AUTONOMOUS GUIDANCE AND CONTROL OF A ROVING ROBOT*

J. A. Miller
Guidance and Control Section
Jet Propulsion Laboratory
Pasadena, California 91103

Introduction

NASA has embarked upon a Robotics Research Project at the Jet Propulsion Laboratory, the purpose of which is to establish a technology base in robotics and semi-autonomous control of unmanned machines or vehicles to support lunar and planetary exploration. The long term objective of the robotics work at JPL is to provide an integrated hardware/software system which is capable of nearly autonomous performance. The present design philosophy seeks to limit the need for human interaction to selection of goals and similar higher level functions. The Rover must then analyze the scene for traversability, generate a planned path to the goal, and follow that path, avoiding any obstacles along the route. We have achieved the capability of performing the scenario just described in a simplified environment; a laboratory with a flat surface, a limited number of obstacles and constant illumination. This paper focuses upon the autonomous guidance and control functions of the Rover which executes the planned trajectory in an incompletely defined environment.

The Robot hardware consists of a flatbed four wheeled roving vehicle upon which is mounted a manipulator, a scan platform with stereo TV cameras and a laser rangefinder, and associated electronics. The vehicle is steered by an Ackerman type double steering system with position servo loops. The four wheels are driven independently by DC torque motors. The present navigation hardware consists of a conventional gyrocompass and digital optical encoder odometers combined to form a dead-reckoning navigation system. The robot is controlled by a real-time local minicomputer (General Automation SPC-16) which is linked to a large remote time-shared computer (Decsystem 10).

The Rover Software System consists of an integrated set of subsystems which include the Prototype Ground System (PGS), the Rover Executive (REX) and Vision, Manipulation and Locomotion Subsystems. The Guidance System consists of those elements of the Locomotion Subsystem software whose primary purpose is to plan and control activation of steering and drive motors such as to cause the Rover to follow a planned p'ath within a predefined error tolerance in an efficient manner. The planned path is provided to the Guidance System in real-time by the Path Planning Module, another element of the Locomotion Subsystem software. For further discussion of the Rover Software System in general and the path planning module In particular see Thompson (1977).

Design Goals

The primary design goals of the guidance system were vehicle safety, subsystem autonomy and operational efficiency. These goals are typical of mission related priorities. Accuracy of performance is not listed as a separate design goal since it is implied by both the need for vehicle safety and subsystem autonomy. Subsystem autonomy refers to the guidance system's ability to detect and correct errors in its own performance and detect and avoid any obstacles which might be encountered without requiring assistance from other Rover processes (i.e. without requesting a new plan from the Path Planning Module). The need for operational efficiency refers to the desire to minimize the required time to complete a traverse (maximize vehicle average traverse speed). Increased speed and autonomy combine to significantly reduce mission costs. Reliability is also an important goal implied by the needs for safety and autonomy. Minimizing vehicle acceleration increases hardware reliability but conflicts with the desire to maximize average speed.

In order to attain these goals, real-time maneuver planning, performance monitoring and error correction are required. Maneuver planning must be done in real-time to integrate turning, error correction and obstacle avoidance maneuvers such that a favorable balance of speed, acceleration and deviation from the planned path is achieved. The following paragraphs describe important aspects of maneuver planning in greater detail.

Maneuver Planning

Path links provided by the Path Planning Module (Thompson, 1977) are defined by a constant minimum radius turn followed by a straight segment. To follow such a path with minimum deviation would require stopping the Rover during steering, since steering requires appreciable time and the turn radius varies during steering. Such a method of operation is rejected, since it reduces the average speed, and places greater stress upon the steering system with a resultant decrease in reliability and lifetime of the steering drive hardware. Steering while moving, however, results in a variable radius turn, forcing the Rover to deviate from the prescribed path. The magnitude of the deviation and its distribution along the planned path depends upon the vehicle speed during steering, the magnitude of the desired heading change and the points of initiation and conclusion of steering. The deviation may be minimized for a given steering speed by designing the maneuvers such that the vehicle begins steering prior to the prescribed turning point and concludes steering an equal distance beyond the prescribed point of completion. This not only minimizes the deviation during the turn but also theoretically eliminates any transverse error at completion of the turning maneuvers.

Selection of a speed while steering becomes a critical parameter in maneuver design since this determines the theoretical magnitude of the deviation as well as the distance required before and after the planned turn. The speed while steering is thus constrained by the maximum transverse deviation

---

*This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS 7-100, sponsored by the National Aeronautics and Space Administration.

allowed and the lengths of the straight segments between turns. The distance required prior to the turn consumes a portion of the straight segment of the previous link. Similarly, some portion of the straight segment of the present link must be reserved for the turn of the subsequent link (undefined at this point). Additional portions must be reserved for transverse error corrections and obstacle avoidance maneuvers, should these be necessary. Real-time updates to vehicle performance and hazard expectancy models are needed to allow for such contingencies.

Guidance Software

The structure of the controlling software evolved as a result of timing, core usage and CPU availability constraints. Other robot processes require memory and CPU time to operate in pseudo-parallel to the guidance software, forcing the latter to operate discontinuously even while the vehicle is moving. To meet these constraints, the main guidance module was designed around a one-second cycle time, with a half second delay. The one second cycle was chosen to provide a fast response capability to incoming commands (a safety feature) and satisfies the need to perform position updates at least once per second. To reduce core requirements, maneuver monitoring and controlling was compressed into two general purpose sections, one for controlling all turning maneuvers, and another for monitoring movement along straight segments. The cyclical structure of the program and the general purpose design of the turn/straight path monitors requires that a complete task-relative Rover state be dynamically defined so that at any point of any cycle the program "knows" what the Rover is doing, what has already been done, what must be done in this cycle, and what it must prepare to do in subsequent cycles, all with a minimum "recognition effort". Appropriate setting and testing of task/Rover state variables is crucial to Guidance System performance.

Delays in maneuver execution due to the cyclical inactivity of the guidance software could cause significant deviations from the designed maneuvers. There also exists a hardware constraint that vehicle drive motors be pulsed frequently to maintain motion (another safety feature). Provision of separate high priority MOVE and STEER programs activated on a clocked interrupt basis by the Real Time Operating System (RTOS) resolves these problems by permitting precise timing of hardware control signals even while the main guidance routine is inactive. The main guidance routine controls these latter processes by scheduling their activity through the RTOS and passing arguments through COMMON. This structure enables the guidance software to meet the efficiency, responsivity and flexibility requirements.

Conclusions

The system described here has been implemented in software and in hardware. A detailed evaluation of system performance has not been conducted since implementation of transverse error correction capabilities is incomplete as of this writing. The path following capability has been successfully demonstrated on numerous occasions, however. Longitudinal errors of 1% and (as yet) uncorrected transverse errors of less than 10% of the turn arc length have been achieved, with final orientation errors of about 1 degree. The somewhat large transverse error is attributed to the steering hardware. Completion of the transverse error correction capability should reduce the transverse errors by a factor of 5 or greater.

References

1. Thompson, A. M., "The Navigation System of the JPL Robot". These proceedings.