Obstacles were detected, and paths planned at speeds in excess of 4.25 m/s. However, the ability of the vehicle's tracker and controller to accurately follow these paths prevented the vehicle from actually driving around obstacles while maintaining this speed. By all indications, an improved controller will enable us to achieve our theoretical maximum speed. Fig. 8 shows a map, vehicle position, and a path found around a clearly inadmissable region.

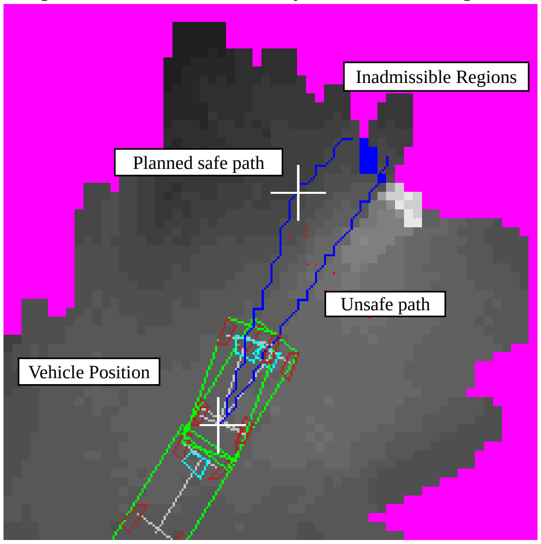

Fig. 8. Example of rough terrain obstacle avoidance

The rapid stop command is not yet implemented, thus testing of vehicle stop in the event of a planning impasse was precluded.

Most natural obstacles found in the testing environment required no more than 4 control points to define a path around them. A total of 4.5km of rough terrain was traversed during the latest series of test runs.

CONCLUSIONS

A system for Autonomous Cross-Country Navigation has been developed and demonstrated successfully at speeds up to 4.25 m/s on moderate terrain. This perception and planning system are the first to support these speeds on natural barren terrain, to integrate a generate and test planning paradigm with simulated traversal of paths for ACCN, to include vehicle dynamics as part of the simulation process, and to account for the increased difficulty of perception at high speed on natural terrain.

The major difficulties encountered were limited sensor horizon and the limited response time of the vehicle. A longer sensor horizon would permit more time for planing, and would allow the system to make direction changes around obstacles more quickly. Better vehicle response would allow the system to successfully follow these planned paths. It should be noted that the vehicle is physically capable of following the planned paths, but that, in some cases, the controller lacks the needed response characteristics.

Future work will allow the system to travel along more tightly constrained paths at higher speeds. Sensor upgrades will permit the longer sensor horizon needed to achieve these goals.

Other future work will include feedback for active sensor pointing, planning for cluttered environments and more complete models of the vehicle's capabilities to permit more robust navigation

ACKNOWLEDGEMENTS

As well as the authors, the software development team has included Omead Amidi, Mike Blackwell, William Burky, Martial Hebert, Jay Gowdy, Annibal Ollero, Dong Hun Shin and Jay West. The authors thank Dr. Behnam Motazed and the NavLab II design and retrofit team for producing a capable ACCN testbed.

This research was sponsored by the Defense Advance Research Projects Agency, through ARPA Order 7557 (Robot System Testing), and monitored by the Tank Automotive Command under contract DAAE07-90-C-R059.

REFERENCES

1. Amidi, O. Integrated mobile robot control. Robotics Institute Technical Report CMU-RI-TR-90-17, Carnegie Mellon University, Pittsburgh.

2 Bhatt, R; Venetsky, L., Gaw, D., Lowing, and D., Meystel, A. (1987). A real-time pilot for an autonomous robot. Proc. of IEEE Conf. on Intelligent Control, 135-139.

3 Daily, M. (1988). Autonomous cross country navigation with the ALV. Proc. IEEE Internat. Conf. on Robotics & Automation, 4, 718-726.

4. Dickmanns, E.D. (1990). Dynamic computer vision for mobile robot control. Proc. of the 19th Internat. Symposium and Exposition on Robots, 314-27.

5. Feng, D. (1989). "Satisficing feedback strategies for local navigation of autonomous mobile robots." Doctoral Thesis, Electrical and Computer Eng. Dept., Carnegie Mellon University, Pittsburgh.

6. Lozano-Perez, T. and Wesley, M. A. (1979). An algorithm for planning collision free paths among polyhedral obstacles. Communications of the ACM, 22:10, 560-570.

7. Moravec, H. P. (1983). The Stanford cart and the CMU Rover. Proc. of the IEEE, 71, 872-884.

8. Olin, K. E. and Tseng, D. (1991). Autonomous cross country navigation. IEEE Expert, 8, 16-30.

9. Pomerleau, D.A. (1991). Efficient training of artificial neural networks for autonomous navigation. Neural Computation, 3:1, 88-97.

10. Chang, T.S., Qui, K., and Nitao, J. J. (1986). An obstacle avoidance algorithm for an autonomous land vehicle. SPIE Conf. Proc, 727, 117-123.

11. Shin, D. H., Singh, S., and Shi, W. (1991) A partitioned control scheme for mobile robot path planning. Proc. IEEE Conf. on Systems Eng.

12. Thompson, A. M. (1987) The navigation system of the JPL robot, IJCAI, 749-757.

13. Wilcox, B. and others. (1987). A vision system for a Mars rover. SPIE Mobile Robots II, 172-179.

- Locomotion support; the motion of the vehicle must not be geometrically impeded by the presence of insurmountable obstacles in front of the wheels.
- Body collision; the vehicle cannot occupy the same space as its surrounding terrain.
- Unknown terrain; the vehicle must not be situated over unknown terrain, (off the map)

Vehicle statics are included with the of a kinematic admissability test. It is intuitive and straightforward to define static admissability as follows.

A way point is Statically Admissable if the vehicle is in a state of static equilibrium at the point.

Thus, the planner is conservative in that the vehicle is required to be statically stable at every point along the path, even though there may be some executable paths that do not meet this requirement.

Control point selection. If an inadmissable way point is found while traversing a local path in simulation, it can be assumed that an inadmissable region is located near the front of the vehicle at this way point. This inadmissible point is added as a candidate control point to the list of control points (the control path) used to generate the trajectory.

Once the inadmissable point is found, it is moved around the edge of the obstacle in successive sideways steps, on each step trying to proceed toward the next control point. When a step straight forward is found to be is admissable, it is assumed that the edge of the obstacle has been found. (See Fig. 6.)

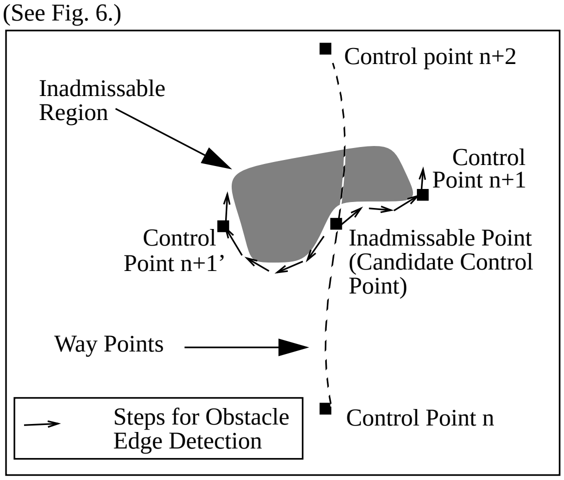

Fig. 6. Example selection of new control points

This procedure can be applied recursively between any two control points when an inadmissable way point is found. In this example (Fig. 7), two possible paths are found. In this way, an inadmissible point is moved around the obstacle, and becomes a control point which indicates the probable edge of an inadmissable region. Because the control point searching is done by continuously moving toward the edges of the known terrain during each search step, termination is guaranteed, as the point cannot be moved past the edge of the known map.

Spatial Search Failure

There are two types of search failure. First, a traversable path may not be calculated before the vehicle enters the stopping segment of the current path. In this case, the vehicle stops, drives backward to the last point in the intended region, and then drives over the next path. Second, the planner may not be able to find a path through the region, possibly due to an impasse. In this case, the current system will terminate the run after the vehicle comes to a stop on the current path.

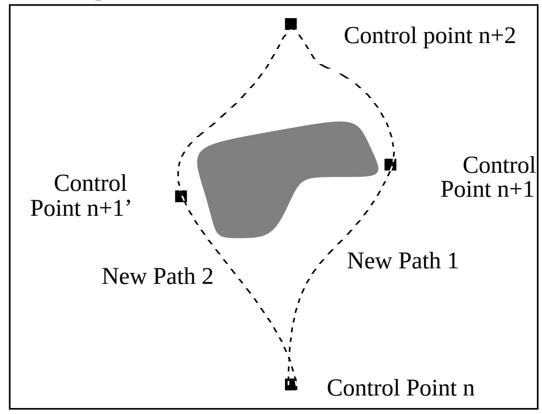

Fig. 7. New paths with obstacle avoided

Temporal Path Generation

Once a spatial path is planned, it is necessary to specify vehicle velocities at each way point along that path. The high speed environment in which the vehicle operates necessitates the use of dynamic constraints on the vehicle’s trajectory.

The velocity profile generator must calculate a set of maximum velocities along this path that ensures that the vehicle will not exceed dynamic limits, thus making the path dynamically admissable to execution. We define dynamic admissability in terms of a set of orthogonal accelerations called performance limits. These acceleration correspond to the vehicle’s lateral, parallel and vertical directions.

A path is considered to be *dynamically admissable* if the accelerations generated during motion do not exceed the vehicular dynamic performance limits.

A *performance limit* describes a maximum permissible vehicle acceleration; it does not necessarily describe a vehicle’s stability limit, but it is never greater. Any velocity profile that, at every way point, is exceeded by this generated profile is considered to be dynamically admissable.

Dynamic constraints are applied to determine the maximum permissible vehicle velocities along every point on the path. Given that the vehicle must be stopped at the last point on the stopping segment, the maximum permissible speed for the previous point can be calculated. Continuing this method of reverse state propagation gives maximum speeds to all points on the path. Permissible is described by the performance limits of the vehicle.

RESULTS

Tests of the vehicle were performed in a large open area, characterized by gently rolling terrain, sparse vegetation, rock outcroppings, and water puddles. The system was implemented with a pure pursuit (Amidi, 1990) path tracker.

Obstacle avoidance was successful at speeds up to 2.0 m/s.

each spatial point along the planned path.

Planning Configuration

The goal of the planner is to extend the known safe path for the vehicle by planning a path across recently sensed terrain. Fig. 4 illustrates the geometric constraints on the planning problem.

The invariant in the planner is that the *next path* is planned while the *current path*. is traversed. Each next path corresponds to a path across a map generated from an image taken at the first point on the current path.

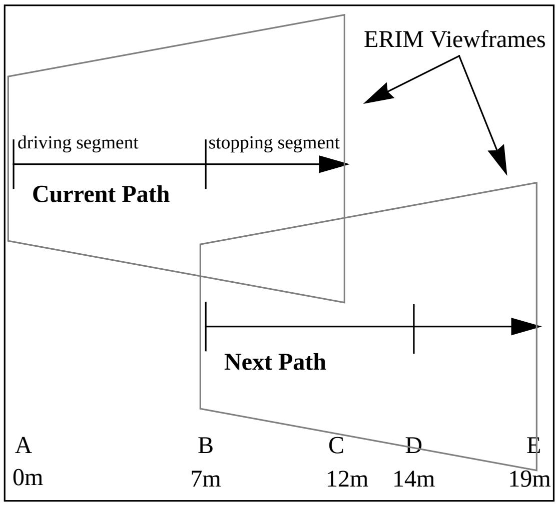

Fig. 4. Geometry for planning

Each path has a *driving segment* and a *stopping segment*. Consider the current path: at the beginning of the driving segment, the vehicle takes an image; as the vehicle drives along this segment, the system plans the next path; if an impasse is reached, and planning fails, the vehicle will drive along the stopping segment, and come to a stop, never having left known terrain. In Fig. 4, an image is taken at A, and path BE is planned while driving AB. If planning fails, BC is driven. Under normal operation, no stopping segment is ever driven, and the vehicle is in continuous motion.

This algorithm provides several guarantees. First, the vehicle can operate in an uncertain environment since the planned path can change radically when unknown obstacles are introduced into the terrain map. Secondly, the system ensures vehicle safety in case of a planning failure, which is essential for high speed driving in rough terrain.

Spatial Trajectory Generation

In the first stage of planning, a generate and test paradigm is used to choose and evaluate successive spatial paths until a safe path which traverses the known terrain is found. This method avoids the problem of preprocessing the terrain to find all unsafe regions. The generate and test method also permits the planner to compute any path needed without limiting the search to a precomputed set.

The planner begins initially with a *global path* describing a course route across a long stretch of terrain. During every sense-plan-drive cycle, the planner tries to move the vehicle along the global path, by choosing *control points.* Control points are points which define the ends of the local path, or which may lead the vehicle around an obstacle. A set of control points which will be used to generate a local path is called a *control path*. Initially, each planning cycle for the next path begins with a control path having two control points: a point from the current path for continuity, and a point a fixed distance ahead to move the vehicle toward the global path.

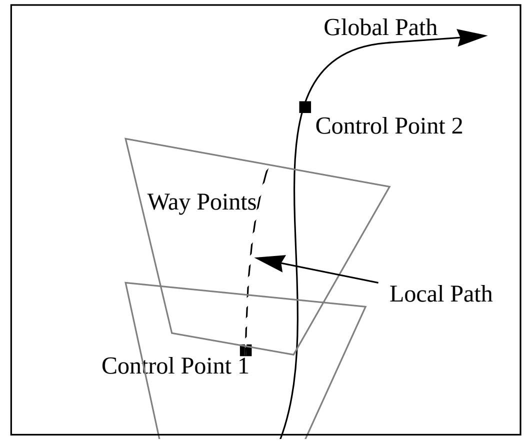

Fig. 5. Global, control, and way points

When a control path is selected for testing, a smooth spatial curve is fit to the control points, and closely-spaced *way points* are generated along the entire length of the curve. Fig. 5 illustrates these concepts. In this figure, Control Point 1 and Control Point 2 comprise the first control path.

The vehicle traverses the way points in simulation. When a way point is tested and found to be unsafe, up to two new control paths are generated. Each new control path will contain an additional control point found by the planner that is intended to modify the path so as to direct the vehicle around either side of the obstacle encountered. These new control paths are added to a list of control paths. The planner then selects a new control path from this list based on the estimated cost (as a function of path length and complexity) of traversing the path. This process repeats until a safe path is found. In theory, this algorithm should be capable of generating traversable trajectories through arbitrarily narrow admissable regions.

In the sections below, the test used to determine safety as well as the algorithm for identifying control points are described.

Kinematic admissibility. Kinematic admissability implies that the geometry of the path during vehicular motion will permit traversal. Traversal can be impeded by an obstacle, such as a tree or a rock, or by vehicle geometry, such as a minimum turning radius. Kinematic admissability may be defined as follows:

A way point is kinematically admissable if there is no geometric cause to impede the proposed motion of the body and if the vehicle and its environment do not occupy common space.

This definition is embodied in a set of four constraints, each of which is evaluated at a given way point to determine kinematic admissability. A waypoint is admissable only if it satisfies all four constraints:

- Minimum turning radius; the curvature of the path at the way point cannot exceed bounds given by the vehicle’s minimum turning radius.

A special edge detector is used to find the ambiguity edge near the top of the range image and all information beyond it is ignored. A median filter is used to remove the outliers associated with regions of high terrain texture. No special treatment of specular regions (e.g. water puddles) was required since they are treated automatically as range shadows by the rest of the system.

Vehicle Motion

Clearly, the transform of coordinates from the sensor frame to the world frame requires knowledge of the vehicle pose when each range pixel was measured. The sensor scanning mechanism requires about 1/2 second to complete a single scan. Hence, at higher vehicle speeds, the use of a single vehicle pose for the entire image will give rise to distortions in the map.

The motion distortion removal function uses a series of vehicle poses which were stored by a synchronous pose logger during image digitization. Currently, 8 poses are associated with each image, and, for each pixel, the function uses the pose that was measured closest in time to the instant when the pixel was measured. In this way, an obstacle that would have otherwise been elongated by 2 meters at current speeds is represented accurately in the map.

Terrain Map Generation

The perception system generates a uniformly sampled grid data structure called a cartesian elevation map (Olin, 1991) which stores in each grid cell the elevation of the corresponding point in the environment. The generation of this data structure requires more than the trivial transformation of coordinates from the sensor frame to the world frame. Since one regularly sampled structure (the image) is converted into another (the map), some nontrivial issues rooted in the sampling theorem must be addressed.

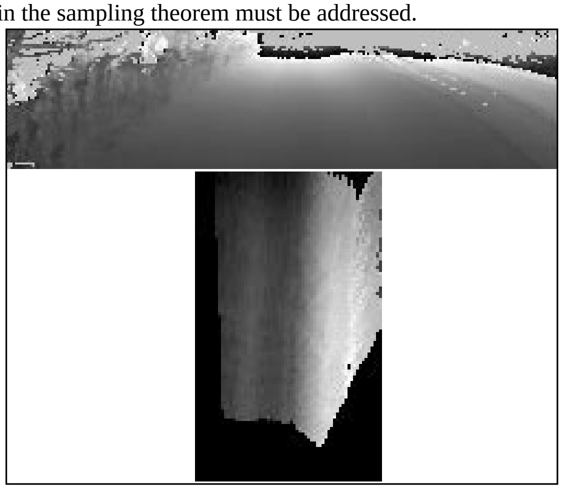

Fig. 3. Typical range image and terrain map

The terrain map generation problem is the inverse of the hidden surface removal problem of computer graphics. The sensor intrinsically removes hidden surfaces since the laser beam is reflected from the closest surface along its trajectory but cannot penetrate to be reflected from any surfaces beyond it. Hence, we must accept that some regions in the elevation map will contain no information. Figure 3 presents a single range image and the portion of the terrain map that results from it, as seen from above.

Clearly, this problem is caused by the sensor geometry. With the sensor attached to the vehicle, the ideal overhead viewpoint is not possible, yet the planner needs to see behind obstacles in order to plan optimally. The sensor configuration also causes regions to be oversampled close to the vehicle and undersampled far away. To some degree, the problem is overcome by fusing the results of several range images from different vantage points into a single map.

The elevation accumulation function computes the average and standard deviation of elevations encountered for oversampled grid points. Range shadows and undersampled regions are filled using a special interpolation algorithm that marks usually large unknown regions as range shadows and records an upper bound on elevation for the grids cells contained in them.

PLANNING SUBSYSTEM

Path planning is defined as finding a path through space from a start point to a goal point while avoiding obstacles. Much of the previous work in this area (Lozano-Perez, 1979), addresses path generation for a polygonal body through a space containing polygonal obstacles. This approach assumes that all obstacles in the traversable space are known. This approach, and other approaches (Thompson, 1987; Feng, 1989) do not satisfy the requirements for ACCN, for the following reasons:

- Polygonal obstacle assumption is not appropriate for rough terrain, since the motion of an obstacle can include any pose (such a steep terrain) that is unsafe. This set of poses can be computationally prohibitive to precompute in it's entirety and may not assume a polygonal shape.
- A limited sensor horizon precludes knowledge of entire search space before planning
- The intrinsic limitations of the vehicle's ability to follow the given path are not considered (e.g. minimum turning radius.)

Work by Olin (1991) and Daily (1988), relaxes the flat world, polygonal obstacle assumption and introduces the notion of simulated traversal of paths, taking into consideration kinematic constraints at discrete points along these paths. Their system, however, limits the space of potential paths by only considering a static set of paths across the sensed area. This methodology does not allow small deviations around obstacles, and prevents more than a single change of direction within a planning cycle. This planner might find a region uncrossable simply because its static space of predetermined paths is insufficient to include a safe path. Furthermore, this system does not consider dynamics, as it operated in a speed region where dynamic effects can be neglected.

The trajectory planning software on the Navlab II considers both kinematic and dynamic constraints on the vehicle, plans paths through tight spaces, copes with a changing sensor horizon, and accounts for the intrinsic ability of the vehicle to track a given path.

The system first performs spatial searching based on kinematic constraints and then applies dynamic constraints to set the temporal portion of the path, that is, the speed at

As a result, sophisticated planning is needed. Hughes' ALV has reached speeds of 3.5 km/hr (Daily, 1988; Olin, 1991) in this manner. However, the speed of the ALV system is small enough that many concerns for vehicle safety are diminished, and dynamics need not be modelled for the planning process.

This paper presents an overview of the software and hardware system which addresses the eight problems introduced above, thus permitting ACCN in natural terrain at speeds higher than that of previous systems.

SYSTEM OVERVIEW

The testbed for the system is the NavLab II, a computer-controlled HMMWV (High Mobility Multi-purpose Wheeled Vehicle). The equipment on the NavLab II can be divided into three categories: computation, sensing, and actuation. For computation, there are 3 Sparcstation II's connected by ethernet. For sensing, an ERIM, a 2-d raster scanning laser range finder, is used. The vehicle is totally self-contained, as there no provisions for off-vehicle communications or power supply while in the field. Power is supplied by two onboard generators.

The NavLab II has a central integrated motion controller; its main function is to coordinate control of the vehicles motion control axes to effectively control the vehicle as a whole. The NavLab II has three independent control axes: the throttle, the brake and the steering wheel. The controller determines the vehicle's state using a combination of encoders (mounted on the transmission output and the engine) and an inertial navigation system. Real time software, running on two processors, coordinates the control of the independent axes, keeps track of the vehicle's state, and handles motion commands (steering and speed) from the navigation computers.

In order to traverse unknown terrain, the navigation software must first sense the terrain in front of the vehicle, plan a trajectory across this terrain map, and then drive along the path. This process cycles as the vehicle drives.

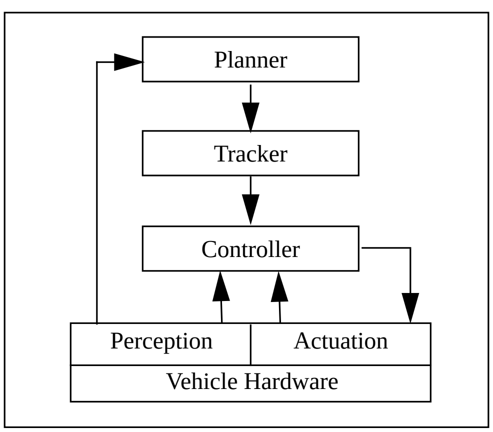

Fig. 1. Software/hardware system interaction

The sensing consists of taking a range image from the ERIM, and transforming the data into a 2.5D discretized terrain map. This operation requires approximately 0.5 seconds.

Path planning is comprised of a spatial planning phase, where a path is generated across the terrain map through heuristic modification to avoid obstacles, and a temporal planning phase, where each point along the path is assigned a speed by considering vehicle dynamics. The separation of dynamics and kinematics makes the planning problem tractable. Computation time for planning is about 0.5 seconds.

The planned path is passed to the path tracker, which tracks the path by issuing commands to the vehicle controller. These components are illustrated in the software system design (Fig. 1). Previous work by Amidi (1990) thoroughly addresses issues involving the tracker and controller.

PERCEPTION SUBSYSTEM

The goal of the perception system for the autonomous navigator is to generate a description of the terrain geometry that is convenient for the path planning subsystem to use, and will accommodate both higher vehicle speeds and rougher terrain than has been achieved in previous systems.

Fig. 2 presents a conceptual view of the software architecture of the perception system:

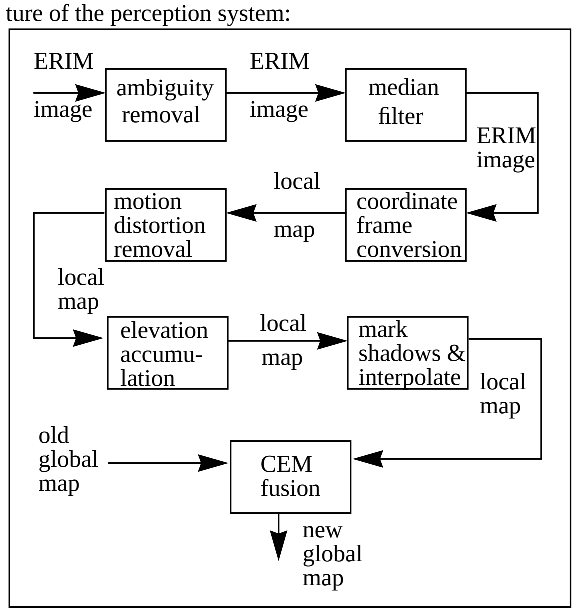

Fig. 2. Perception software architecture

The ambiguity removal and median filter modules remove noise and inaccuracies from the input image. Coordinate frame conversion converts the range information into elevation information. While this is being performed, the motion distortion removal module accounts for vehicle motion to correct the data. The elevation accumulation and shadow marking modules remove holes in the map and compute variation statistics when information overlaps. The Cartesian elevation map (CEM) fusion module is responsible for adding new elevation data to the global map, maintaining a scrolling window into the map which follows the vehicle, and estimating the vehicle's z coordinate by analyzing overlapping map data.

Sensor Limitations

The data generated by the ERIM sensor is subject to limitations of ambiguity of the range measurement beyond the laser modulation wavelength, complete lack of information when the beam is reflected off a specular region of the terrain, and degraded accuracy as range is increased or in regions of high texture.

A SYSTEM FOR AUTONOMOUS CROSS-COUNTRY NAVIGATION

Barry L. Brumitt, R. Craig Coulter, Alonzo Kelly, Anthony Stentz
Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa. 15213

Abstract *Autonomous Cross-Country Navigation requires a system which can support a rapid traverse across challenging terrain while maintaining vehicle safety. This work describes a system for autonomous cross country navigation as implemented on the NavLab II, a computer-controlled off-road vehicle at Carnegie Mellon. The navigation software is discussed. The perception subsystem constructs digital maps of the terrain from range sensor data in real time. The planning subsystem uses a generate-and-test scheme to find safe trajectories through the map, and validates them through simulated driving. The planning process considers both kinematic and dynamic constraints on vehicle motion. The system was successful in achieving 300m autonomous runs on moderate terrain at speeds up to 4.25 m/s.*

Keywords *Autonomous Mobile Robots, Obstacle Avoidance, Navigation, Image Processing.*

INTRODUCTION

Autonomous Cross-Country Navigation (ACCN) can be applied to tasks such as military reconnaissance and logistics, medical search and rescue, hazardous waste site study and characterization, as well as industrial applications, including automated excavation, construction and mining. The need to safely travel at a reasonable velocity on terrain with unknown obstacles is common to all these endeavors.

This paper addresses robot perception, planning and control required to support the fastest possible autonomous navigation on rough terrain. This objective poses new issues not encountered in slower navigation systems on flat terrain.

- Uncertain environment: sensing must be done simultaneously with driving; a precomputed path is impossible.
- Vehicle safety: if computing fails, or a safe trajectory cannot be found, the vehicle must be brought to a stop to avoid collision.
- Computation time: as vehicle speed increases, perception and planning must be performed more rapidly.
- Rugged terrain: the geometry of the terrain is sufficiently complex that the assumption of flat terrain with sparse obstacles does not suffice.
- Dynamics: the vehicle's speed is large enough that dynamics as well as kinematics must be modelled.
- Imaging geometry: the complexity of map generation increases in the context of rougher terrain.
- Sensor limitations: at higher speeds, technological limitations of the laser range finder are significant.
- Motion during digitization: given higher operating speeds, image distortion is an issue.

Early work in outdoor navigation includes the Stanford Cart (Moravec, 1983). This system drove in a stop-and-go manner at a slow speeds, modelling terrain as flat with sparse obstacles.

Higher speeds have been achieved in system operating in a constrained environment. Road following systems assume benign and predictable terrain. Obstacles are assumed to be sparse, so that bringing the vehicle to a stop rather than driving around obstacles is a reasonable strategy. In this way, speeds of up to 100 km/hr. have been achieved in road following (Dickmanns, 1990; Pomerleau, 1991) using cameras, and speeds up to 30 km/hr. have been achieved on smooth terrain with sparse obstacles using a single scanline lidar (Shin, 1991). While these systems consider problems of dynamics and vehicle safety, they do not provide the capability to drive around obstacles.

Obstacle avoidance has been achieved on approximately flat natural terrain with discrete obstacles (Feng, 1989; Bhatt, 1987; Chang, 1986.) On a smooth gently sloping streambed with sparse obstacles, the JPL rover has achieved a performance of 100 meters in 8 hours (Wilcox, 1987)

When navigating over rough natural terrain, no assumptions about the shape of the terrain ahead can be made. Obstacles may not only be discrete objects in the environment, but can correspond to unsafe configurations as well.

Obstacles were detected, and paths planned at speeds in excess of 4.25 m/s. However, the ability of the vehicle's tracker and controller to accurately follow these paths prevented the vehicle from actually driving around obstacles while maintaining this speed. By all indications, an improved controller will enable us to achieve our theoretical maximum speed. Fig. 8 shows a map, vehicle position, and a path found around a clearly inadmissable region.

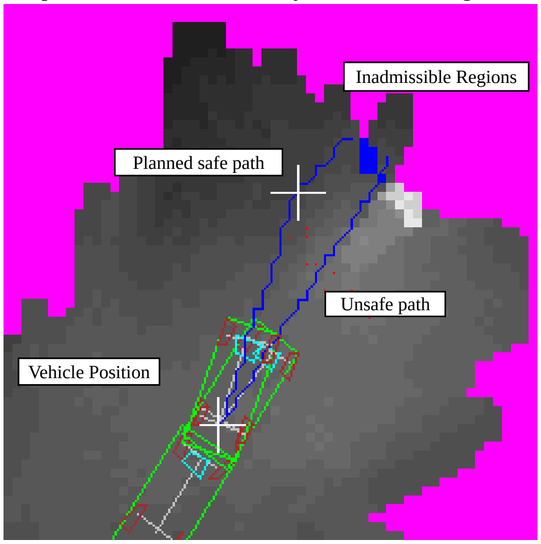

Fig. 8. Example of rough terrain obstacle avoidance

The rapid stop command is not yet implemented, thus testing of vehicle stop in the event of a planning impasse was precluded.

Most natural obstacles found in the testing environment required no more than 4 control points to define a path around them. A total of 4.5km of rough terrain was traversed during the latest series of test runs.

CONCLUSIONS

A system for Autonomous Cross-Country Navigation has been developed and demonstrated successfully at speeds up to 4.25 m/s on moderate terrain. This perception and planning system are the first to support these speeds on natural barren terrain, to integrate a generate and test planning paradigm with simulated traversal of paths for ACCN, to include vehicle dynamics as part of the simulation process, and to account for the increased difficulty of perception at high speed on natural terrain.

The major difficulties encountered were limited sensor horizon and the limited response time of the vehicle. A longer sensor horizon would permit more time for planing, and would allow the system to make direction changes around obstacles more quickly. Better vehicle response would allow the system to successfully follow these planned paths. It should be noted that the vehicle is physically capable of following the planned paths, but that, in some cases, the controller lacks the needed response characteristics.

Future work will allow the system to travel along more tightly constrained paths at higher speeds. Sensor upgrades will permit the longer sensor horizon needed to achieve these goals.

Other future work will include feedback for active sensor pointing, planning for cluttered environments and more complete models of the vehicle's capabilities to permit more robust navigation

ACKNOWLEDGEMENTS

As well as the authors, the software development team has included Omead Amidi, Mike Blackwell, William Burky, Martial Hebert, Jay Gowdy, Annibal Ollero, Dong Hun Shin and Jay West. The authors thank Dr. Behnam Motazed and the NavLab II design and retrofit team for producing a capable ACCN testbed.

This research was sponsored by the Defense Advance Research Projects Agency, through ARPA Order 7557 (Robot System Testing), and monitored by the Tank Automotive Command under contract DAAE07-90-C-R059.

REFERENCES

1. Amidi, O. Integrated mobile robot control. Robotics Institute Technical Report CMU-RI-TR-90-17, Carnegie Mellon University, Pittsburgh.

2 Bhatt, R; Venetsky, L., Gaw, D., Lowing, and D., Meystel, A. (1987). A real-time pilot for an autonomous robot. Proc. of IEEE Conf. on Intelligent Control, 135-139.

3 Daily, M. (1988). Autonomous cross country navigation with the ALV. Proc. IEEE Internat. Conf. on Robotics & Automation, 4, 718-726.

4. Dickmanns, E.D. (1990). Dynamic computer vision for mobile robot control. Proc. of the 19th Internat. Symposium and Exposition on Robots, 314-27.

5. Feng, D. (1989). "Satisficing feedback strategies for local navigation of autonomous mobile robots." Doctoral Thesis, Electrical and Computer Eng. Dept., Carnegie Mellon University, Pittsburgh.

6. Lozano-Perez, T. and Wesley, M. A. (1979). An algorithm for planning collision free paths among polyhedral obstacles. Communications of the ACM, 22:10, 560-570.

7. Moravec, H. P. (1983). The Stanford cart and the CMU Rover. Proc. of the IEEE, 71, 872-884.

8. Olin, K. E. and Tseng, D. (1991). Autonomous cross country navigation. IEEE Expert, 8, 16-30.

9. Pomerleau, D.A. (1991). Efficient training of artificial neural networks for autonomous navigation. Neural Computation, 3:1, 88-97.

10. Chang, T.S., Qui, K., and Nitao, J. J. (1986). An obstacle avoidance algorithm for an autonomous land vehicle. SPIE Conf. Proc, 727, 117-123.

11. Shin, D. H., Singh, S., and Shi, W. (1991) A partitioned control scheme for mobile robot path planning. Proc. IEEE Conf. on Systems Eng.

12. Thompson, A. M. (1987) The navigation system of the JPL robot, IJCAI, 749-757.

13. Wilcox, B. and others. (1987). A vision system for a Mars rover. SPIE Mobile Robots II, 172-179.

- Locomotion support; the motion of the vehicle must not be geometrically impeded by the presence of insurmountable obstacles in front of the wheels.
- Body collision; the vehicle cannot occupy the same space as its surrounding terrain.
- Unknown terrain; the vehicle must not be situated over unknown terrain, (off the map)

Vehicle statics are included with the of a kinematic admissability test. It is intuitive and straightforward to define static admissability as follows.

A way point is Statically Admissable if the vehicle is in a state of static equilibrium at the point.

Thus, the planner is conservative in that the vehicle is required to be statically stable at every point along the path, even though there may be some executable paths that do not meet this requirement.

Control point selection. If an inadmissable way point is found while traversing a local path in simulation, it can be assumed that an inadmissable region is located near the front of the vehicle at this way point. This inadmissible point is added as a candidate control point to the list of control points (the control path) used to generate the trajectory.

Once the inadmissable point is found, it is moved around the edge of the obstacle in successive sideways steps, on each step trying to proceed toward the next control point. When a step straight forward is found to be is admissable, it is assumed that the edge of the obstacle has been found. (See Fig. 6.)

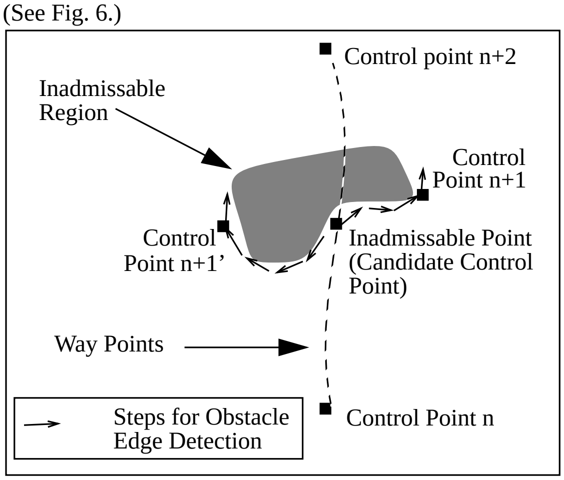

Fig. 6. Example selection of new control points

This procedure can be applied recursively between any two control points when an inadmissable way point is found. In this example (Fig. 7), two possible paths are found. In this way, an inadmissible point is moved around the obstacle, and becomes a control point which indicates the probable edge of an inadmissable region. Because the control point searching is done by continuously moving toward the edges of the known terrain during each search step, termination is guaranteed, as the point cannot be moved past the edge of the known map.

Spatial Search Failure

There are two types of search failure. First, a traversable path may not be calculated before the vehicle enters the stopping segment of the current path. In this case, the vehicle stops, drives backward to the last point in the intended region, and then drives over the next path. Second, the planner may not be able to find a path through the region, possibly due to an impasse. In this case, the current system will terminate the run after the vehicle comes to a stop on the current path.

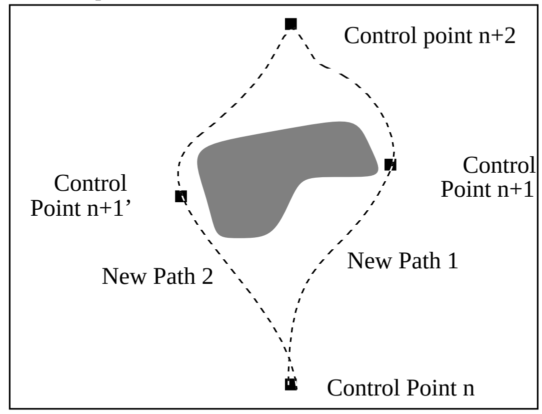

Fig. 7. New paths with obstacle avoided

Temporal Path Generation

Once a spatial path is planned, it is necessary to specify vehicle velocities at each way point along that path. The high speed environment in which the vehicle operates necessitates the use of dynamic constraints on the vehicle's trajectory.

The velocity profile generator must calculate a set of maximum velocities along this path that ensures that the vehicle will not exceed dynamic limits, thus making the path dynamically admissable to execution. We define dynamic admissability in terms of a set of orthogonal accelerations called performance limits. These acceleration correspond to the vehicle's lateral, parallel and vertical directions.

A path is considered to be *dynamically admissable* if the accelerations generated during motion do not exceed the vehicular dynamic performance limits.

A *performance limit* describes a maximum permissible vehicle acceleration; it does not necessarily describe a vehicle's stability limit, but it is never greater. Any velocity profile that, at every way point, is exceeded by this generated profile is considered to be dynamically admissable.

Dynamic constraints are applied to determine the maximum permissible vehicle velocities along every point on the path. Given that the vehicle must be stopped at the last point on the stopping segment, the maximum permissible speed for the previous point can be calculated. Continuing this method of reverse state propagation gives maximum speeds to all points on the path. Permissible is described by the performance limits of the vehicle.

RESULTS

Tests of the vehicle were performed in a large open area, characterized by gently rolling terrain, sparse vegetation, rock outcroppings, and water puddles. The system was implemented with a pure pursuit (Amidi, 1990) path tracker.

Obstacle avoidance was successful at speeds up to 2.0 m/s.

each spatial point along the planned path.

Planning Configuration

The goal of the planner is to extend the known safe path for the vehicle by planning a path across recently sensed terrain. Fig. 4 illustrates the geometric constraints on the planning problem.

The invariant in the planner is that the *next path* is planned while the *current path*. is traversed. Each next path corresponds to a path across a map generated from an image taken at the first point on the current path.

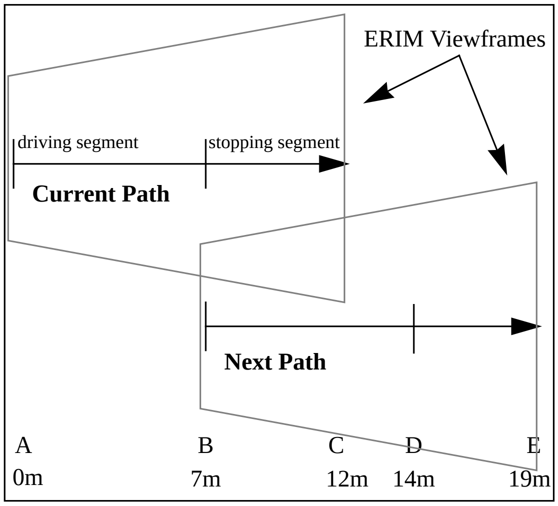

Fig. 4. Geometry for planning

Each path has a *driving segment* and a *stopping segment*. Consider the current path: at the beginning of the driving segment, the vehicle takes an image; as the vehicle drives along this segment, the system plans the next path; if an impasse is reached, and planning fails, the vehicle will drive along the stopping segment, and come to a stop, never having left known terrain. In Fig. 4, an image is taken at A, and path BE is planned while driving AB. If planning fails, BC is driven. Under normal operation, no stopping segment is ever driven, and the vehicle is in continuous motion.

This algorithm provides several guarantees. First, the vehicle can operate in an uncertain environment since the planned path can change radically when unknown obstacles are introduced into the terrain map. Secondly, the system ensures vehicle safety in case of a planning failure, which is essential for high speed driving in rough terrain.

Spatial Trajectory Generation

In the first stage of planning, a generate and test paradigm is used to choose and evaluate successive spatial paths until a safe path which traverses the known terrain is found. This method avoids the problem of preprocessing the terrain to find all unsafe regions. The generate and test method also permits the planner to compute any path needed without limiting the search to a precomputed set.

The planner begins initially with a *global path* describing a course route across a long stretch of terrain. During every sense-plan-drive cycle, the planner tries to move the vehicle along the global path, by choosing *control points.* Control points are points which define the ends of the local path, or which may lead the vehicle around an obstacle. A set of control points which will be used to generate a local path is called a *control path*. Initially, each planning cycle for the next path begins with a control path having two control points: a point from the current path for continuity, and a point a fixed distance ahead to move the vehicle toward the global path.

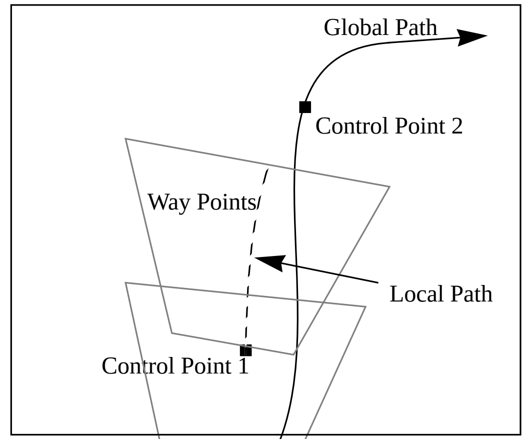

Fig. 5. Global, control, and way points

When a control path is selected for testing, a smooth spatial curve is fit to the control points, and closely-spaced *way points* are generated along the entire length of the curve. Fig. 5 illustrates these concepts. In this figure, Control Point 1 and Control Point 2 comprise the first control path.

The vehicle traverses the way points in simulation. When a way point is tested and found to be unsafe, up to two new control paths are generated. Each new control path will contain an additional control point found by the planner that is intended to modify the path so as to direct the vehicle around either side of the obstacle encountered. These new control paths are added to a list of control paths. The planner then selects a new control path from this list based on the estimated cost (as a function of path length and complexity) of traversing the path. This process repeats until a safe path is found. In theory, this algorithm should be capable of generating traversable trajectories through arbitrarily narrow admissable regions.

In the sections below, the test used to determine safety as well as the algorithm for identifying control points are described.

Kinematic admissibility. Kinematic admissability implies that the geometry of the path during vehicular motion will permit traversal. Traversal can be impeded by an obstacle, such as a tree or a rock, or by vehicle geometry, such as a minimum turning radius. Kinematic admissability may be defined as follows:

A way point is kinematically admissable if there is no geometric cause to impede the proposed motion of the body and if the vehicle and its environment do not occupy common space.

This definition is embodied in a set of four constraints, each of which is evaluated at a given way point to determine kinematic admissability. A waypoint is admissable only if it satisfies all four constraints:

- Minimum turning radius; the curvature of the path at the way point cannot exceed bounds given by the vehicle's minimum turning radius.

A special edge detector is used to find the ambiguity edge near the top of the range image and all information beyond it is ignored. A median filter is used to remove the outliers associated with regions of high terrain texture. No special treatment of specular regions (e.g. water puddles) was required since they are treated automatically as range shadows by the rest of the system.

Vehicle Motion

Clearly, the transform of coordinates from the sensor frame to the world frame requires knowledge of the vehicle pose when each range pixel was measured. The sensor scanning mechanism requires about 1/2 second to complete a single scan. Hence, at higher vehicle speeds, the use of a single vehicle pose for the entire image will give rise to distortions in the map.

The motion distortion removal function uses a series of vehicle poses which were stored by a synchronous pose logger during image digitization. Currently, 8 poses are associated with each image, and, for each pixel, the function uses the pose that was measured closest in time to the instant when the pixel was measured. In this way, an obstacle that would have otherwise been elongated by 2 meters at current speeds is represented accurately in the map.

Terrain Map Generation

The perception system generates a uniformly sampled grid data structure called a cartesian elevation map (Olin, 1991) which stores in each grid cell the elevation of the corresponding point in the environment. The generation of this data structure requires more than the trivial transformation of coordinates from the sensor frame to the world frame. Since one regularly sampled structure (the image) is converted into another (the map), some nontrivial issues rooted in the sampling theorem must be addressed.

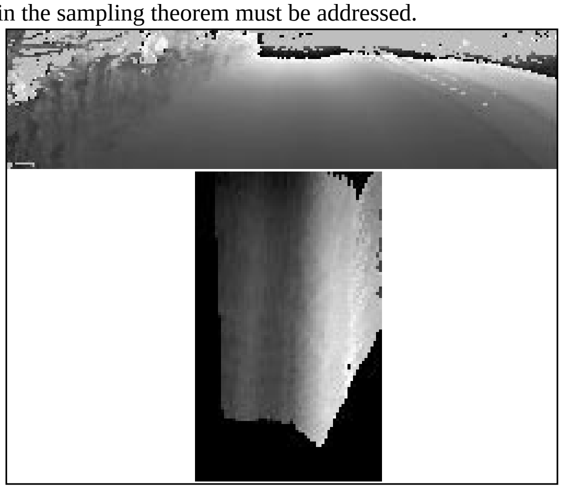

Fig. 3. Typical range image and terrain map

The terrain map generation problem is the inverse of the hidden surface removal problem of computer graphics. The sensor intrinsically removes hidden surfaces since the laser beam is reflected from the closest surface along its trajectory but cannot penetrate to be reflected from any surfaces beyond it. Hence, we must accept that some regions in the elevation map will contain no information. Figure 3 presents a single range image and the portion of the terrain map that results from it, as seen from above.

Clearly, this problem is caused by the sensor geometry. With the sensor attached to the vehicle, the ideal overhead viewpoint is not possible, yet the planner needs to see behind obstacles in order to plan optimally. The sensor configuration also causes regions to be oversampled close to the vehicle and undersampled far away. To some degree, the problem is overcome by fusing the results of several range images from different vantage points into a single map.

The elevation accumulation function computes the average and standard deviation of elevations encountered for oversampled grid points. Range shadows and undersampled regions are filled using a special interpolation algorithm that marks usually large unknown regions as range shadows and records an upper bound on elevation for the grids cells contained in them.

PLANNING SUBSYSTEM

Path planning is defined as finding a path through space from a start point to a goal point while avoiding obstacles. Much of the previous work in this area (Lozano-Perez, 1979), addresses path generation for a polygonal body through a space containing polygonal obstacles. This approach assumes that all obstacles in the traversable space are known. This approach, and other approaches (Thompson, 1987; Feng, 1989) do not satisfy the requirements for ACCN, for the following reasons:

- Polygonal obstacle assumption is not appropriate for rough terrain, since the motion of an obstacle can include any pose (such a steep terrain) that is unsafe. This set of poses can be computationally prohibitive to precompute in it’s entirety and may not assume a polygonal shape.
- A limited sensor horizon precludes knowledge of entire search space before planning
- The intrinsic limitations of the vehicle’s ability to follow the given path are not considered (e.g. minimum turning radius.)

Work by Olin (1991) and Daily (1988), relaxes the flat world, polygonal obstacle assumption and introduces the notion of simulated traversal of paths, taking into consideration kinematic constraints at discrete points along these paths. Their system, however, limits the space of potential paths by only considering a static set of paths across the sensed area. This methodology does not allow small deviations around obstacles, and prevents more than a single change of direction within a planning cycle. This planner might find a region uncrossable simply because its static space of predetermined paths is insufficient to include a safe path. Furthermore, this system does not consider dynamics, as it operated in a speed region where dynamic effects can be neglected.

The trajectory planning software on the Navlab II considers both kinematic and dynamic constraints on the vehicle, plans paths through tight spaces, copes with a changing sensor horizon, and accounts for the intrinsic ability of the vehicle to track a given path.

The system first performs spatial searching based on kinematic constraints and then applies dynamic constraints to set the temporal portion of the path, that is, the speed at

As a result, sophisticated planning is needed. Hughes' ALV has reached speeds of 3.5 km/hr (Daily, 1988; Olin, 1991) in this manner. However, the speed of the ALV system is small enough that many concerns for vehicle safety are diminished, and dynamics need not be modelled for the planning process.

This paper presents an overview of the software and hardware system which addresses the eight problems introduced above, thus permitting ACCN in natural terrain at speeds higher than that of previous systems.

SYSTEM OVERVIEW

The testbed for the system is the NavLab II, a computer-controlled HMMWV (High Mobility Multi-purpose Wheeled Vehicle). The equipment on the NavLab II can be divided into three categories: computation, sensing, and actuation. For computation, there are 3 Sparcstation II's connected by ethernet. For sensing, an ERIM, a 2-d raster scanning laser range finder, is used. The vehicle is totally self-contained, as there no provisions for off-vehicle communications or power supply while in the field. Power is supplied by two onboard generators.

The NavLab II has a central integrated motion controller; its main function is to coordinate control of the vehicles motion control axes to effectively control the vehicle as a whole. The NavLab II has three independent control axes: the throttle, the brake and the steering wheel. The controller determines the vehicle's state using a combination of encoders (mounted on the transmission output and the engine) and an inertial navigation system. Real time software, running on two processors, coordinates the control of the independent axes, keeps track of the vehicle's state, and handles motion commands (steering and speed) from the navigation computers.

In order to traverse unknown terrain, the navigation software must first sense the terrain in front of the vehicle, plan a trajectory across this terrain map, and then drive along the path. This process cycles as the vehicle drives.

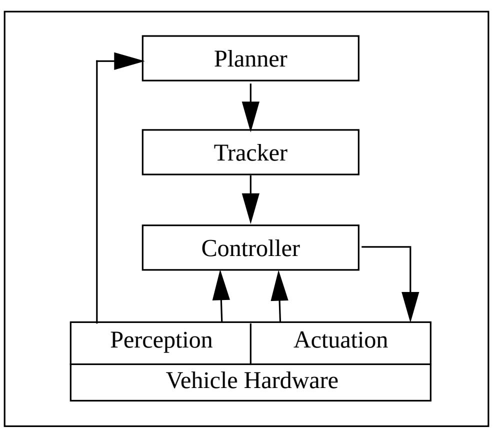

Fig. 1. Software/hardware system interaction

The sensing consists of taking a range image from the ERIM, and transforming the data into a 2.5D discretized terrain map. This operation requires approximately 0.5 seconds.

Path planning is comprised of a spatial planning phase, where a path is generated across the terrain map through heuristic modification to avoid obstacles, and a temporal planning phase, where each point along the path is assigned a speed by considering vehicle dynamics. The separation of dynamics and kinematics makes the planning problem tractable. Computation time for planning is about 0.5 seconds.

The planned path is passed to the path tracker, which tracks the path by issuing commands to the vehicle controller. These components are illustrated in the software system design (Fig. 1). Previous work by Amidi (1990) thoroughly addresses issues involving the tracker and controller.

PERCEPTION SUBSYSTEM

The goal of the perception system for the autonomous navigator is to generate a description of the terrain geometry that is convenient for the path planning subsystem to use, and will accommodate both higher vehicle speeds and rougher terrain than has been achieved in previous systems.

Fig. 2 presents a conceptual view of the software architecture of the perception system:

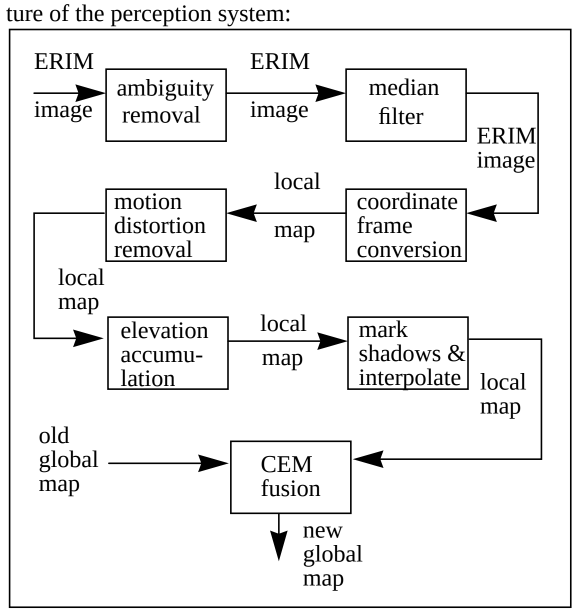

Fig. 2. Perception software architecture

The ambiguity removal and median filter modules remove noise and inaccuracies from the input image. Coordinate frame conversion converts the range information into elevation information. While this is being performed, the motion distortion removal module accounts for vehicle motion to correct the data. The elevation accumulation and shadow marking modules remove holes in the map and compute variation statistics when information overlaps. The Cartesian elevation map (CEM) fusion module is responsible for adding new elevation data to the global map, maintaining a scrolling window into the map which follows the vehicle, and estimating the vehicle's z coordinate by analyzing overlapping map data.

Sensor Limitations

The data generated by the ERIM sensor is subject to limitations of ambiguity of the range measurement beyond the laser modulation wavelength, complete lack of information when the beam is reflected off a specular region of the terrain, and degraded accuracy as range is increased or in regions of high texture.

A SYSTEM FOR AUTONOMOUS CROSS-COUNTRY NAVIGATION

Barry L. Brumitt, R. Craig Coulter, Alonzo Kelly, Anthony Stentz
Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa. 15213

Abstract *Autonomous Cross-Country Navigation requires a system which can support a rapid traverse across challenging terrain while maintaining vehicle safety. This work describes a system for autonomous cross country navigation as implemented on the NavLab II, a computer-controlled off-road vehicle at Carnegie Mellon. The navigation software is discussed. The perception subsystem constructs digital maps of the terrain from range sensor data in real time. The planning subsystem uses a generate-and-test scheme to find safe trajectories through the map, and validates them through simulated driving. The planning process considers both kinematic and dynamic constraints on vehicle motion. The system was successful in achieving 300m autonomous runs on moderate terrain at speeds up to 4.25 m/s.*

Keywords *Autonomous Mobile Robots, Obstacle Avoidance, Navigation, Image Processing.*

INTRODUCTION

Autonomous Cross-Country Navigation (ACCN) can be applied to tasks such as military reconnaissance and logistics, medical search and rescue, hazardous waste site study and characterization, as well as industrial applications, including automated excavation, construction and mining. The need to safely travel at a reasonable velocity on terrain with unknown obstacles is common to all these endeavors.

This paper addresses robot perception, planning and control required to support the fastest possible autonomous navigation on rough terrain. This objective poses new issues not encountered in slower navigation systems on flat terrain.

- Uncertain environment: sensing must be done simultaneously with driving; a precomputed path is impossible.
- Vehicle safety: if computing fails, or a safe trajectory cannot be found, the vehicle must be brought to a stop to avoid collision.
- Computation time: as vehicle speed increases, perception and planning must be performed more rapidly.
- Rugged terrain: the geometry of the terrain is sufficiently complex that the assumption of flat terrain with sparse obstacles does not suffice.
- Dynamics: the vehicle’s speed is large enough that dynamics as well as kinematics must be modelled.
- Imaging geometry: the complexity of map generation increases in the context of rougher terrain.
- Sensor limitations: at higher speeds, technological limitations of the laser range finder are significant.
- Motion during digitization: given higher operating speeds, image distortion is an issue.

Early work in outdoor navigation includes the Stanford Cart (Moravec, 1983). This system drove in a stop-and-go manner at a slow speeds, modelling terrain as flat with sparse obstacles.

Higher speeds have been achieved in system operating in a constrained environment. Road following systems assume benign and predictable terrain. Obstacles are assumed to be sparse, so that bringing the vehicle to a stop rather than driving around obstacles is a reasonable strategy. In this way, speeds of up to 100 km/hr. have been achieved in road following (Dickmanns, 1990; Pomerleau, 1991) using cameras, and speeds up to 30 km/hr. have been achieved on smooth terrain with sparse obstacles using a single scanline lidar (Shin, 1991). While these systems consider problems of dynamics and vehicle safety, they do not provide the capability to drive around obstacles.

Obstacle avoidance has been achieved on approximately flat natural terrain with discrete obstacles (Feng, 1989; Bhatt, 1987; Chang, 1986.) On a smooth gently sloping streambed with sparse obstacles, the JPL rover has achieved a performance of 100 meters in 8 hours (Wilcox, 1987)

When navigating over rough natural terrain, no assumptions about the shape of the terrain ahead can be made. Obstacles may not only be discrete objects in the environment, but can correspond to unsafe configurations as well.