to monitor and modify the dynamic environment and to plan and schedule tasks accordingly. Complete development of this approach will allow the continued advancement of applications for free-ranging robots.

REFERENCES

1. Hopkins, S.H., and P.J. Drazan, "Semiautonomous Systems in Automatic Assembly," IEEE Proceedings, vol. 132, Part D, No. 4, July 1985, pp. 174-177

2. Huang, C.L., and J.T. Tou, "Automatic Generation of 3-D Pictorial Drawing from Intensity Image," Proceedings of SPIE Conference: Applications of Artificial Intelligence V, May 1987, p. 502

3. Kaiser, Donald Leo, and P.J. Hawkins, "Motion Planning for a Free-Flying Robot," Proceedings of NASA Conference on Artificial Intelligence for Space Applications, November 1986, pp. 247-255

4. Jentsch, Winfried, "Off-line Planning of Collision-free Trajectories and Object Grasping for Manipulating Robots," Proceedings of the Third Annual Conference on Artificial Intelligence and Information-Control Systems of Robots, June 1874, pp. 193-196

5. Taghaboni, F. and J.M.A. Tanchoco, "Dynamic Scheduling and Control of Free-Ranging Automated Guided Vehicle Systems," Proceedings of the 1986 Fall Industrial Engineering Conference, December 1986, pp. 127-129

6. Trivedi, M.M., R.C. Gonzalez, and M.A. Abidi, "Developing Sensor-driven Robots for Hazardous Environments," Proceedings of SPIE Conference: Applications of Artificial Intelligence V, Vol. 786 May 1987, pp. 185-188

7. Weisbin, Charles R., "Intelligent-Machine Research at CESAR," AI Magazine, Vol. 8, No. 1, Spring 1987, pp. 62-74