NISTIR 7166

# *Achieving Intelligent Performance in Automomous Driving*

*James Albus*
*John Evans*
*Craig Schlenoff*
*Tony Barbera*
*Elena Messina*
*Stephen Balakirsky*

U. S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Intelligent Systems Division
Gaithersburg, MD 20899-8230

**National Institute of Standards and Technology**
Technology Administration
U.S. Department of Commerce

NISTIR 7166

# Achieving Intelligent Performance in Automomous Driving

*James Albus*
*John Evans*
*Craig Schlenoff*
*Tony Barbera*
*Elena Messina*
*Stephen Balakirsky*

U. S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Intelligent Systems Division
Gaithersburg, MD 20899-8230

October 2003

U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary

TECHNOLOGY ADMINISTRATION
Phillip J. Bond, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arden L. Bement, Jr., Director

# Achieving Intelligent Performance in Autonomous Driving

**A report prepared for**

**Dr. Doug Gage, Program Manager**

**DARPA Mobile Autonomous Robot Software Program**

**Prepared by**

**Intelligent Systems Divison**

**National Institute of Standards and Technology**

**Jim Albus ,PI**

**John Evans[1]**
**Craig Schlenoff**
**Tony Barbera**
**Elena Messina**
**Stephen Balakirsky**

**October, 2003**

[1] John Evans is retired from NIST and is working as an independent consultant for the effort.

# Executive Summary

The Intelligent Systems Division of the National Institute of Standards and Technology has been supporting the DARPA Mobile Autonomous Robot Software (MARS) program over the past two calendar years.

Dr. Doug Gage, the DARPA MARS Program Manager, has expressed interest in an evaluation of what it will take to achieve human level driving skills in terms of time and funding. NIST has approached this problem from several perspectives: considering the current state-of-the-art and extrapolating from there, decomposing the tasks identified by the Department of Transportation for on-road driving and comparing that with accomplishments to date, analyzing computing power requirements by comparison with the human brain, and conducting a Delphi Forecast using the MARS researchers as the experts in the field of autonomous driving.

**Demo III: Current State-of-the-Art**

Within DEMO-III, positive and negative obstacles can be detected, but little object classification is performed. Using the LADAR, terrain is only classified as either vegetation or ground. By adding color images from cameras, terrain can be further classified as green vegetation, dry vegetation, soil/rock, ruts, tall grass, and outliers, but only at very course resolution. The Demo III XUV is badly nearsighted and sensor limited in its performance.

The primary form of knowledge representation in the world model is multiple occupancy grid maps with different size cells as a function of the planning horizon at different levels of control. Underlying data structures are used to associate terrain features with cells in the map. Because of limitations in the object classification, only a small set of data structures are available based on sensor data, while a larger set of data structures is available based upon a priori information.

Planners in the DEMO-III vehicle use value-driven graph search techniques based upon cost-based computations at all levels within the 4D/RCS hierarchy. Multiple planners work concurrently at differing time horizons. Though higher-level planners have been developed to support tactical behaviors and have been tested in simulation, they have not been implemented in any substantial way on the DEMO-III vehicle. Planners have primarily performed waypoint following, obstacle avoidance, and ensuring stability of the vehicle based on the sensed support surface characteristics.

- Based on extrapolation from the Demo III experience, it will take a new generation of sensors and another fifteen calendar years of work at the current level of effort to achieve intelligent on-road driving capability.

**DoT Driver Education Task Analysis Decomposition**

Using the Department of Transportation Driver Education Task Analysis, which identifies 1339 different driving tasks that must be covered in a Drivers' Ed course that are relevant to autonomous driving, an analysis has been made of the number of finite state machine commands that would be required to execute those tasks, the state inputs from the perception system that would be needed to drive those state machines, and the situations and entities that would have to be perceived and understood to correctly identify the necessary states. Table E-1 summarizes our estimation of the number of state tables, situations, world model states, world model entities, and world model entity attributes we believe are necessary to enable autonomous on-road driving, as described above.

| Knowledge | Total Number |
|---|---|
| State Tables (behaviors) | 129 |
| Situations | 1000 |
| World Model States | 10000 |
| World Model Entities | 1000 |
| World Model Attributes | 7000 |

**Table E-1: Knowledge Summary**

The state tables can be completed with a modest effort of two man-years. The major problem is obviously then in perception and world modeling. Analysis of driving tasks has proceeded to the point that the requirements for a new generation of sensors can be identified.

- Perception is the largest problem in autonomous driving, both for on-road and off-road driving. A new generation of sensors is needed to provide the necessary visual acuity. First prototypes can be produced in two to three calendar years at a cost of $5-8 Million; refined, field hardened and tested production versions will ultimately take something like $20-30 Million in engineering costs. The software for perception is at least twice this amount, so total costs for perception will be in the neighborhood of $100 Million or more.

It will take substantial effort to develop the perception and knowledge engineering capabilities to set the 10,000 states that drive the state tables to generate correct driving behaviors. Comparing the accomplishments under Demo III to the requirements from this analysis, an estimate of necessary resources can be made.

- Based on the Task Decomposition of DoT driving tasks, it is estimated that approximately $300-400 million in funding will be needed to achieve intelligent on-road driving skills. The ARL and TACOM autonomous mobility programs

together total approximately $50 Million per calendar year (for multiple projects, not all of which are relevant). Assuming $15-20 Million is relevant funding, this would imply that it will take approximately two decades of additional work at current support levels to reach intelligent on-road driving performance.
- Increased funding would shorten this time horizon. If adequate funding were available, it is estimated that intelligent on-road driving could be achieved within a decade, possibly as soon as 2010.

**Analysis of Computing Power**

Using several approaches to estimation, it is concluded that computing requirements for driving at intelligent skills will be in the range of $10^{11}$ to $10^{14}$ instructions per second and that a credible attack on the problem will require a minimum level of $10^{11}$ to $10^{12}$ instructions per second. Cluster computers could be built with today's processors to achieve these levels.

- Adequate computing power using cluster computers is now or will soon be available. Computing power should not be a gating element, but engineering attention needs to be paid to providing adequate processors with adequate inter-processor communication and software development tools to researchers.

**Delphi Forecast**

A Delphi forecast, named for the Oracle at Delphi who was said to be able to forecast the future, is a poll of experts as to when a certain future event might take place. The concept is that a mean prediction of experts is as good an indicator of future events as is possible to achieve. A poll was taken of the MARS researchers at the MARS Principal Investigators' meeting in San Diego in April, 2003.

- Based on the consensus of MARS researchers, it will take 15-20 calendar years and of the order of $500M to achieve intelligent on-road driving skills.

Several MARS researchers emphasized that setting human level driving skills as the goal was not the correct approach, that militarily useful capabilities would be achieved short of that goal. Individual responses were sought from many of the participants to clarify their positions; those are presented below in Section 6. All researchers felt that continued research was needed.

- Targeting specific military driving modes to be solved in the foreseeable future will still require continued research in sensors, perception, knowledge management and planning.

## Conclusions

**While the spread in these estimates is significant, the overall conclusions are that:**

- **Militarily useful autonomous driving capabilities can be developed in approximately ten to twenty calendar years on continued research. The time scale will depend upon the level of funding available.**
- **The cost will be in the range of three to five hundred million dollars, which is consistent with current funding levels of Army autonomous mobility programs extended over twenty calendar years.**
- **If adequate funding were available, it is estimated that intelligent on-road driving could be achieved within a decade, possibly as soon as 2010.**
- **The biggest single problem is perception. The attack on the problem should start with development of a new generation of sensors designed specifically for autonomous driving.**
- **Continued research in sensors, perception, knowledge management and planning, at a level at least equal to current funding is essential, even if the scope is reduced to targeting specific military driving modes to be solved in the near term.**

# 1.0 Introduction

The Intelligent Systems Division at the National Institute of Standards and Technology (NIST) has been supporting the Defense Advanced Projects Agency (DARPA) Mobile Autonomous Robot Software (MARS) program over the past two calendar years.

Dr. Doug Gage, the MARS Program Manager, proposed that a significant benchmark for autonomous driving would be a system equivalent to a human chauffeur. This "robot chauffeur" would be able to navigate roads and traffic on highways and in cities, finding and driving to a requested destination. This is, more or less, the capability that Army recruits bring with them to boot camp. The Army then provides additional training for those selected to be Scouts, adding specific skills in off-road driving and understanding of tactical behaviors. The Army could provide the same incremental training for an autonomous system to produce a capable robot scout.

The questions important to planning at DARPA and the Army are, then, when will we achieve human equivalent driving capability and how much effort will it take?

NIST has addressed this question in four different ways:

- Extrapolating from the State of the Art as represented by the Army Demo III Experimental Unmanned Ground Vehicle project
- Estimating the amount of effort to build an autonomous driving system with the capabilities defined by the Department of Transportation Manual Driver Education Task Analysis
- Estimating necessary computer processing capability by comparison with the human brain; and
- Using a Delphi Forecast to poll the MARS researchers to obtain a consensus estimate of experts in the field of autonomous driving.

Dr. Gage believes strongly that there is an inverse relationship between the time needed to achieve a goal and the level of funding for work toward that goal. Obviously you can't make a baby with nine women in one month, but in most cases you can accelerate technology development with increased levels of funding. His "time/money" slide is shown in Figure 1. He points out that this is a caricature of "management decision space" and is not meant to represent actual programmatic data (KITT is the intelligent car from the TV series Knight Rider and DATA is the character in Star Trek, Next Generation).

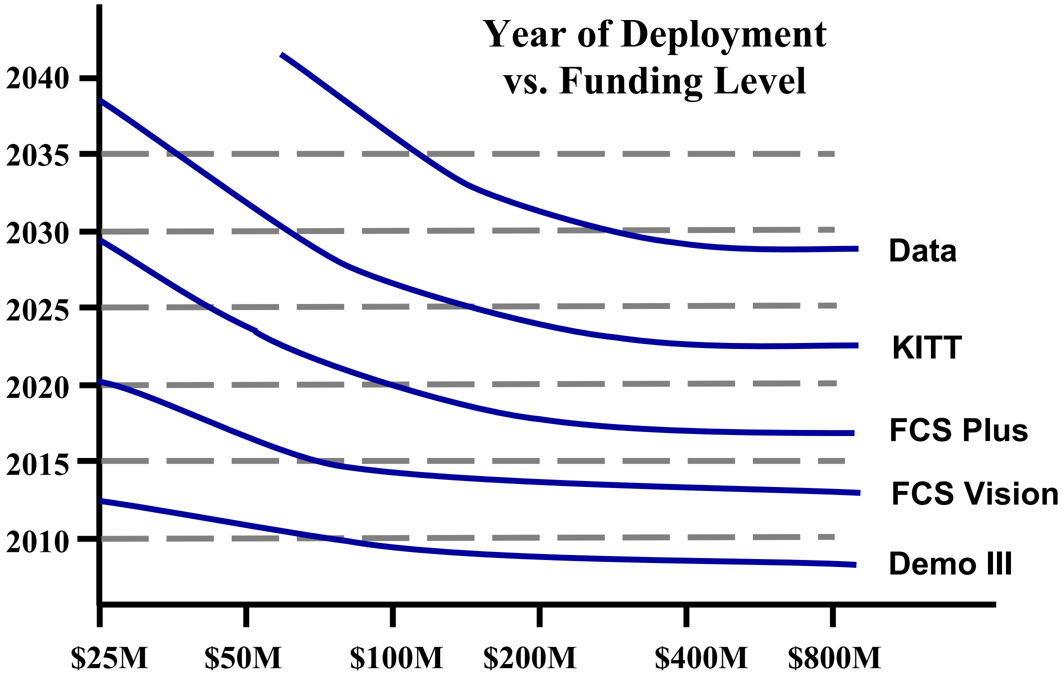


**Figure 1**

While we don't know what these curves really look like, some inverse relationship between funding and time scale is undoubtedly valid within ranges of modest funding relative to the goal complexity.

This report argues from several different standpoints as to what might be the levels of effort required to achieve the "robot chauffer." As has just been pointed out, there is a trade-off between time to achieve a goal and levels of funding; we estimate time frames assuming current levels of funding and then point out the chances to reduce those time frames.

## 1.1. Needs for Future Combat Systems Vehicles

The first question to address is the target goal point: what vehicles are we trying to drive and where are we driving them?

This report assumes that appropriate vehicle platforms are being developed under other programs. For example, the XUV platform used in the Demo III program was specifically developed for autonomous scout missions. Future Combat Systems is developing three new platforms, a small sensor platform, the Unmanned Armed Reconnaissance Vehicle (UARV), and a robot "Mule" transport vehicle. In addition, the UGCV program has an articulated vehicle under development and the Tactical Mobile Robot (TMR) program developed the "Packbot" and "Throwbot" platforms that will be wrapped into FCS but which are not suitable for highway driving. Finally, many different vehicles have been converted for teleoperation by the Department of Defense and could be further modified for autonomous driving by the addition of an Autonomous Navigation System package of sensors, computers, and software.

The primary targets for advanced autonomous driving capability are the FCS and Demo III platforms. These are under development with substantial funding commitments and will be available in production versions before intelligent on-road driving is achieved. Production versions of wheeled vehicles are expected to be qualified for highway driving.

Appropriate vehicle platforms and the ANS baseline are assumed. The problem set that needs to be addressed, then, is the sensors, the computing platforms and the software beyond the required ANS capabilities of supervised teleoperation that are needed for intelligent on-road driving.

Following Dr. Gage's direction, this report focuses on the sensors, computers and software for autonomous on-road driving, the "robot chauffeur," with Future Combat Systems as the primary ultimate customer.

## 1.2. Needs for Intelligent Transportation Systems (DOT)

Researchers in the Department of Transportation Intelligent Transportation Systems program envisages extensive use of automated vehicle guidance (AVG) technology for public transit vehicles, for local shuttles to service public transportation stops, and for automobiles and trucks in urban environments. [9]

DOT points out that it will be impossible to build sufficient additional road infrastructure to accommodate the increase in population and the increase in vehicles per capita that can be expected in the future. The only option is to increase the effective utilization of existing infrastructure through better public transit and through AVG technology.

DOT sees AVG technology as embodying modifications to the roadway infrastructure (marked and controlled lanes with computer supervision, wireless communication with automated vehicles, and controlled entrance and exit gates for AVG lanes) as well as the sensors and controls needed for basic AVG technology. The sensors and controls needed for the DOT scenarios are therefore somewhat simpler than those needed for the general-purpose unrestricted "robot chauffeur."

Substantial progress in bringing adaptive cruise control and lane following to commercial and public service applications has been made around the world. This represents an excellent baseline for further work toward autonomous driving.

## 1.3. Report Structure

Chapter 2 of this report summarizes the state of the art in terms of the Demo III experience.

Chapter 3 provides a task analysis based on the DOT manual

Chapter 4 considers the needs for improved sensors.

Chapter 5 analyzes computing power requirements

Chapter 6 presents the results of the Delphi Forecast carried out at the April, 2003, MARS Principal Investigators' meeting in San Diego.

Finally, Chapter 7 itemizes the main conclusions drawn in earlier chapters.

## 2.0 Current State of the Art

In order to determine how much is it going to take to reach intelligent performance in on-road and off-road driving, we must first understand what is achievable now. We can use our current capabilities as a benchmark, and extrapolate out to determine what it would take to achieve intelligent level of on-road driving performance.

The DEMO III Experimental Unmanned Vehicle (XUV) effort seeks to develop and demonstrate new and evolving autonomous vehicle technology, emphasizing perception, navigation, intelligent system architecture, and planning. [16] Many believe that this effort represents the state of the art in autonomous driving. As such, we will use this effort to serve as a benchmark to represent what we can do now, and then project to the capabilities needed to enable intelligent levels of performance. [16]

The autonomous navigation system (ANS) within the DEMO-III effort was recently declared to have reached Technology Readiness Level 6 (TRL-6), indicating that the ANS has been demonstrated and tested in a relevant environment. [5] Though focusing primarily on off-road driving, the authors believe that the technology used in DEMO-III will lend itself well as a starting point for on-road driving, as discussed below in Section 3.

In this section, we will look at the state of the art of the overarching architecture, and the three main subsystems within the autonomous navigation systems: perception/sensory processing, world modeling, and behavior generation.

### 2.1. Architecture

Within DEMO-III, the 4D Real-Time Control System (4D/RCS, the 4D referring to planning in three spatial dimensions plus time, as used in the German autonomous driving program) was used as the underlying architecture within the autonomous mobility system. This architecture provides a reference model for the identification and organization of software components for autonomous driving of military unmanned vehicles. 4D/RCS defines ways of interacting to ensure that missions, especially those involving unknown or hostile environments, can be analyzed, decomposed, distributed, planned, and executed intelligently, effectively, efficiently and in coordination. To achieve this, the 4D/RCS reference model provides well-defined and highly coordinated functional modules for sensory processing, world modeling, knowledge management, cost/benefit analysis, and behavior generation, and defines the interfaces and messaging between those functional modules. The 4D/RCS architecture is based on scientific principles and is consistent with military hierarchical command doctrine. [1]

Figure 2-1 shows a high-level block diagram of a 4D/RCS reference model architecture for a notional Future Combat System (FCS) battalion. 4D/RCS prescribes a hierarchical control principle that decomposes high-level commands into actions that employ physical actuators and sensors. Characteristics such as timing and node functionality may differ in various implementations.

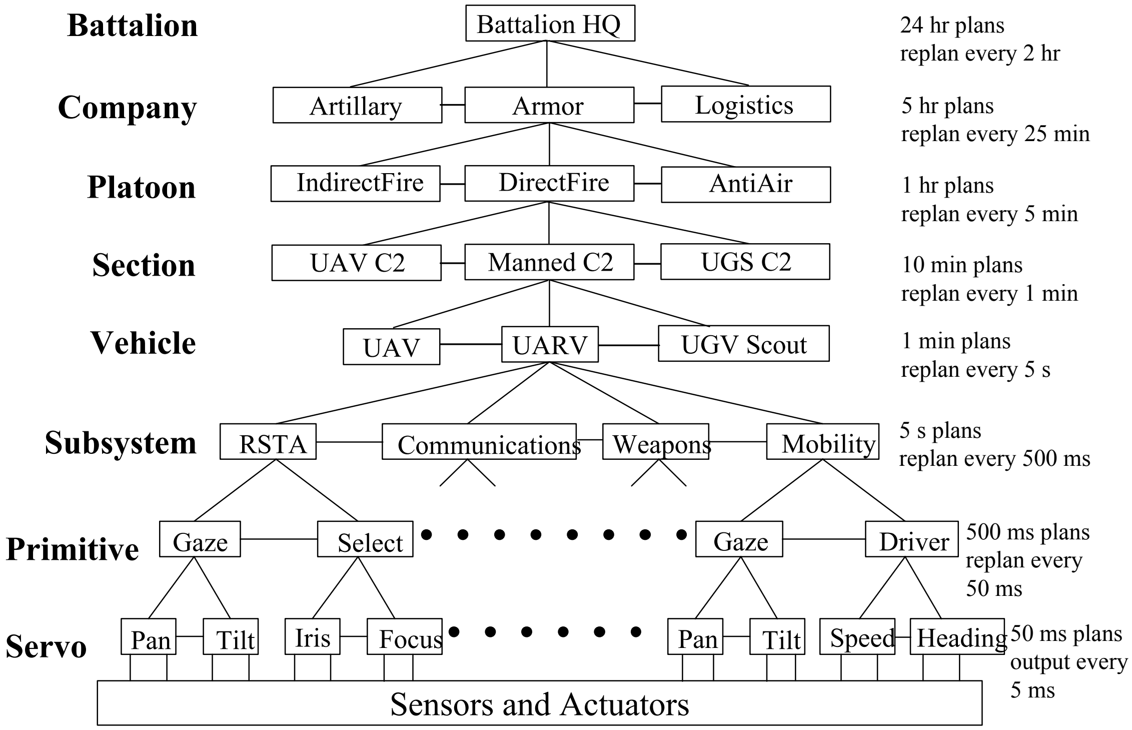


**Figure 2-1: A high level block diagram of a typical 4D/RCS reference model architecture.** Commands flow down the hierarchy, and status feedback and sensory information flows up. Large amounts of communication may occur between nodes at the same level, particularly within the same subtree of the command tree. UAV = Unmanned Air Vehicle, UARV = Unmanned Armed Reconnissance Vehicle, UGS = Unattended Ground Sensors

The functions of the various levels in this hierarchical decomposition are as follows:

- At the Servo level, commands to actuator groups are decomposed into control signals to individual actuators. In the example shown in Figure 2-1, outputs to actuators are generated every 5 milliseconds (ms). Plans that look ahead 50 ms are regenerated for each actuator every 5 ms. Plans of individual actuators are synchronized so that coordinated motion can be achieved for multiple actuators within an actuator group.

- At the Primitive level, multiple actuator groups are coordinated and dynamical interactions between actuator groups are taken into account. Plans look ahead 500 ms and are recomputed every 50 ms.

- At the Subsystem level, all the components within an entire subsystem are coordinated, and planning takes into consideration issues such as obstacle avoidance and gaze control. Plans look ahead 5 seconds (s) and replanning occurs every 500 ms.

- At the Vehicle level, all the subsystems within an entire vehicle are coordinated to generate tactical behaviors. Plans look ahead 1 min and replanning occurs every 5 s.

- At the Section level, multiple vehicles are coordinated to generate joint tactical behaviors. Plans look ahead about 10 minutes (min) and replanning occurs about every minute.

- At the Platoon level, multiple sections containing a total of 10 or more vehicles of different types are coordinated to generate platoon tactics. Plans look ahead about an hour (hr) and replanning occurs about every 5 min.

- At the Company level, multiple platoons containing a total of 40 or more vehicles of different types are coordinated to generate company tactics. Plans look ahead about 5 hr and replanning occurs about every 25 min.

- At the Battalion level, multiple companies containing a total of 160 or more vehicles of different types are coordinated to generate battalion tactics. Plans look ahead about 24 hr and replanning occurs at least every 2 hours.

At all levels, task commands are decomposed into jobs for lower level units and coordinated schedules for subordinates are generated. At all levels, communication between peers enables coordinated actions. At all levels, feedback from lower levels is used to cycle subtasks and to compensate for deviations from the planned situations.

Figure 2-1 shows levels that are specific to military vehicles and, above the vehicle level, to the coordinated control of multiple military vehicles. Each vehicle will contain surrogate levels for the higher levels of planning above the vehicle level, such that if communications are lost with external higher-level planners, each vehicle can autonomously generate appropriate plans for itself on its own. In Section 3 a hierarchical decomposition will be shown for on-road driving, where each vehicle is assumed independent and must create its own plans for complete trips, and where the specific level designations are renamed appropriately.

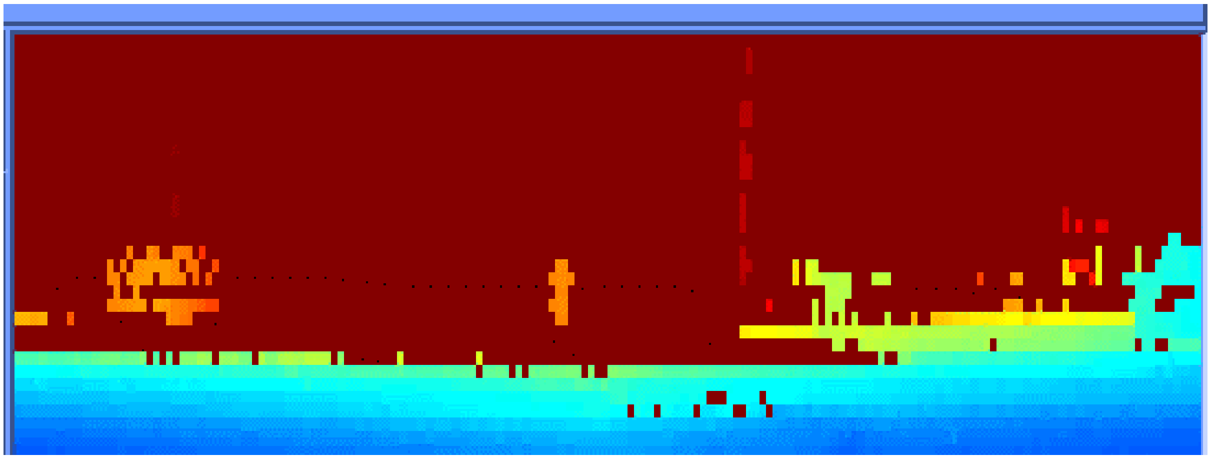
## 2.2. Sensors/Sensory Processing

Sensory processing algorithms use sensor data to compute vehicle position, range, obstacle lists, obstacle positions, and terrain information. The suite of sensors used in the mobility system include a General Dynamics/Schwartz Electro-Optics Scanning Laser

Rangefinder (LADAR)[3], a pair of color cameras for stereo vision, a stereo pair of Forward-Looking Infra-Red (FLIR) cameras, a stereo pair of monochrome cameras, a pan-tilt platform, a global positioning system (GPS) sensor, a force bumper that alerts the system to obstacles in the vehicle's immediate path, and an inertial navigation system (INS) sensor. All sensors are mounted on the vehicle, which is equipped with electric actuators on the steering, brake, transmission, transfer case, and parking brake. Feedback from the sensors provides the controller with engine rotations per minute, speed, temperature, fuel level, etc. A Kalman filter computes vehicle position and orientation using data from the internal dead reckoning system and the carrier phase differential GPS unit.

### 2.2.1. LADAR sensor

The LADAR sensor provides approximately 60,000 point range measurements per second in an image array of 32 by 180 pixels covering a field of view (FOV) of about 20° in elevation by 90° in azimuth. The sensor is mounted on a pan/tilt platform to increase its rather narrow 20° vertical field of view (FOV). The range of the tilt motion is ± 30° resulting in an accessible elevation field of view of about 80°. Using *a priori* knowledge about the location and orientation of the LADAR mounting on the vehicle, calibration factors, and vehicle position data, the range information is transformed into position and orientation values in a world coordinate frame. A typical frame is shown in Figure 2-2 below. The resolution is quite course (.5 degree/pixel) and the sensor can only see the ground out to about 20 m, so the vehicle is quite nearsighted. This scene is of a soldier at a distance of about 20m. Obviously the image is very crude and does not allow object identification at any distance.



[3] Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

**Figure 2-2: Demo III Scanning LADAR Image**

Obstacles are defined as objects that project more than some distance above or below the ground plane (defined as the plane on which the wheels of the vehicle lie). Positive obstacles, which extend above the ground plane, are detected directly in the range images, while negative obstacles are detected by inference as holes in the world model map.

After a group of pixels has been labeled as an obstacle, additional processing is performed to classify the obstacle type. The quality of the GD/SEO range data precludes more than a coarse classification, which currently identifies only vegetation and ground. [11]

### 2.2.2. Stereo vision sensors

Stereo vision provides another way of computing range information. The system is equipped with a color camera pair with a 60° FOV and a FLIR camera pair with a 40° FOV for night vision. The stereo system includes an iris controller; an image acquisition unit; a stereo range algorithm; positive and negative obstacle detection algorithms; and a terrain classification algorithm.

A multi-resolution approach, working from coarse to fine, is taken to determine correspondence between the left and right images, resulting in a range image. For each range image column, a set of obstacle detectors is applied to extract gaps and discontinuities in the range data that indicate non-traversable regions. Non-traversable regions are classified into either negative or positive obstacles. Negative obstacles are detected by checking for gaps in the range data followed by a range jump. Positive obstacles are detected by checking for upward slanted edges in the range data, i.e., any upward protrusion out of the ground plane steep enough to be non-traversable or to cause a tip-over hazard.

The LADAR data generally proved to be more robust than stereo. Stereo does not work well when there are few definite verticals and does not work well when there is too much fine-grained texture across the entire scene.

Terrain classification is performed on color images taken from one of the stereo images. Classification types currently include green vegetation, dry vegetation, soil/rock, ruts, tall grass, and outliers. The classification algorithm relies on color, and is based on Bayesian assignment. [11]

***Salient Point: Within DEMO-III, positive and negative obstacles can be detected, but little object classification is performed. Using the LADAR, terrain is only classified as either vegetation or ground. By adding color images from cameras, terrain can be***

***further classified as green vegetation, dry vegetation, soil/rock, ruts, tall grass, and outliers, but only at very course resolution.***

## 2.3. World Model

For the purpose of this paper, we describe the world model as

> "the system's internal representation of the external world. It provides a central repository for storing sensory data in a unified representation, and decouples the real-time sensory updates from the rest of the system… World modeling processes fuse information from multiple sensors, including navigation sensors, LADAR, and stereo vision. The world model incorporates a set of maps at multiple resolutions. Each map fuses sensory information and a priori knowledge into its occupancy grid representation. Information at different hierarchical levels has different spatial and temporal resolutions. The map is north-oriented and scrolls as the vehicle moves. Various features are integrated over time, computing confidence and filtering out spurious false detections." [11]

### 2.3.1. Subsystem and Primitive Levels

Data from multiple sensor modalities is fused in an occupancy grid map in a way suitable for path planning and vehicle control. The map consists of a two-dimensional array (301x301 cells) containing information extracted from the processed sensor data. The total extent of the map used in Demo III is 120 m x 120 m, so each cell in the map grid represents an area of 0.4m x 0.4m. The information stored in a cell includes:

- The average ground elevation height, the variance of the height, and an elevation confidence measure.
- A data structure describing the terrain covered by the cell. This includes a terrain label (tall grass, dry vegetation, ruts, etc.) and a cost factor for determining the relative safety of traversing that cell.
- A linked list structure describing the type of object viewed by the sensor (e.g., roads, buildings, fences, etc.). Each object has a name, a position, a confidence measure, and a time stamp. [11]. Note that because of limitations in object classification, this linked list is available in concept but has not been fully implemented on the DEMO-III vehicle. Only a small set of data structures can be classified based upon sensor data.

### 2.3.2. Vehicle and Section Levels

The vehicle and section levels also use a modified form of the obstacle grid map, with each cell representing a 1m x 1m space or a 10m x 10m space. At these higer levels, however, an *a priori* knowledge base is linked to cells in these maps which contains a very rich representation of features in the outside world at a resolution and extent that is dictated by the level of the architecture where it resides. Information in the knowledge database is stored in attribute layers, where each group of related features is represented as an independent layer. In Demo III, layers include an *a priori* layer that contains static knowledge about the environment and an obstacle layer that contains dynamic knowledge. The basic form of the layer is a combination of a regular *n*-dimensional grid of cells that represents the system's discrete state space with regard to the layer's features and a database of specific feature instantiations. Each cell of the grid structure contains a set of flags that denotes which of that layer's possible features are contained in the cell and pointers to the specific instantiations of each contained feature. Features, along with their attributes, are stored in an underlying relational database. A feature may be a road, with attributes including the number of lanes, speed limit, road marking, etc. If a cell in the obstacle map contains a road object, a bi-directional pointer would exist between the instantiation of the feature in the relational database and the cell in the obstacle map. [11]

***Salient Point: The primary form of knowledge representation in the world model is multiple occupancy grid maps with different size cells as a function of the planning horizon at different levels of control. Underlying data structures are used to associate terrain features with cells in the map. Because of limitations in the object classification, only a small set of data structures are available based on sensor data, while a larger set of data structures is available based upon a priori information.***

## 2.4. Behavior Generation/Planner

The behavior generation subsystem uses value-driven graph search techniques based upon cost-based computations at all levels within the previously described 4D/RCS reference model architecture. The function of the behavior generation at every level of the hierarchy is the same: to create ordered time-tagged sets of actions to be performed by the subordinate levels and to execute these actions.

### 2.4.1. Section and Vehicle Level

The role of the section level planner is to generate plans that last approximately 10 minutes and span approximately 10,000 meters in length with waypoints approximately every 500 meters. The role of the vehicle level planner is to generate plans that last approximately 1 minute and span approximately 1,000 meters in length with waypoints every 50 meters.

At the section and vehicle level, the planner mixes a rule-base with a value-driven cost evaluation to perform behavior generation. This allows vehicles to move across the battlefield in an intelligent fashion. For example, this means that the vehicle does not only move across the battlefield in a safe manner, but also can perform specific military behaviors that are governed by rules from military doctrine, such as formation maintenance or over-watch, while seeking out or avoiding certain terrain features to allow for stealthy movement. [2] The planner at these levels also plans on incrementally created planning graphs as described in [3].

This planner was ported to the XUV for DEMO-III but not used to its full capacity due to the emphasis on the lower-level mobility and planning issues. This planner was used to a greater extent, however, in other unmanned vehicle demonstrations. Most tactical behaviors, such as the ones described in the previous paragraph, remain elusive and were not exhibited in any meaningful capacity during the DEMO-III effort.

### 2.4.2. Subsystem and Primitive Level Planner

The role of the subsystem level planner is to generate plans that last approximately 5 seconds and span approximately 100 meters in length with waypoints approximately every 5 meters. The subsystem level representation only contains obstacles and a priori data. The trajectories used by this level are straight-line approximations. Vehicle dynamics are considered at the primitive level and are ignored at the subsystem level. The planner finds the optimal shortest obstacle-free path available in the graph. [13]

The role of the primitive level planner is to generate plans that last approximately 0.5 seconds and span approximately 10 meters in length with waypoints approximately every 0.5 meters. At the primitive level, the support surface is used to determine the stability as well as the roughness of the ride through several potential plans. The primitive level utilizes a set of pre-computed trajectory path templates that include the vehicle dynamics, including linear and angular speed and acceleration. The throttle, brake, and steering actuators can only change the linear and angular speed at certain limited rates, which means the vehicle can only execute certain limited trajectories. A set of those trajectories that span the possible set of all trajectories are overlaid on the occupancy grid map. Each trajectory is then followed from cell to cell, calculating the cost to traverse each potential path. The cost function includes vehicle pitch and roll, roughness of the terrain, terrain characteristics, and all linear and angular accelerations. In addition, each possible trajectory is checked for protruding objects that may hit the undercarriage.

Another important factor is whether a cell contains recent sensor data. The cost evaluation will assign larger costs to trajectories that place wheels of the vehicle in cells that have never been seen by a sensor because these cells have unknown elevation and may be holes or ditches. All of these parameters are taken under consideration in order to calculate the cost of each trajectory. Replanning at this level is done at 4-10 Hz. [13]

***Salient Point: Planners in the DEMO-III vehicle use value-driven graph search techniques based upon cost-based computations at all levels within the 4D/RCS hierarchy. Multiple planners work concurrently at differing time horizons. Though higher-level planners have been developed to support tactical behaviors and have been tested in simulation, they have not been implemented in any substantial way on the DEMO-III vehicle. Planners have primarily performed waypoint following, obstacle avoidance, and ensuring stability of the vehicle based on the sensed support surface characteristics.***

## 2.5 Extrapolating from the Demo III Experience

The discussion above highlights perception as the “tallest pole in the tent.” Demo III has had some significant success, but it is badly nearsighted and can see only with very course resolution. Cost-based planning has been quite successful to the extent that the sensor generated obstacle maps contain adequate data. Large obstacles, both positive and negative, are routinely avoided and the vehicles can successfully follow roads and waypoints across modest off-road terrain.

As will be analyzed in the coming sections, the most important near term focus should be on new generations of sensors and sensory perception. The current CTA extension of Demo III is indeed committing substantial resources to new generations of LADAR, but more work is needed. There is no one responsible for developing sensors specifically for autonomous driving but there should be.

Considering the time and resources that have been spent on Demo III, it is roughly estimated that another decade and total funding of the order of several hundred million dollars will be needed to achieve capability close to intelligent performance in driving. ***This is roughly a continuation of current levels of funding for approximately another fifteen calendar years.***

As was pointed out in the Introduction, Section 1, there is a trade-off between levels of funding and time to realize needed capabilities. In this case ***we estimate that doubling of effort (i.e. doubling of funding) would cut the time to realize intelligent driving to no more than a decade. That is, intelligent driving could be achieved within one decade and possibly as soon as 2010 if adequate funding were provided.***

Of particular value to the Demo III continuation would be fielding of multiple vehicles with full support teams such that demonstrations and testing could be carried out in parallel with development. The current practice is to stop development during demonstrations and field tests since the same vehicles and same staff is responsible for all of these activities.

# 3.0 Task Decomposition

## 3.1. Approach

As part of the DARPA MARS program, an effort has focused on analyzing what it would take to achieve intelligent performance for on-road driving. The goal of this effort is to provide a task analysis description of the on-road driving task at a level of detail to be able to support work in the design and development of autonomous driving systems. This effort, therefore, requires the collection, ordering, and representation of the knowledge set that encompasses all of the on-road driving activities. This knowledge set has been assembled from a number of different sources. The single largest source document has been the Department of Transportation (DOT) manual entitled Driver Education Task Analysis, Volume 1, Task Descriptions [14], authored by James McKnight and Bert B. Adams. Table 3-1 lists each section of the DOT manual, and includes the number of driving tasks that were listed in each section that are appropriate for autonomous driving. Examples of tasks that are not appropriate include adjusting mirrors, changing the oil, and adjusting head support.

Significant additional sources have been the DOT Manual of Uniform Traffic Control Devices (MUTCD) document [17], numerous state traffic law documents, and considerable discussion by the authors in attempting to mine their own driving task knowledge.

| **DOT Manual Section #** | **Section Description** | **Number of Relevant Task Items** |
|---|---|---|
| 11 | Pre Operation | 5 |
| 12 | Starting | 30 |
| 13 | Accelerating | 54 |
| 14 | Steering | 17 |
| 15 | Speed Control | 13 |
| 16 | Stopping | 20 |
| 17 | Backing Up | 12 |
| 18 | Skid Control | 17 |
| 21 | Surveillance | 32 |
| 23 | Navigation | 10 |
| 24 | Urban Driving | 16 |
| 25 | Highway Driving | 10 |
| 26 | Freeway Driving | 22 |
| 31 | Following | 30 |
| 32 | Passing | 67 |
| 33 | Entering & Leaving Traffic | 18 |
| 34 | Lane Changing | 13 |
| 35 | Parking | 76 |

| | | |
|---|---|---|
| 36 | Reacting To Traffic | 202 |
| 41 | Negotiating Intersections | 132 |
| 42 | On Ramps and Off Ramps | 82 |
| 43 | Negotiating Hills | 26 |
| 44 | Negotiating Curves | 13 |
| 45 | Lane Usage | 11 |
| 46 | Road Surface & Obstructions | 135 |
| 47 | Turnabouts | 35 |
| 48 | Off-Street Areas | 55 |
| 49 | RR Crossings, Bridges, Tunnels | 55 |
| 51 | Weather Conditions | 21 |
| 52 | Night Driving | 32 |
| 61 | Hauling & Towing Loads | 42 |
| 62 | Responding to Car Emergencies | 31 |
| 63 | Parking Disabled Cars | 5 |
| Total | | 1339 |

**Table 3-1: Relevant DOT Manual Task Items**

The above documents provided a large set of the on-road knowledge as it applies to human drivers. These documents, however, have the shortcoming of not detailing the assumed driving knowledge such as the understanding of what attributes of roads and intersections are to be perceived, how vehicles are to be characterized, how objects (both animate and inanimate) are to be sensed in order to allow an autonomous computer control system to recognize and reason about them relative to the driving task context. As a result, a major effort of this work has been to attempt to define the database structures that might be used to represent all of the knowledge required about roads and entities.

The overall approach is to analyze the driving tasks through a discussion of a large number of scenarios of particular on-road driving subtasks and to derive from these descriptions a task decomposition tree representation of all the task activities at various levels of abstraction and detail. From this task tree we can organize the activities into a more rigorous layering by the artifice of identifying an organizational structure of agent control modules that are responsible for executing the different levels of the task decisions. The organization structure which was developed for this effort can be found in Figure 3-1.

One may notice that the terminology used in the agent control models at each level of the control hierarchy is different then that present in Figure 2-1 in Section 2. In both hierarchies, the terminology used is tailored towards the domain of interest. Table 3-2 shows the correlation between the terminology of Figure 2-1 in Section 2 and Figure 3-1 in this section. While the terminology is different, the levels correspond and the time horizons for planning and replanning are very similar.

| Figure 2-1 in Section 2<br>(Future Combat Systems Hierarchy) | Figure 3-1 in Section 3<br>(On-Road Driving Hierarchy) |
|---|---|
| Servo | Steer Servo, Speed Servo |
| Primitive | GoalPath Trajetory |
| Subsystem | Elemental Maneuver Subsystem |
| Vehicle | DriveBehavior Manager |
| Section | RouteSegment Manager |
| Blatoon | Destination Manager |
| Company | Journey Manager |

**Table 3-2: Terminology Correlation Between Control Hierarchies**

This use of separate executing agents organized into an execution hierarchy provides a mechanism to formalize the task decision tree by assigning certain decisions to particular agent control modules. This creates well-defined sets of subtask commands from each supervisor agent control module to its subordinate agent control module, thus forcing us to group and label various sets of related activities of the driving task with a context identifier such as “PassVehInFront”, “TurnLeftAt_StopSign”, “PullOffOnto_LeftShoulder” etc. Each of these identifiers is really a subtask goal command at different levels in the execution hierarchy. The task decision rules appropriate to each of these subtask goal commands that identify the partial task decomposition of the driving task that occurs within the one agent control module’s level of responsibility can be encoded within Finite State Machines (FSMs). These FSMs can be represented in both a state graph (Figure 3-2) as well as a state-table format (Figure 3-3). In each of these FSMs are structured the set of rules that identify both the particular situations that will trigger the FSM to step to the next state and the output action which is the result of this task decision. This applies a well-structured formalism to the task description while keeping it easily understandable to the user since each FSM only encodes the small number of rules associated with one particular subtask activity at one level in the task decomposition decision tree. [4]

It should be noted that there is a distinction between the agent hierarchy in Figure 3-1 and the organizational unit hierarchy in Figure 2-1. An example of an agent hierarchy is (private, lieutenant, captain, major, colonel, general). An example of an organizational hierarchy is (vehicle, section, platoon, company, battalion). Further discussion of this distinction can be found on page 60 of [1].

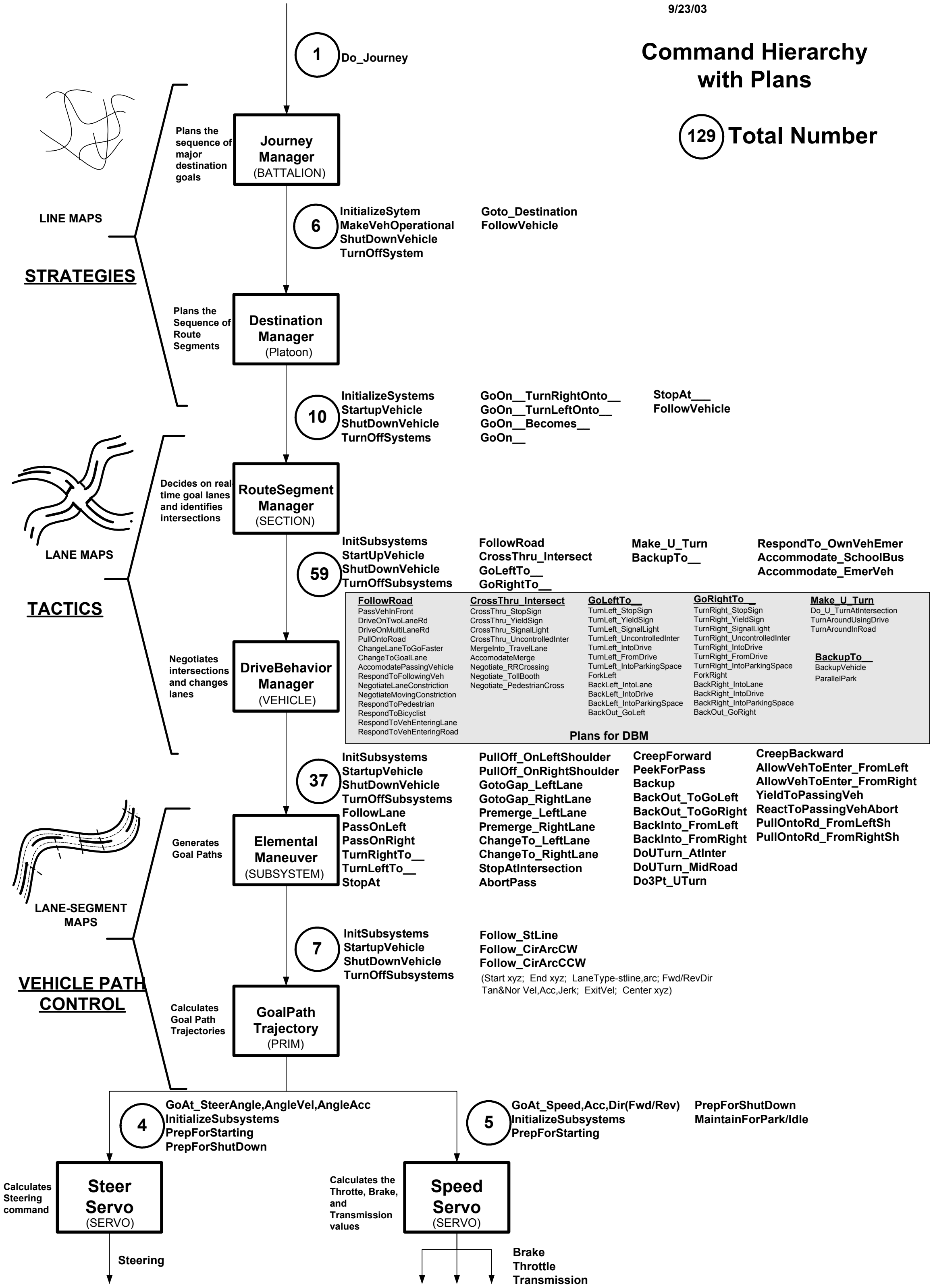
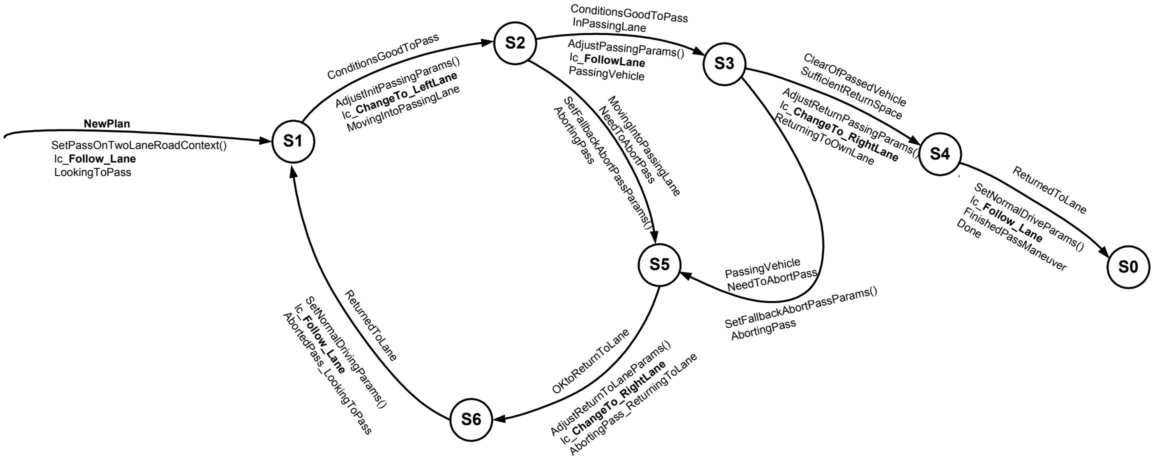


**Figure 3-1: Command Hierarchy With Plans**



PLAN STATE GRAPH

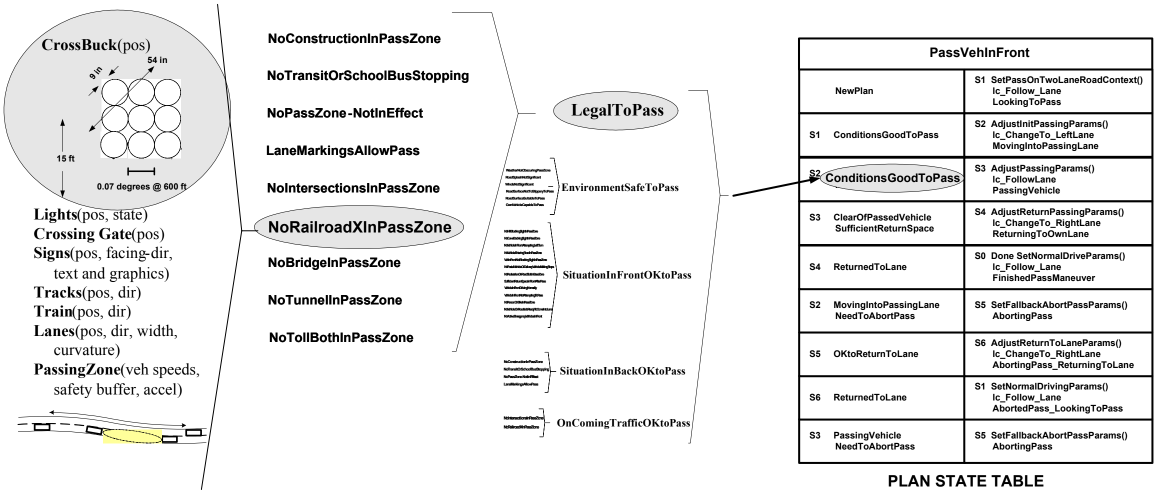
**Figure 3-2: State Graph Representation of "Pass on Two-Lane Road"**

| PassVehInFront | |
|---|---|
| NewPlan | S1 SetPassOnTwoLaneRoadContext()<br>lc_Follow_Lane<br>LookingToPass |
| S1 ConditionsGoodToPass | S2 AdjustInitPassingParams()<br>lc_ChangeTo_LeftLane<br>MovingIntoPassingLane |
| S2 ConditionsGoodToPass<br>InPassingLane | S3 AdjustPassingParams()<br>lc_FollowLane<br>PassingVehicle |
| S3 ClearOfPassedVehicle<br>SufficientReturnSpace | S4 AdjustReturnPassingParams()<br>lc_ChangeTo_RightLane<br>ReturningToOwnLane |
| S4 ReturnedToLane | S0 Done SetNormalDriveParams()<br>lc_Follow_Lane<br>FinishedPassManeuver |
| S2 MovingIntoPassingLane<br>NeedToAbortPass | S5 SetFallbackAbortPassParams()<br>AbortingPass |
| S5 OKtoReturnToLane | S6 AdjustReturnToLaneParams()<br>lc_ChangeTo_RightLane<br>AbortingPass_ReturningToLane |
| S6 ReturnedToLane | S1 SetNormalDrivingParams()<br>lc_Follow_Lane<br>AbortedPass_LookingToPass |
| S3 PassingVehicle<br>NeedToAbortPass | S5 SetFallbackAbortPassParams()<br>AbortingPass |

**Figure 3-3: State Table Representation of "Pass on Two-Lane Road"**

Once the FSMs have been encoded for each agent control module for all of the driving tasks, we have essentially represented the main decision processing knowledge set as many small groups of well ordered rules in an easily referenced (by task context and level of abstraction), and easily modifiable (each FSM can easily have additional rules added to it as additional alternate actions and their triggering situations are discovered) format.

The FSMs described above are used to encode the task decomposition knowledge. Each line of each state table uses some symbolic value to describe the present situation that must be matched in order to execute the corresponding output action of that rule. The processing required to evaluate that this particular situation is true can be thought of as a knowledge tree lying on its side, funneling left to right, from the detailed sensory processing branching until all of the values have been reduced to the one appropriate situation identification encoded in a symbolic value such as "ConditionsAreGoodToPass" (see Figure 3-4). This lateral tree represents the layers of refinement processing made on the present set of world model data to come to the conclusion that a particular situation now exists such as "ConditionsAreGoodToPass".



| PassVehInFront | |
|---|---|
| NewPlan | S1 SetPassOnTwoLaneRoadContext()<br>lc_Follow_Lane<br>LookingToPass |
| S1 ConditionsGoodToPass | S2 AdjustInitPassingParams()<br>lc_ChangeTo_LeftLane<br>MovingIntoPassingLane |
| S2 ConditionsGoodToPass | S3 AdjustPassingParams()<br>lc_FollowLane<br>PassingVehicle |
| S3 ClearOfPassedVehicle<br>SufficientReturnSpace | S4 AdjustReturnPassingParams()<br>lc_ChangeTo_RightLane<br>ReturningToOwnLane |
| S4 ReturnedToLane | S0 Done SetNormalDriveParams()<br>lc_Follow_Lane<br>FinishedPassManeuver |
| S2 MovingIntoPassingLane<br>NeedToAbortPass | S5 SetFallbackAbortPassParams()<br>AbortingPass |
| S5 OKtoReturnToLane | S6 AdjustReturnToLaneParams()<br>lc_ChangeTo_RightLane<br>AbortingPass_ReturningToLane |
| S6 ReturnedToLane | S1 SetNormalDrivingParams()<br>lc_Follow_Lane<br>AbortedPass_LookingToPass |
| S3 PassingVehicle<br>NeedToAbortPass | S5 SetFallbackAbortPassParams()<br>AbortingPass |

PLAN STATE TABLE

**Figure 3-4: World Model Data Dependencies**

The identification of these layers of knowledge processing to evaluate to the situation value is done in reverse. We know that we cannot change into the oncoming traffic lane (the "ChangeToLeftLane" action) during the passing operation until "ConditionsAreGoodToPass". Now we have to determine what are all of the things that have to be taken into consideration in order for this to be true. To determine this, many different example scenarios are reviewed to determine all of the pieces of knowledge required for all of these variations. The results are grouped by category into (in this example) five major evaluation areas. Thus, to be able to say that the "ConditionsAreGoodToPass", we first had to evaluate that each of the five sub groups were true, namely, the five situations of "LegalToPass", "EnvironmentSafeToPass", "SituationInFrontOKtoPass", "SituationInBackOKtoPass", and "OncomingTrafficOKtoPass", all had to be true.

In this example, we have clustered all of the rules of the road that pertain to the passing operation at this level of task detail into the “LegalToPass” sub group evaluation. We have itemized nine world states to be evaluated and we have named them with the identifiers such as “NoConstructionInPassZone”, “NoTransitOrSchoolBusStopping”, “NoPassZone-NotInEffect”, “LaneMarkingsAllowPass”, “NoIntersectionsInPassZone”, “NoRailroadXInPassZone”, etc.

These world states can now be further broken into the primitive world model entities we need to be able to measure (such as vehicles, their speed, direction, location, lane markings, signs, railroad tracks, etc.) in order to determine that these world states exist. These primitive world model entities then set the requirements for the sensory processing system we need to build to support these control tasks. Everything has been determined in the context of individual tasks we want the system to be able to do.

## 3.2. Metrics From The Task Decomposition Effort

Based upon preliminary work performed using the above analysis technique, we can estimate:

- the number of state tables that are necessary to capture all the behaviors that we wish the vehicle to execute,
- the number of situations that are needed to trip the actions in the state table,
- the number of world model states that must be true for a situation to be evaluated as true,
- the number of world model entities that must exist to be able to evaluate the world model states, and
- the number of attributes that must exist for the world model entities.

The current status of this effort is discussed later in this section and summarized in Table 3-3 in Section 3.2.6.

### 3.2.1. World Model States

As shown in Figure 3-1, there are **129 state tables (commands)** that are captured among all of the control modules in the task decomposition hierarchy. Each state table can be seen as a type of behavior that the vehicle must be able to exhibit while driving on-road. These behaviors are:

- **Into Journey Manager:** Do_Journey
- **Out of Journey Manager**: InitializeSystem, MakeVehOperational, ShutDownVehicle, TurnOffSystem, Goto_Destination, FollowVehicle
- **Out of Destination Manager:** InitializeSystem, StartUpVehicle, ShutDownVehicle, TurnOffSystem, GoOn__TurnRightOnto__, GoOn__TurnLeftOnto__, GoOn__Becomes__, GoOn__, StopAt__, FollowVehicle

- **Out of Route Segment Manager:** InitSubsystems, StartUpVehicle, ShutDownVehicle, TurnOffSubsystems, FollowRoad, CrossThru_Intersect, GoLeftTo__, GoRightTo__, Make_U_Turn, BackupTo__, RespondTo_OwnVehEmerg, Accommodate_SchoolBus, Accommodate_EmerVeh

- **Within Drive Behavior Manager:**
  - **FollowRoad:** PassVehInFront, DriveOnTwoLaneRd, DriveOnMultiLaneRd, PullOntoRoad, ChangeLanesToGoFaster, ChangeToGoalLane, AccommodatePassingVeh, RespondToFollowingVeh, NegotiateLaneConstriction, RespondtoPedestrian, RespondToBicyclist, RespondToVehEnteringLane, RespondToVehEnteringRoad
  - **CrossThru_Intersect:** CrossThru_StopSign, CrossThru_YieldSign, CrossThru_SignalLight, CrossThru_UncontrolledInter, MergeInto_TravelLane, AccommodateMerge, Negotiate_RRCrossing, Negotiate_TollBooth, Negotiate_PedestrianCross
  - **GoLeftTo:** TurnLeft_StopSign, TurnLeft_YieldSign, TurnLeft_SignalLight, TurnLeft_UncontrolledInter, TurnLeft_IntoDrive, TurnLeft_FromDrive, TurnLeft_IntoParkingSpace, ForkLeft, BackLeft_IntoLane, BackLeft_IntoDrive, BackLeft_IntoParkingSpace, BackOut_GoLeft
  - **GoRightTo:** TurnRight_StopSign, TurnRight_YieldSign, TurnRight_SignalLight, TurnRight_UncontrolledInter, TurnRight_IntoDrive, TurnRight_FromDrive, TurnRight_IntoParkingSpace, ForkRight, BackRight_IntoLane, BackRight_IntoDrive, BackRight_IntoParkingSpace, BackOut_GoRight
  - **Make_U_Turn:** Do_U_TurnAtIntersection, TurnAroundUsingDrive, TurnAroundInRoad
  - **BackupTo__:** BackupVehicle, ParallelPark

- **Out of Drive Behavior Manager:** InitSubsystems, StartUpVehicle, ShutDownVehicle, TurnOffSubsystems, FollowLane, PassOnLeft, PassOnRight, TurnRightTo__, TurnLeftTo__, StopAt, PullOff__OnLeftShoulder, PullOff__OnRightShoulder, GotoGap_LeftLane, GoToGap_RightLane, PreMerge_LeftLane, PreMerge_RightLane, ChangeTo_LeftLane, ChangeTo_RightLane, StopAtIntersection, AbortPass, CreepForward, PeekforPass, BackUp, BackOut_ToGoLeft, BackOut_ToGoRight, BackInto_FromLeft, BackInto_FromRight, DoUTurn_AtInter, DoUTurn_MidRoad, Do3Pt_Uturn, AllowVehToEnter_FromLeft, AllowVehToEnter_FromRight, YieldToPassingVehicle, ReactToPassingVeh, ReactToPassingVehAbort, PullOntoRd_FromLeftSh, PullOntoRoad_FromRightSh

- **Out of Elemental Maneuver Subsystem:** InitSubsystems, StartUpVehicle, ShutDownVehicle, TurnOffSubsystems, Follow_StLine, Follow_CirArcCW, Follow_CirArc,CCW

- **Out of Prim Trajectory to Steer Servo:** GoAt_SteerAngle,AngleVel,AngleAcc, InitializeSubsystem, PrepforStarting, PrepForShutDown

- **Out of Prim Trajectory to Speed Servo:** GoAt_Speed, Acc, Dir(Fwd/Rev), InitializeSubSystems, PrepforStarting, PrepforShutDown, MaintainForPark/Idle

### 3.2.2. Situations

Situations are shown on the left column of the state table, and indicate what has to be true about the world for an action in the state table to occur. In the effort, we estimate that there are, on average, seven situations per state table. Considering that we current have 129 state tables, that would result in approximately **1000 situations**. In the case of passing on a two lane undivided road, as shown in Figure 3-3, the situations are:

- Conditions Good To Pass
- Conditions Good To Pass In Passing Lane
- Cleared Of Passed Vehicle / Sufficient Return Space
- Returned To Lane
- Moving Into Passing Lane / Need to Abort Pass
- OK To Return To Lane
- Returned To Lane
- Passing Vehicle / Need to Abort Pass

In this case there are eight situations. In other state machines, there are often slightly more or slightly less. Overall, seven is shown to be a reasonable average among all of the state tables.

### 3.2.3. World Model States

World model states are individual states of the world that must collectively be true for an overall situation to be true. We estimate that there are, on average, 10 world model states per situation. Considering that we currently have approximately 1,000 situations, that would result in approximately **10,000 world model states**.

As an example, referring to Figure 3-4, in order for the situation "ConditionsGoodToPass" to be true, all of the world model states must evaluate to true, including:

- LegalToPass (which includes the world model states):
  - NoConstructionInPassZone
  - NoTransitOrSchoolBusStopping
  - NoPassZone-NotInEffect
  - LaneMarkingsAllowPass

    - NoIntersectionInPassZone
    - NoRailRoadOnPassZone
    - NoBridgeInPassZone
    - NoTunnelInPassZone
    - NoTollBoothINPassZone
- EnvironmentSafeToPass (which includes the world model states):
    - WeatherNotObscuringPassZone
    - RoadSplashNotSignificant
    - WindsNotSignificant
    - RoadSurfaceNotTooSlipperyToPass
    - RoadSurfaceSuitableToPass
    - OwnVehicleCapableToPAss
- SituationInFrontOKToPass (which includes the world model states):
    - NoHillBlockingSightInPassZone
    - NoCurveBlockingSightInPassZone
    - NoVehicleInFrontAttemptingLeftTurn
    - NoVehicleEnteringRoadInPassZone
    - VehInFrontNotBlockingSightINPassZone
    - NoPostalVehicleOrDeliveryVehicleMakingStops
    - NoPedestrianOnRoadSideInPassZone
    - SufficientReturnSpaceInFrontAfterPass
    - VehicleInFrontDrivingNormally
    - VehicleInFrontNotAttemptingToPass
    - NoPersonOnBikeInPAssZone
    - NoVehicleOnRoadsideReadyToComeIntoLane
    - NoActiveEmergencyVehicleInFront
- SituationInBackOKToPass (which includes the world model states):
    - VehicleInBackNotAttemptingToPass
    - VehicleInBackNotTailgating
    - VehicleINBackNotClosingRapidly
    - NoActiveEmeregencyVehiclesFollowing
- OnComingTrafficOKToPass (which includes the world model states):
    - NoAbnormalOnComingVehicleBehavior
    - SufficientTime/DistToAvoidOnComingVehicle

In this case there are 39 world model states. This represents one of the more complex state tables that was analyzed. Overall, 10 seems to be a reasonable average among all of the situations.

### 3.2.4. World Model Entities

World model entities are objects in the world that can be given a name and have attributes and state. For the most part, these are “physical things” that have geometric and dynamic properties and characteristics, and are either known *a priori* or can be detected by the sensors. Again referring to Figure 3-4, world model entities include:

- Own vehicle

- Construction
- School bus
- No passing zone sign
- Lane markings
- Pedestrian crossing
- Pedestrians
- Indicators of other road intersecting
- Railroad crossings
- Bridge
- Tunnel
- Toll Booth
- Weather visibility
- Splash
- Wind
- Road surface friction
- Road integrity
- Road visibility
- Vehicle in front
- Vehicle in front field of view
- Postal vehicle or delivery vehicle
- Road in front of vehicle in front
- Vehicle in front state
- Bicyclist
- Motorcyclist
- Vehicle on side of road
- Emergency vehicle
- Vehicle in back
- Vehicle following
- Oncoming vehicle

World model entities are ubiquitous, in the sense that they can be generally usable to determine multiple different world model situations or states. For example, the states of vehicles in front of you can be used to determine if it is safe to pass (as in Figure 3-4), but can also be used to choose an appropriate following distance One way to estimate the number of world model entities is to sum up all the unique world model entities among all of the state tables. Since all of the state tables and supporting world model states are not yet completed, one can only do a gross estimation based on progress to date. As such, we estimate that approximately 1000 world model entities need to be represented to enable on-road driving.

### 3.2.5. World Model Attributes

Attributes and states of world model entities can be computed from sensory signals, or can be predicted from *a priori* knowledge. In many cases, knowledge of the task defines what attributes need to be sensed. For example, referring to Figure 3-4, attributes of the "vehicle in back" that are important to know for this activity are the vehicle's:

- Position
- Speed
- Heading
- Acceleration/Deceleration
- Behavior
- Turn Indicators
- Headlights
- Horn
- Assigned Intent

On average, we estimate that there are approximately seven attributes of interest for each world model entity. Considering that we estimate that there are approximately 1000 world model entities of interest, that results in approximately 7,000 world model attributes.

### 3.2.6. Summary

**Salient Point: Table 3-3 summarizes our estimation of the number of state tables, situations, world model states, world model entities, and world model attributes we believe are necessary to enable autonomous on-road driving, as described above.**

| **Knowledge** | **Total Number** |
|---|---|
| **State Tables (behaviors)** | **129** |
| **Situations** | **1000** |
| **World Model States** | **10000** |
| **World Model Entities** | **1000** |
| **World Model Attributes** | **7000** |

**Table 3-3: Knowledge Summary**

## 3.3. Comparison to Capabilities in DEMO-III

Now that we have  estimated what knowledge is necessary to enable autonomous on-road driving, we will explore how much of this knowledge has been encoded in the DEMO-III effort described in Section 2 to determine where we are now and how far we have left to go.

Although DEMO-III is focusing on off-road driving as opposed to on-road driving, it is the authors' belief that many of the same underlying functionalities at the lower levels are fundamentally the same. In both case, the vehicle is recognizing objects, planning trajectory paths, and performing lane/path following. As such, the authors' feel that the DEMO-III effort serves as a reasonable benchmark to set time and funding estimates for implementing autonomous on-road driving.

It should be noted that although DEMO-III uses a cost-based planning approach as opposed to the finite state machine approach described in this section, it is still possible to draw meaningful correlation between the approaches by comparing the functionality that are able to be accomplished in each approach.

As mentioned in Section 2.4.2, much of the work exhibited in DEMO-III focused on waypoint following and trajectory generation. If we compare the DEMO-III capabilities to the state tables listed in Section 3.2.1, we can show that 10 out of the 129 commands have been implemented in DEMO-III. These 9 commands are shown below:

- **Into Journey Manager:** (none)
- **Out of Journey Manager**: (none)
- **Out of Destination Manager:** (none)
- **Out of Route Segment Manager:** (none)
- **Within Drive Behavior Manager:** (none)
- **Out of Drive Behavior Manager:** InitSubsystems, TurnOffSubsystems, FollowLane
- **Out of Elemental Manuever Subsystem:** Follow_StLine, Follow_CirArcCW, Follow_CirArc,CCW (note that these are all combined into one command in DEMO-III)
- **Out of Prim Trajectory to Steer Servo:** InitSubsystem, TurnOffSubsystem, GoAt_SteerAngle,AngleVel,AngleAcc,
- **Out of Prim Trajectory to Speed Servo:** InitSubsystem, TurnOffSubsystem, GoAt_Speed,Acc,Dir(Fwd/Rev), MaintainForPark/Idle

We can estimate that DEMO-III was able to accomplish about 8% (10/129) of the tasks that are needed to achieve acceptable behavior while driving on-road.

Now, if we look at the amount of time and money that have been put into DEMO-III to realize that 8%, we can estimate that there has been approximately 10 calendar years of effort at a funding level of approximately 30 million dollars, fairly evenly split between the efforts of General Dynamic Research Systems (GDRS) and NIST. This is only the money that has been applied to the vehicle navigation system, not what has been applied to building the hardware for the vehicle. Assuming that all commands are at equal level of complexity, namely, that the effort needed to realize the command is equivalent for all commands, then if 30 million dollars gets you 8% of the way there, that it would take between 350 and 400 million dollars to get you 100% of the way to achieving acceptable behavior while driving on-road.

Current funding for Army autonomous mobility programs at ARL and TACOM total approximately $50M per calendar year. This funding covers many projects and only a part of it is targeting the problem this report addresses. Funding specifically for autonomous navigation is of the order of $15-20M per calendar year. The conclusion is that, if current funding is continued, it will take more than twenty calendar years to reach intelligent driving capability.

The Future Combat System (FCS) Autonomous Navigation System (ANS) effort is the ultimate target for autonomous driving capability. This effort is funded at 145 million dollars over four years, which corresponds to about 35 million dollars per calendar year. A major caveat, however, is that most of the $145 Million will go toward hardening already proven capability, not advancing the state of the art, since the ANS procurement specification only requires supervised teleoperation (which was demonstrated more than a decade ago under the Demo II program) with autonomy as a goal, not a requirement. If the goals for autonomous driving are to be achieved, other programs must be funded by DARPA and the Army.

**Salient Point: Based upon the functionality achieved in DEMO-III and the driving task analysis performed by NIST as part of the DARPA MARS project, we estimate that it will take approximately 300 to 400 million additional dollars to achieve acceptable autonomous driving behavior, which would take 20 calendar years or more based upon current funding levels.**

## *3.4 Comparison to Current Status of Task Decomposition Effort*

As mentioned earlier, we have only begun to explore all of the knowledge that is necessary to enable acceptable on-road-driving. Table 3-4 compares what has been accomplished to date against what is necessary to completely capture the knowledge for acceptable on-road driving.

| **Knowledge** | **Total Number** | **Completed To Date** | **Percentage Completed** | **Time to Complete the Effort** |
|---|---|---|---|---|
| State Table | 129 | 60 | 46% | 0.5 person-month |
| Situation | 1000 | 500 | 50% | 0.5 person-month |
| World Model State | 10000 | 500 | 5% | 1 person-year |
| World Model Entity | 1000 | 100 | 10% | 0.25 person-year |
| World Model Entity Attribute | 7000 | 200 | 3% | 0.25 person-year |
| World Model Entity Attribute Sensor Resolution Specification | 7000 | 5 | 0.1% | 0.5 person-year |
| Total | | | | 2 person years |

**Table 3-4: Knowledge Capture Summary and Progress To Date**

It is important to note that the goal of this effort is to determine the knowledge that is necessary to capture to enable autonomous on-road driving, not to implement this knowledge on the vehicle itself. This is primarily a research effort as opposed to an engineering effort.

With that in mind, we can approximate, from a research perspective, how long and how much money it will take to complete this effort. Up until the time this paper was written, the task decomposition effort has been funded at a level of approximately $350K over the course of two calendar years. As shown in Table 3-3, two persons are needed to complete the task decomposition effort. At a loaded salary of $250K per person, that results in a necessary funding level of $500K.

**Salient Point: It will take two person years and $500K in funding to complete the task decomposition effort in order to determine all of the knowledge that is necessary to capture to enable autonomous on-road driving. This will provide the detailed requirements for the perception and world modeling capability needed for intelligent autonomous driving and will in an of itself provide the structure of the behavior generation side of the control hierarchy. Enough has been done to identify the requirements for the next generation of sensors; these requirements are presented in the Section 4.**

## 3.5 Comparison to Current Status of the Cost-Based Search Effort

In addition to the finite state machine-based approach mentioned earlier in Section 3, cost-based planning represents another popular approach to controlling autonomous vehicles.

The cost-based planning system that is currently used by the autonomous vehicle for on-road planning is an implementation of the incrementally created graph planning approach developed by Balakirsky [3]. As in many planning algorithms, this algorithm incorporates a graph search algorithm that strives to find the cheapest path through a graph that is composed of nodes (representing system states) connected by edges (representing system actions). The cost of a path through the graph is defined as the sum of the action costs (the edges) plus the costs of having occupied the traversed states (the nodes). It is these costs that must be developed in order to achieve human-level driving performance.

One such graph search algorithm is Dijkstra's shortest path algorithm [8]. An example of this algorithm is shown in Figure 3-5 and may be summarized as follows:

1) Initialize the search. This includes setting the initial cost of all nodes (in the figure nodes are shown as circles and node costs are the bold numbers next to them) to infinity, and creating a set of *open* nodes that only contains the goal node ($n_g$) at a cost of zero. An *open* node is a node that the search has reached but not evaluated. Nodes that have been fully evaluated are shown as bold circles in the figure.

2) Find the least expensive member of the *open* set (denote this node by $n_{cheap}$) and remove it from the *open* set.
3) Compare $n_{cheap}$ to the start node ($n_s$). This search proceeds from the goal to the start, so if $n_{cheap}$ is equal to the start node the search is finished. It can be noted that this search may also proceed from start to goal without loss of generality.
4) Expand $n_{cheap}$. During this step, the cost of reaching each of $n_{cheap}$ 's predecessors (nodes connected by lines in the figure) must be determined. The following steps occur for each predecessor:
    a. Determine the cost of the edge that connects $n_{cheap}$ to the predecessor and the cost of occupying the predecessor.
    b. If the sum of these two costs plus the cost of $n_{cheap}$ is less then the current cost of the predecessor, the edge is maintained as a forward pointing edge (set to bold in the figure), any previous forward pointing edge is removed, and the predecessor is added to the *open* set.
5) Go to step 2.

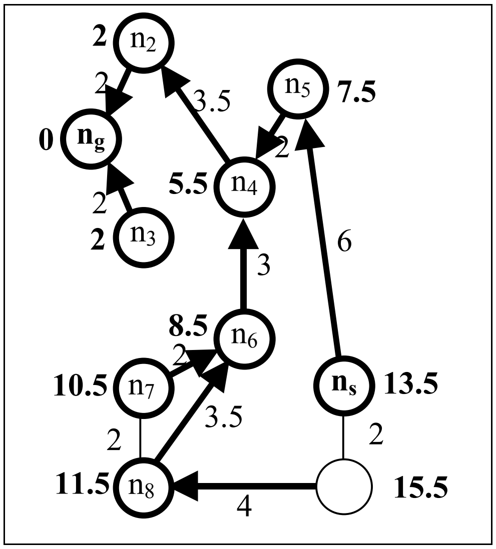


**Figure 3-5: Example of Dijkstra graph search.**

An example of this algorithm's application is shown in Figure 3-5. The optimal path from any expanded node to $n_g$ lies along the decreasing cost path of bold edges (follow the arrows). For this example, the search proceeds from the node labeled $n_g$ to the node labeled $n_s$. The search terminates at the optimal answer when the node $n_s$ is examined for expansion. The optimal path found may be seen to be $n_s - n_5 - n_4 - n_2 - n_g.$

While cost-based algorithms may differ in how they place and connect their planning nodes, they must all perform the above-described search. As seen from the algorithm description, each loop of the algorithm must make multiple calls to a cost generating function (step 4a). A single plan may entail several hundred or even thousands of algorithm loops, and the cost generator is at the heart of the loop, making its performance critical.

This cost function and the overall planning framework has been developed for basic road driving. As of the time that this paper was published, the cost-based system was capable of planning routes on/with:

- All type of roads including straight and curved lane segments
- Any number of lanes on a roadway
- Uni-directional and bi-directional traffic
- Multiple classes of static objects
- Moving objects in the environment assuming that the trajectory can be probabilistically determined
- Approximately 12 cost factors, including speed limit conformance, proximity with static and dynamic objects, conformance to lane markings, and abiding by a small set of the rules of the road.

Enhancements to the planner that are expected to be accomplished in the next 12 months include:

- Ensuring that the planner can run in real-time through the introduction of a vehicle executor
- Dealing with intersections, initially with traditional 4-way intersections and then with more complicated intersections that include exit and entrance ramps, etc.

In this analysis, we make the following assumptions:

- While enhancements to the basic infrastructure still need to be made (for example the inclusion of intersections), this work is trivial compared to the development time/effort for the cost model. In other words, the representation and implementation of the costs within the cost model is the primary indicator of the time and effort it will take to enable on-road driving
- The factors in which we apply costs are roughly the same as the 1,000 world model entities that were described in Section 3.2.3.
- Only 60% of the entities that are not already captured need to have a true weight associated with them. The rest of the entities will have a yes/no type of value, such that if the state evaluates to true (e.g., a non-traversable object in the arc being evaluated), it will have an extremely large cost, thus prohibiting the connected node from ever being evaluated independently of the rest of the environment. If the state evaluates to false (i.e., it does not exist), it will have zero cost. The entities that have a yes/no type require very little work to model while the entities that need to be captured by a weight require a more significant amount of work.
- The time and effort to encode variable cost attributes increase at a squared-rate as the number of variable cost attributes increase. This is mainly due to the relationship between individual variable cost attributes, which increase as the number of variable cost attributes increases.

Based on the above assumptions, there are 1,000 entities that would need to be represented in this approach. Using our assumption that only 60% of them would need to have true weighting values associated with them (while the others only need to be indicated with a yes/no value by including a very large cost), that would leave 600 states that would need costs associated with them.

Considering that it took approximately three working days to associate costs with the existing 12 states, and that this time would grow in a squared fashion as more states were introduced, it would take approximately 20 person-years to capture all of the necessary costs to enable intelligent on-road driving.

# 4.0 Sensors

The task decomposition described in Section 3 assumes the availability of sensors and sensory processing systems that work at a specified level such that the vehicle control system can recognize objects, and characteristics of objects, and then make appropriate decisions based upon what it sees. The task decomposition effort has progressed to the point that the requirements on the sensors and sensory processing software can be specified, as described below.

## 4.1. Requirements of Sensor Resolution For On-Road Driving

In this section, we will look at some detailed examples of requirements for sensory processing, following through with our passing example described in Section 3.0. In particular, we will look at what it required of the sensors on the vehicle to determine, at any given time and speed, if it is legal to pass.

As shown in Figure 3-4, in order for a passing operation to be legal, there cannot be:

- Any construction in the passing zone,
- A transit or school bus stopping in the passing zone,
- A no passing zone sign in the passing zone,
- Lane marking that prohibit passing
- Intersections in the passing zone
- Railroad crossing in the passing zone
- A bridge in the passing zone
- A tunnel in the passing zone
- A toll booth in the passing zone

Therefore, the sensory processing system must detect these items, or indicators that these items are approaching, at a distance that allows the vehicle to pass safely. In this analysis we make a few assumptions:

- The vehicle can accelerate comfortably at 1.7 m/s$^2$
- Our vehicle is positioned approximately one second behind the vehicle in front of it (i.e., our vehicle will be at the preceding vehicle current position in one second traveling at constant velocity)
- Our vehicle will begin merging back into its original lane when it is one car length in front of the vehicle it is passing
- The merging operation which brings the vehicle back into our vehicle's original lane will take one second
- The average length of a vehicle is 5 meters.

With these assumptions, we explored what distance our vehicle would travel during a passing operation, how long it would take to travel that distance, and what the final velocity of the vehicle would be assuming initial speeds of 13 m/s (30 mph), 18 m/s (40 mph), and 27 m/s (60 mph). We limited the vehicle to traveling no faster than 9 m/s (20 mph) faster than its original speed when starting to pass at the higher speeds. Table 4-1 shows the results.Table 4-1 shows the results.

| Speed (m/s) | Time to Complete Pass (s) | Distance Traveled in Pass (m) | Final Velocity at End of Pass (m/s) |
|---|---|---|---|
| 13 | 6.3 | 120 | 24 |
| 18 | 6.8 | 160 | 27 |
| 27 | 7.8 | 260 | 36 |

**Table 4-1: Pertinent Values for Passing Operation at Various Speeds**

Note that in this analysis we are assuming un-occluded visibility.

Assuming on-coming traffic is moving at the same speed, the sensor must detect on-coming vehicles at 2x the distance traveled in passing.

If we look at the “ no railroad crossing in passing zone” requirement, we note that there are multiple markings that can indicate a railroad crossing is upcoming, such as a crossbuck just before the railroad crossing, or railroad signs at pre-defined distances before the railroad crossing. Table 4-2 shows the specification on how far before a railroad crossing a warning signs should be placed, what sign the size must be, and what the size of the letter on the signs must be, according to the Manual of Uniform Traffic Control Devices (MUTCD) [17].

| Speed (m/s) | Distance from Railroad Crossing (m) | Sign Dimensions (m x m) | Letter height (m) |
|---|---|---|---|
| 13 | 100 | 0.450 x 0.450 | 0.125 |
| 18 | 145 | 0.450 x 0.450 | 0.125 |
| 27 | 235 | 0.450 x 0.450 | 0.125 |

**Table 4-2: Specifications for Railroad Crossing Signs**

Considering that the railroad warning sign is a pre-defined distance before the railroad crossing, we can subtract that distance from the full passing distance shown in Table 4-1 to identify the distance forward our sensors need to be able to see. This resulting distance is showing Table 4-3.

| Speed (m/s) | Passing Distance (m) | Warning Sign Distance (m) | Sensor Sign Distance (m) |
|---|---|---|---|
| 13 | 120 | 100 | 19 |
| 18 | 160 | 145 | 14 |

| | | | |
|---|---|---|---|
| 27 | 260 | 235 | 18 |

**Table 4-3: Sensor Sight Distance for Railroad Warning Sign**

This sets the specification for how far a sensor must be able to “see” to determine if there is a railroad crossing sign in the passing zone. However, we can take this one step further and determine what the resolution of the sensors must be to read the sign.

If we assume that the sign needs to be read (e.g., we do not know what the sign indicates based on its shape and/or color), and that for each letter in the sign, we need a 20x20 array of pixels hits on that letter to be able to recognize the letter. Using simple trigonometry based upon the distance to the sign and the size of the letters on the size as shown in Table 4-2, we can show that we need a camera that has resolutions of about 0.02 degrees for all three cases above.

In some cases, a warning sign is not present and the sensors must rely on recognizing a crossbuck that is immediately before the railroad crossing. In this case, we assume that we need an array of 5 x 5 pixel hits on the crossbuck to recognize it by shape, and that the size of the crossbuck is the standard 900 x 900 mm total dimensions, as specified by the MUTCD manual. Based on this information, we would need a sensor with a resolution as shown in Table 4-4 below.

| **Speed (m/s)** | **Sensor Resolution (degrees)** |
|---|---|
| **13** | **0.0875** |
| **18** | **0.0649** |
| **27** | **0.0404** |

**Table 4-4: Sensor Sight Distance for Crossbuck**

Analysis of several other driving scenarios show that the figures in Table 4-4 are fairly representative of the sensor resolution which is necessary for on-road driving.

## 4.2. Next Generation LADAR

One of the primary sensors we expect to be most valuable in on-road driving is LADAR. The LADAR used in Demo III, as described in Section 2 above, is clearly inadequate in resolution and does not have the range required for full speed highway driving. A next generation of laser range sensors has appeared on the market in the past two years, with approximately ten times the speed (600,000 points per second) and much better range (beyond 100 m). Figure 4-1 shows a typical scene. This is a very high resolution scan which takes many seconds, but the same technology could produce a 256 x 256 range image at 10 frames per second or better.

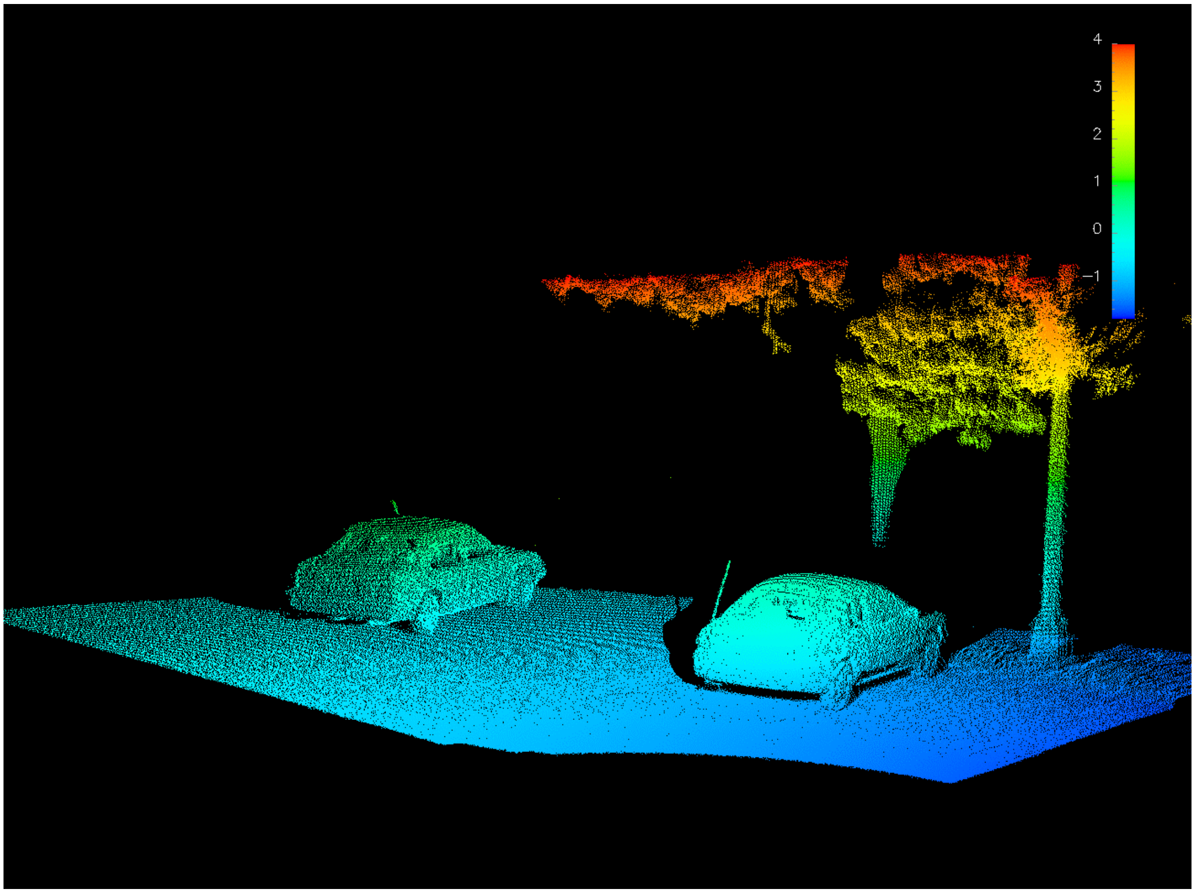


**Figure 4-1 High Resolution LADAR Image.Range to the nearest car is about 7 meters.**

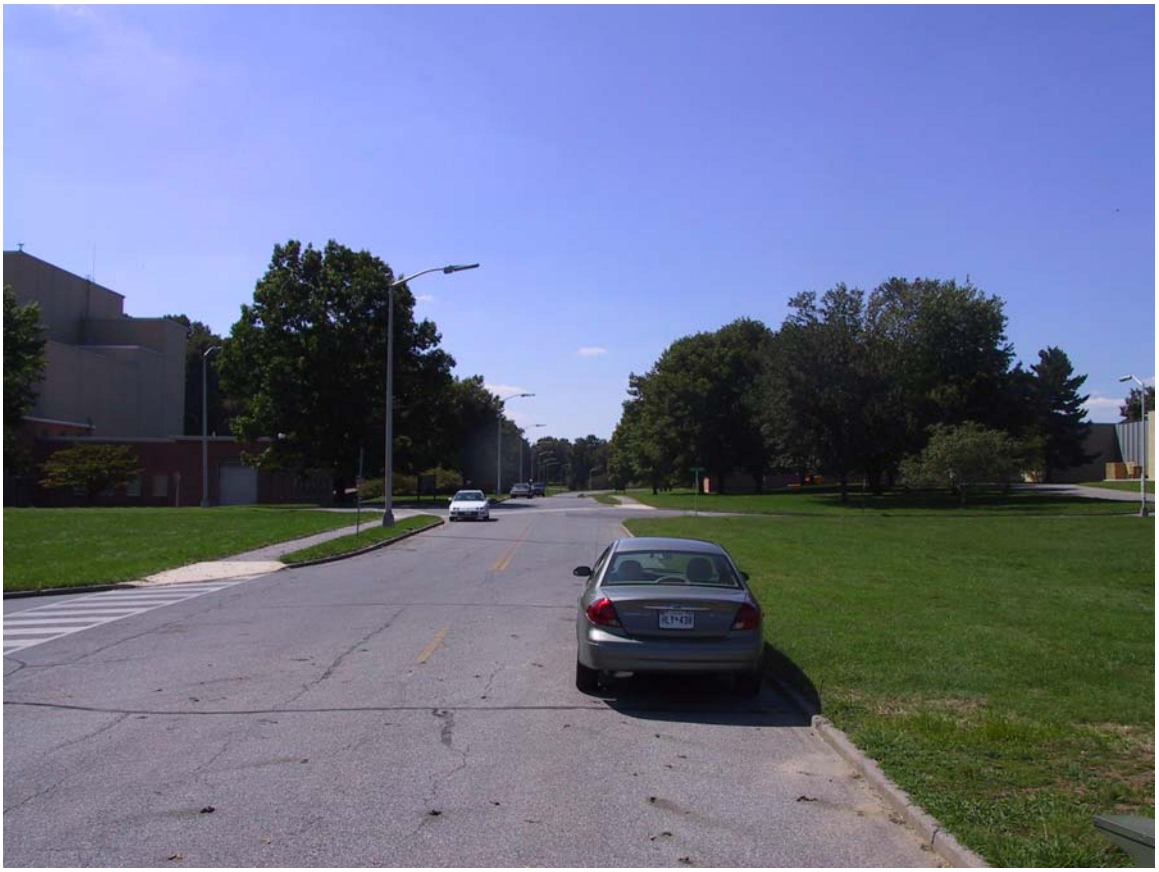


**Figure 4-2. A CCD picture of the road ahead. The car directly in front is 10 meters away. The white car in the on-coming lane is 50 m away. The car behind it is 100 m away, and the car behind it is 150 meters away.**

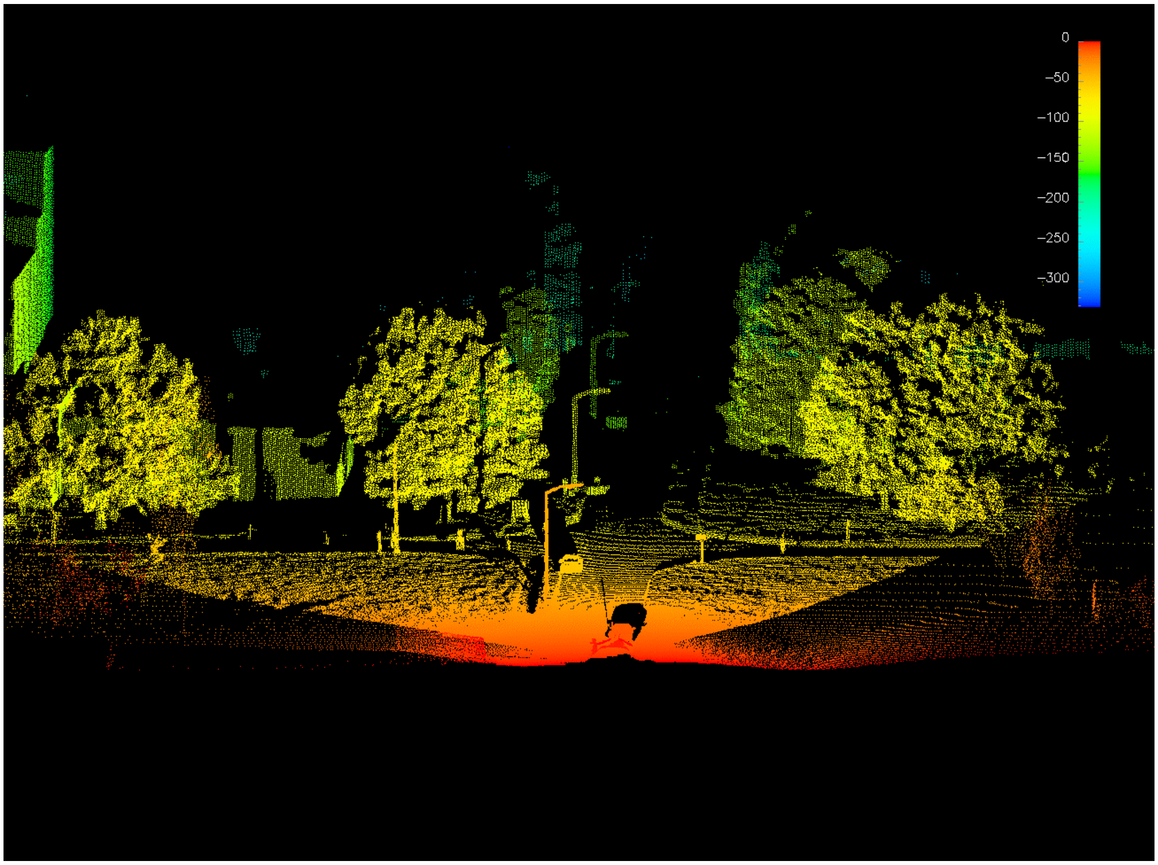


**Figure 4-3 A LADAR point cloud taken from the same position as the photo in Figure 4-2. The image is color coded for distance. Red is zero. Green is about 175 m. See scale at top right. Note the four cars. The car at 150 m is clearly visible. Returns from the ground disappear at about 75 m.**

Based upon experience from DEMO-III and a survey of available technology, a Broad Agency Announcement (BAA) was released in June 2002. Phase 1 of the BAA focused on the design of a LADAR for on road driving with the specs shown in Table 4-5.

| **Sensor Type** | **Range Resolution** | **FOV Vert and Horiz** | **Resolution – Vert and Horiz** | **Ground Range** | **Vertical Surface Range** | **Scan Rate** | **Stabilization *** |
|---|---|---|---|---|---|---|---|
| **Wide FOV LADAR** | 5-10 cm or better | About 40 x 90 degs | 0.25 - 0.3 degs or better | 40-50 m or better | 125-200 m or better | 10 frames /sec or better | 0.3 deg |
| **Narrow FOV LADAR** | 5-10 cm or better | About 5 x 5 degs | 0.05-0.06 degrees or better | 40-50 m or better | 125-200 m or better | 10 frames /sec or better | 0.03 degs |
| **Wrap around** | 10-15 cm | About 0.5 x | 0.5 x 0.5 degs | N/a | 50 m | About 10 | N/a |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **LADAR** | | 360 degs | | | | frames /sec | |

**Table 4-5: Next Generation LADAR Specifications**

In addition to these specifications, the LADAR must also:

- Operate in full sunlight
- Be eye-safe
- Be capable of penetrating dust, fog, smoke, grass and light foliage
- Be small sized, low cost, and ruggedly designed

Based on the BAA, four Phase 1 awards were made and the results of these awards have been reviewed. Phase 2 awards, focusing on the development of the LADAR, are pending the availability of funds. ***Based upon the four award results, it is estimated that a prototype of a LADAR with the above specifications will take anywhere from 16-30 months to manufacture and cost between one and three million dollars.***

## 4.3 Next Generation Vision Systems

Similar to the LADAR specifications above, the Table 4-6 are the specifications for camera systems that we believe can be implemented with currently available commercial technology within the next 24 months at a cost of less than one million dollars.

| Sensor Type | FOV Vert and Horiz | Resolution – Vert and Horiz | Scan Rate | Stabilization |
|---|---|---|---|---|
| **Wide FOV camera** | About 21 x 28 degs | 0.1 degs or better | 10 frames/sec or better | 0.1 deg |
| **Narrow FOV camera** | About 2 x 2 degs | 0.01 degrees or better | 10 frames/sec or better | 0.01 degs |
| **Wrap around camera** | About 90 x 360 degs | 1.0 degrees or better | About 10 frames/sec | N/a |

**Table 4-6: Color Camera Specifications**

The importance of high resolution foveal vision should be emphasized as a good solution to the resolution/processing load trade. For example, the MARS work on reading road signs shows that you need high resolution to be able to read road signs, and that means the signs get quite close before they are legible if you have a single fixed resolution camera. High resolution in only a (steerable) part of the field of view would allow signs to be read at a much greater distance. As another example, consider Dickmann’s camera configuration with a high resolution central field of view and multiple cameras providing peripheral fields of view. The view of the central fields of view are shown in the figure below. Note how difficult it is to really see any detail in the low resolution image but how the high resolution image provides detail but lacks any context. The two scenes together make the highway scene understandable.

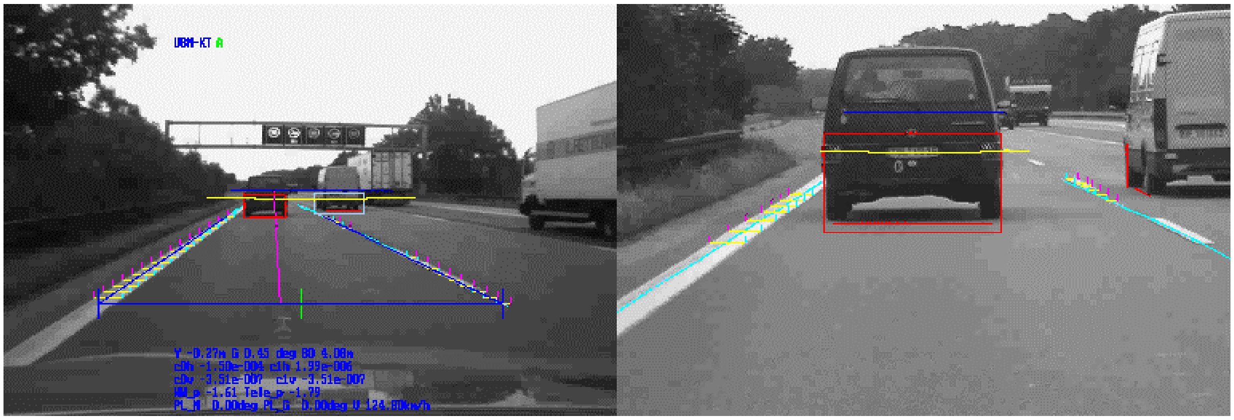

UBM-KT A
Y -0.27m G 0.45 deg B0 4.08m
c0h -1.50e-004 c1h 1.99e-006
c0v -3.51e-007 c1v -3.51e-007
WW_p -1.61 Tele_p -1.79
PL_M 0.00deg PL_G 0.00deg V 124.80km/h

**Figure 4-2: Foveal/Peripheral Camera Views from Autonomous Driving Program at Universitat de Bundeswehr (Munich, Germany)**

## 4.4 Comparison with Requirements

In this section, we compare the required sensor resolution that we derived from the task decomposition effort in Section 5.1 to LADAR and vision specifications expected to be available in the next 16-30 months as described in sections 5.2.and 5.3. Table 4-7 shows the results.

| Speed (m/s) | Needed Resolution Based on Task Decomposition (Sect 5.1) | Expected LADAR Resolution – Narrow FOV (Sect 5.2) (degrees) | Expected Camera Resolution – Narrow FOV (Sect 5.3) (degrees) |
|---|---|---|---|
| 13 | 0.1042 | 0.05 | 0.02 |
| 18 | 0.0711 | 0.05 | 0.02 |
| 27 | 0.0406 | 0.05 | 0.02 |

**Table 4-7: Comparison of Needed and Expected Sensor Resolution**

As shown in Table 4-7, it appears that the needed resolution from both the vision and the LADAR sensor should be available within the next 16-30 months assuming that funding becomes available to pay for the required development effort.

## 4.5 Sensory Processing

Perception is currently seen as the major roadblock to autonomous mobility. In order to make progress, the focus of perception research for autonomous vehicles needs to change, and the resources allocated to it must be increased substantially.

Sensory processing needs to undergo major changes, not so much to the basic algorithms and low level processing, but in the way these procedures are applied to sensory data and how sensing interacts with planning and execution modules of an autonomous vehicle. Sensing and sensory processing must become highly active, involve multiple cooperating and competing processes, be intentional and focused, and be inherently error tolerant. While the basic sensory processing algorithms will not change much, the way they are applied and combined will be fundamentally different. Even with greatly increased processor speeds, applying the algorithms to all the sensory data all the time will not be feasible. Focusing attention and using temporal as well as spatial characteristics of the data will be essential for successful perception.

For successful understanding of the environment around the vehicle, multiple sensors and sensory modalities will run in real time on the moving vehicle and will return fused, time-stamped data. Sensors will be active in the sense of being pointable and zoomable (both of which could be accomplished without moving parts if the sensors have enough pixels). Sensory processing will include sensor control, tied to the intentions of the vehicle (e.g., looking for signs, markings, and other vehicles that impact the behavior of the vehicle). The sensors will need to attach position information to each data sample.

Real-time segmentation will be carried out on each image based on color, texture, range, and other features, giving a vector of characteristics at each spatial location (or each pixel if the pixel has no range information). Sensory processing will be carried out both on individual images and on combined image data (maps). Information will flow between these two kinds of processing, reinforcing or attenuating their results. Attention will be focused on subsets of sensory data according to the state of the vehicle, and sensory processing algorithms will be selected according to the information needed. Thus, a sensor may be pointed to the side of the street to look for signs giving the speed limit of the current stretch of road. Sign detection and number detection algorithms would be applied to regions of the images that correspond to the expected height of the signs. Similarly, other features of the data will be isolated and tracked. New basic algorithms will be developed as needed and tailored to the domain, but in most cases modest improvements in image-processing techniques will be sufficient.

A range of objects will be recognized in real time, depending on the task being carried out. These will include stationary objects like signs, markings, telephone poles, parked cars, etc., and moving objects such as pedestrians or other vehicles. Sensory processing will try to identify stationary objects to determine what information they provide. Sensory processing will try to identify moving objects, compute their relative velocities and accelerations, and determine time to contact. Recognition will be of two sorts: recognition based on expectation (top down), and recognition of unexpected objects or aberrant situations (bottom up). For example, recognizing road signs could be an ongoing subtask that would scan selected locations of a sensor's field of view and apply templates taken from the manual of road signs. For most unexpected situations, motion detection and fast but simple processing of the entire sensory data would be applied, with regions that appear aberrant being added to the list of regions for attention. As an example, construction activity or a multi-vehicle wreck may not be recognizable from data in the a priori knowledge base, but they must be sensed and placed in the world model as objects in the roadway that must be avoided.

At higher levels, the sensory processing will attempt to build situation awareness. This will require sensory processing modules to be tightly linked with the planning and execution modules. Sensing will be driven by the intentions of the system and the associated knowledge requirements, which can usually be known a priori. It is well known that humans modify their eye fixation patterns depending on the task they are trying to accomplish. A similar mechanism will be necessary to enable the situation to be understood rapidly enough for interaction.

At all levels, a major factor in sensory processing will be the use of range information as well as color, texture, etc. Knowing range to an object allows recognition to be based on actual size and surface shapes instead of just coloring or texture. This makes many operations relatively simple (e.g., segmentation, recognition). Another major factor will be communication between different levels of the sensory processing and world model hierarchy, and with the planning and execution modules. This will affect which algorithms are applied to which sensor data and the confidence in their results.

All of the processing will need to take place in bounded time, although the bounds will depend on the type of processing. Measuring color or texture, for example, would take place at the input rate at which sensor data, whereas object identification would be needed at the rate at which decisions about objects are made. Confidences will be associated with each measurement, and will be adjusted over time as new information about each feature become available.

Overall, there will be a movement in sensory processing away from pure bottom-up processing to top-down and bottom up processing tightly coupled to planning and execution. As the goals of the vehicle change, the algorithms applied to sensory data will change and the interpretation of the environment may also change.

***Clearly there is a great deal of work to be done in model based perception. A new generation of sensors is the starting point for attacking this problem. While prototypes of next generation sensors have been estimated at $3-4 Million over 2 to 3 calendar years, the total engineering effort in achieving refined, field tested and hardened deployable versions will take up to a decade and will cost $20-30 Million. The software engineering effort is at least twice that great and probably more. Achieving the required level of perception is a decade long effort costing in excess of $100 Million.***

# 5.0 Computer Processing Capability Analysis

The benchmark for intelligent systems is human levels of performance. The question arises as to when we will achieve such levels of performance in autonomous driving. This is an issue of direct importance to military force planning and to highway safety.

Many researchers have pointed out the there are useful levels of performance well below true human levels of performance. The Future Combat Systems "Mule" and convoying capability would be examples. Still, for the purposes of technology forecasting, it is useful to ballpark human computational capabilities.

One necessary precursor of achieving the goal is adequate computing power. Dr. Doug Gage argues that computing power is not the principal problem, that even if we had the necessary computing power we wouldn't know what to do with it. This report lays out one specific research approach to autonomous driving that has had significant early success. The authors believe that this approach will prove viable in achieving human levels of performance at some point in the future.

Research to date has indicated the need for massive computing power to provide the necessary perception and world modeling capabilities for autonomous driving, well beyond the levels employed to date. In attempting to ballpark resources and time scales to reach minimum levels of human equivalent performance in autonomous driving, it is necessary to quantify what levels of computing are needed.

To reverse Dr. Gage's argument, if researchers had functional software for autonomous driving, transported magically from the future, they would only be able to test and demonstrate that software if they had appropriate computers to run it on. This chapter attempts to ballpark what levels of computing power might be needed to run such software.

## 5.1 Global Estimates

Several authors have addressed this issue, with greater or lesser credibility and generating greater or lesser levels of hostility from those who disagree with them. Several quantitative assessments stand out among recent books:

Ray Kurzweil [12] argues that there are $10^{11}$ neurons in the human brain, with an average of 1000 synapses per neuron, and that each synapse can perform approximately 100 computations per second. He thus concludes one needs $10^{16}$ computations per second to equal the performance of the human brain, and, by Moore's Law, predicts that desktop computers will reach this level by approximately 2025.

James Albus [2] modifies this calculation by noting that there is massive redundancy in neural circuits (since memory representations are distributed and to cope with noise and attrition of neurons over time). Using a factor of 100 to 1000 for redundancy, the equivalent processing power of the human brain is of the order of $10^{13}$-$10^{14}$ computations per second.

It can further be argued that the computational power of one synapse is somewhat less than one byte. Current computers are reaching 64 bit word lengths, and 128 bit word lengths can be expected in the future. Thus, current computers are crunching 8 bytes with each computational cycle and in the future will operate on 16 bytes in each cycle. One can therefore argue that computers only need to achieve $10^{12}$-$10^{13}$ computations per second to match the computational processing power of the human brain.

Churchland and Sejnowski[6] estimate $10^{12}$ neurons in the brain, an order of magnitude larger than Kurzweil and Albus. That would give an estimate of $10^{13}$-$10^{14}$ computations per second to match computational processing power.

None of these sources cite any definitive reference studies and all use scaling from typical neuron densities in the cerebral cortex, which varies, and layering also varies, so an order of magnitude estimate seems to be the best one can do. Grossberg [10]argues that the number of neurons is not a useful measure of computational power, that instead it is the local processing architecture that is the key to effective neuronal computing.

Moravec [15] makes a more interesting calculation. He points out that the retina does edge and motion detection computations for each of $10^6$ pixels at a rate of about 10 times per second. He then notes that we know how to duplicate these calculations on a computer. It takes 100 calculations to do run spatial and temporal filters for one pixel, so the computer processing equivalent of the retina is

$$10^6 \text{ pixels x 100 instructions/pixel x 10 /second} = 10^9 \text{ instructions/second}$$

He then takes the ratio of the number of neurons in the cerebral cortex divided by the number of neurons in the retina, which is about $10^5$, and concludes that the total processing power of the brain is $10^9$ x $10^5$ = $10^{14}$ instructions/second.

Moravec argues that redundancy in the cortex should be comparable to redundancy in the retina. He does not address computer word length.

The above citations point to a range of estimates of the processing power of the human brain in the range of $10^{12}$-$10^{14}$ instructions per second.

An interesting additional benchmark is provided by Big Blue, the special purpose computer that beat Garry Kasparov at chess. Big Blue had an equivalent processing power of 3 x $10^{12}$ instructions per second. This was superhuman performance in one small domain of human endeavor, but one that is considered important in terms of strategic planning abilities. Quite interesting was the fact that Big Blue used cost based search and stored patterns to evaluate moves; these are basically the strategies used in path planning in Demo III.

Clearly the task of driving does not take the entire computational capability of the human mind at all times since it is possible to drive and daydream, listen to the radio, talk on a cell phone, eat, talk, plan, and any of numerous other simultaneous tasks. Some of these clearly distract the driver in an unsafe manner, leading to legislation restricting the use of handheld phones, for example. However, when totally focused on new and unusual or difficult driving situations, or in bad weather or emergency situations, a good driver is totally focused on the task at hand.

Perception is the most compute intensive task in routine driving. Visual processing accounts for some 10-20% of the visual cortex, auditory processing another 10% and motor control about 10%. Add to that some level of planning and symbolic reasoning needed for following traffic laws and analyzing various road situations and a level of 50% or so of the total computational capability of the brain might be employed, on an intermittent basis, in driving.

If we expect robot vehicles to be always focused on the task at hand and not subject to distraction, then we will need to be at least within an order of magnitude of the computing power of the brain to achieve human levels of performance.

A significant advantage in computing power for robot vehicles comes from the use of LADARs for range imaging. The mammalian visual system commits large amounts of processing to processing stereo images to obtain depth information; LADARs deliver that information directly from a single image. So there may be some reduction in processing needed for robotic driving, perhaps a factor of two in perception processing.

We thus argue that $10^{11}$ instructions per second would be a good estimate of the lower end of the computing power needed (an order of magnitude below the lowest level argued above), and $10^{14}$ would be a highest end estimate (the highest level above).

Again making the argument that useful levels of performance will be achieved well before full human levels of performance, then $10^{11}$-$10^{12}$ instructions per second seems like a best estimate target range for minimally sufficient computing power for good autonomous driving.

## 5.2 Moore's Law

Gordon Moore, one of the inventors of the integrated circuit and founder and Chairman of Intel, noted in about 1970 that the number of transistors on a chip was doubling every eighteen months[4]. This was an observation of manufacturing efficiency using ever better lithography process technology. Since the cost of a chip is more or less constant, the implication is that you get twice as much computing power per dollar every 18 months.

[4] Moore's original estimate was a twelve month doubling period; apparently he revised that to twenty-four months some ten years later. An eighteen month doubling period has been widely used as "Moore's Law" since the 1970's. Actual doubling periods have ranged between twelve and twenty-four months.

Moore's Law has held true for more than three decades. In fact, the doubling period has been decreasing and was approximately twelve months between 1995 and 2002 before lagging this year.

Sources in the semiconductor industry have predicted the end of viability of current lithography techniques for manufacturing ever more powerful chips by 2020 at the latest. Moore's Law is expected to hold true for at least this decade, however. Other approaches to computing, including quantum computing, optical computing and molecular electronics, are subjects of active research and may become viable as lithography reaches its twilight years.

Both Kurzweil and Moravec present graphs of computing power (per thousand dollars) and note that there is a more or less continuous curve over the past one hundred calendar years! That period covers five different computing technologies: mechanical, electro-mechanical, vacuum tube, discrete transistor, and integrated circuit based computers. Even more interesting is that the slope of the curve increases over time: this is a growth rate faster than exponential.

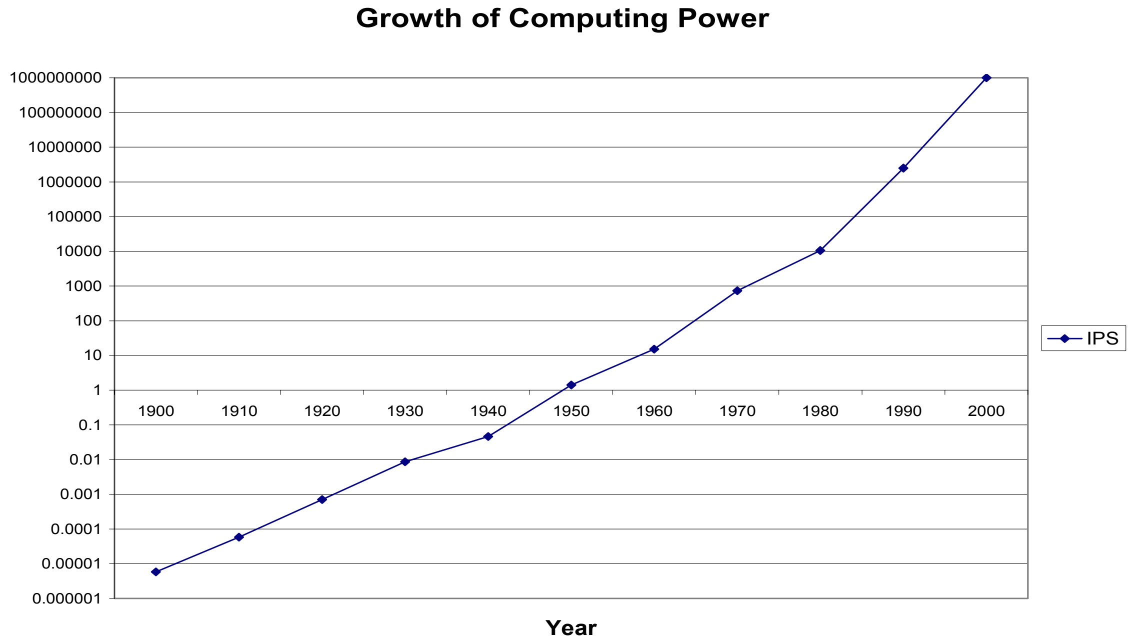


**Figure 5-2: Growth of Computing Power per $1000 over 100 years**[5]

Note that the curve is not a straight line and the doubling period is decreasing over time. The curve has continued past the year 2000, reaching about 6 x $10^9$ instructions per second per thousand dollars this year.

[5] Interpolated from source data in Moravec [15]pp320-321.

## 5.3 Availability of Adequate Computing Power

Computing power per dollar has been nearly doubling every year since 1995. This is faster than historical trends and may not continue unabated, and various doubling periods should be considered in forecasting. Using a baseline of $10^9$ instructions per second per $1000 in the year 2000, we can extrapolate when different levels of processing power will be available for different assumptions of doubling periods:

| | 12 Month Doubling | 15 Month Doubling | 18 Month Doubling |
|---|---|---|---|
| $10^{11}$ instructions/sec | 2007 | 2009 | 2011 |
| $10^{12}$ instructions/sec | 2010 | 2013 | 2015 |
| $10^{13}$ instructions/sec | 2014 | 2017 | 2020 |

**Table 5-1: Moore's Law Predictions of Available Computing Power per $1000**

It would seem that adequate computing power will be available in single processors for only $1000 between 2007 and 2015 if the estimate of $10^{11}$-$10^{12}$ instructions per second is correct.

The military is not constrained to using $1000 computers. Cluster computers with processing power of $10^{11}$ ips could be assembled for less than $20,000 with today's P4 or G5 or Itanium processors and $10^{12}$ ips could be similarly attained in three or four calendar years.

Researchers have in general not been pushing computing power nor computing architectures. The Demo III project uses multiple dual-processor G4 boards but found that inter-processor communication was such a severe problem that the final demos were executed with world modeling and path planning on a single board. Clearly inter-processor communication in cluster computers is as important as individual processor speed.

***The conclusion is that adequate computing power is now or will soon be available with cluster computers to mount a credible attack on autonomous driving. The caveat is that significant engineering effort should be focused on creating appropriate cluster computers that provide adequate processors and adequate inter-processor communication and appropriate development and debugging tools to support researchers.***

Given a development period of three to four calendar years for software to run on new computing architectures, a forecast is made of 2010 or 2011 for reaching a minimal level of human-equivalent performance in autonomous driving.

## 5.4 Confirmation from Other Sources

Other researchers have forecasted 2010 as a reasonable time frame for reaching human levels of performance in autonomous driving.

- Ernst Dickmanns [7], of the Universitat de Bundeswehr in Munich, spoke at NIST in 1999. He estimated that it would take another ten calendar years before adequate computing would be available for truly safe autonomous driving. He felt it would take a factor of 1000 computing power beyond what he was working with at the time to achieve his goals. This would be computing power in the $10^{11}$-$10^{12}$ range.

- The Department of Transportation Intelligent Vehicle Initiative in the early 1990's was focused on autonomous driving. Their programmatic forecast was human level driving by 2010.

# 6.0 Delphi Forecast

As another approach to Technology Forecasting, NIST received approval from Dr. Gage to carry out a Delphi forecast on autonomous driving at the spring MARS PI meeting, held in San Diego April 6-10, 2003.

A Delphi forecast, named for the Oracle at Delphi who was said to be able to forecast the future, is a poll of experts as to when a certain future event might take place. The concept is that a mean prediction of experts is as good an indicator as is possible to achieve.

NIST conducted a Delphi forecast for the Robotic Industries Association in the 1970's, with some success, involving very interesting and useful interaction between university, Government and industry researchers. It was based on this former experience that we proposed to address the current topic of intelligent skill in autonomous on-road driving.

A letter was sent to MARS researchers before the April PI meeting in San Diego, explaining the Delphi procedure and asking attendees to consider two questions:

> "As a MARS PI you are considered to be an expert in autonomous robot software. We ask you to answer the following two questions:
>
> 1. When will human level driving be accomplished in autonomous systems (at a level adequate to get a driver's license)?
>
> 2. What is your assumption of funding (per year or in total) to achieve this result?
>
> Note that the time it takes to achieve a milestone of this magnitude depends upon the funding level. If you wish to give multiple answers (different years with different funding levels) please do so."

## 6.1 Results: Round 1

Several responses were received by email prior to the meeting. These results showed a striking bi-modal distribution, with estimates made by Government and industry researchers being generally much more optimistic than predictions made by university researchers.

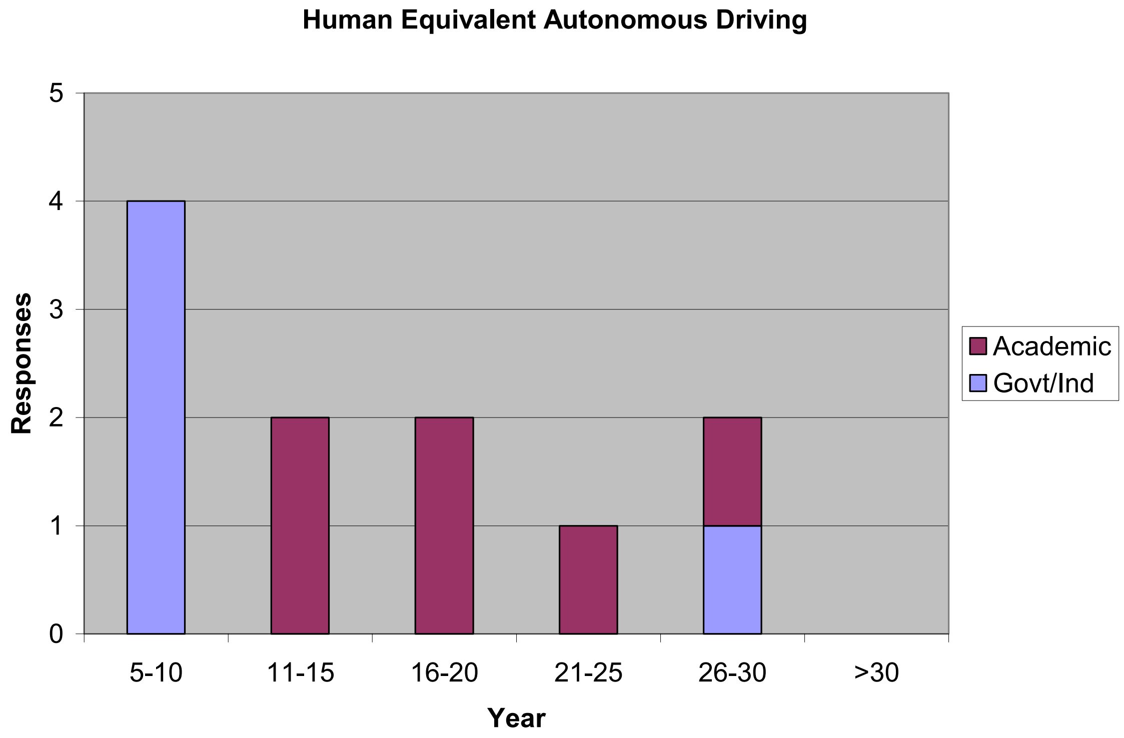


**Figure 6-1: Early Round One Forecasts**

Additional inputs were received at the PI meeting in San Diego. The final first round results are shown in Figure 6-1.

To the extent that participants identified themselves there was still a bi-modal distribution, although not nearly as marked, with several academics predicting 10 calendar years and the outliers past 30 calendar years being responses from industry participants.

Many of the inputs received contained notes and comments justifying the predictions. This is generally what is sought in Round 2 of a Delphi. With an agenda slot to make a presentation to the participants at the meeting, it was decided to try to include the significant comments in the presentation and to only carry out two rounds.

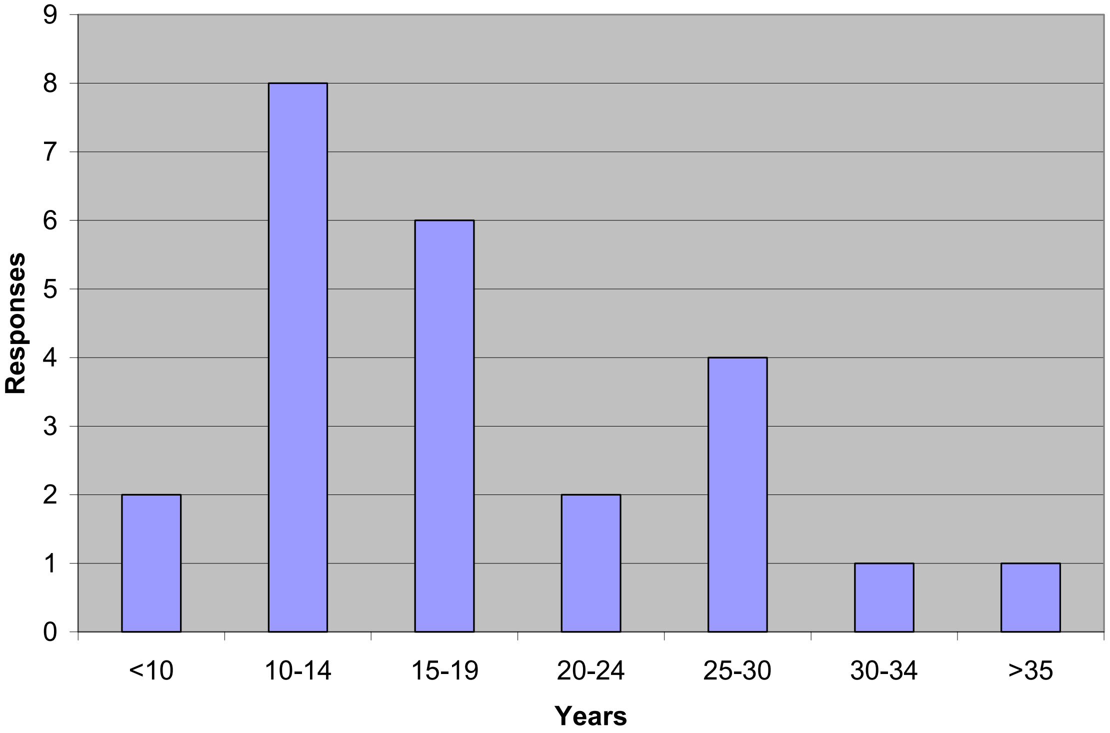


**Figure 6-2: Final Round One Forecasts**

The median prediction is 15 calendar years, with first and third quartiles at 10 and 20 calendar years.  In terms of funding, the median prediction was $350 M, with the first and third quartiles at $100 M and $1000 M.

## 6.2 Clarification and First Round Comments

The lack of any consensus in the predictions, and the apparent difference in outlook between Government/Industry and Academic groups led us to pose the following possible explanations:

- Different definitions of the problem
- Different estimates of the level of funding to be provided
- Different estimates of technological difficulty
- Different estimates of the current state-of-the-art
- Different presumptions of scale of engineering effort

These were addressed in the presentation to the meeting on the second day, before the second round was conducted.

## Definition of the Problem

Dr. Jim Albus, Senior Technical Fellow at NIST, provided the following definition of intelligent driving ability:

- Ability to drive on-road and off-road
- Ability to drive on highways, winding roads, streets, dirt roads, and trails
- Ability to obey rules of the road
- Ability to cope with on-coming traffic, city streets, pedestrians, traffic signs and signals, and intersections
- Ability to read maps and pick routes from point A to point B
- Ability to find a parking space and park
- Ability to drive day and night, rain or shine, snow, sleet, mud
- Ability to safely maintain control at operational speed under all conditions
- Ability to deal with tall grass, weeds, woods, ditches, stumps, and marsh hidden by vegetation

Later discussion with participants brought out the fact that many of these capabilities (particularly the last three) are far beyond what is required to get a driver's license and that we had, in fact, changed the question we were asking. We were now after a higher level of skill than had been considered in the first round, but one that is closer to what we thought Doug Gage was originally after. This increased level of difficulty is reflected in the second round results which push the predictions further into the future.

As another way of defining the problem, the Levels of Autonomy used by Boeing in the solicitation for the Autonomous Navigation System for Future Combat Systems were presented. These are shown in the table below. Doug Gage pointed out that abstract levels are not really a useful taxonomy, that you need to define specific capabilities to do engineering. While this is true, it was felt that the Boeing FCS chart did bring out useful points that would focus the problem definition.

| Level | Description | Perception/ Situation Awareness | Decision Making | Capability | Example |
|---|---|---|---|---|---|
| 1 | Tethered Teleoperation | None | None | Tethered Steer, Speed Brake | Tethered Operator |
| 2 | Remote Teleoperation | Driving Sensors | None | Remote Steer, Speed Brake | Remote Operator |
| 3 | Advanced Teleoperation | Local Vehicle State | Vehicle Health Vehicle | Remote with vehicle state knowledge | Remote Operator with Vehicle State |

| | | | State Info | | Knowledge |
|---|---|---|---|---|---|
| 4 | Supervised, Externally Planned Route | Basic Perception, World Model | Externally generated dense way point path | Operator helps with obstacles | Basic leader-follower |
| 5 | Supervised Internally Planned Route | Sensors for obstacles and hazards | Local planning/ replanning | Operator helps with hazards and obstacles | Convoying, remote path following |
| 6 | Unsupervised Hazard Negotiation/ Avoidance | Local perception correlated with WM | Cost based path planning | Open terrain with operator intervention | Basic open and rolling terrain navigation |
| 7 | Basic Autonomous operations | Path planning using internal WM | Complex obstacles and terrain | Limited speed, operator directed/assisted tactical behaviors | Robust open terrain navigation |
| 8 | Autonomous Fusion of Sensors and Data | Sensor fusion | Robust planning for complex terrain | Complex terrain, limited speed, little operator help, scripted tactical behaviors | Robust complex terrain navigation |
| 9 | Data Fusion of similar data among Cooperative Vehicles | Advanced decisions based on shared data from other vehicles | Complex obstacles and terrain | Complex terrain at full speed; Autonomous initiation of scripted tactical behaviors | Coordinated group achievement of goals |
| 10 | Autonomous Collaborative Operations | Fusion of ANS and RSTA data among all vehicles | Collaborative Reasoning, planning, and execution; Tactical behaviors based on situation | Achieve goals in collaboration with no operator oversight | Final goal of FCS |

**Table 2: Levels of Autonomy for Future Combat Systems**

The FCS solicitation has Levels 1-6 as required deliverables and higher levels as a program goal. It is anticipated that at least Level 7 should be available in a hardened state by the year 2006 (when the technology will be frozen for final design for manufacturing).

The point was made that Demo III vehicles had already achieved Level 7, Basic Autonomous Operations, at Technology Readiness Level 6 (a demonstration of capability in a relevant environment) in experiments this past winter at Toelle Army Depot in Utah and at Ft. Indiantown Gap in Pennsylvania. It was further stated that the Demo III program should achieve at least Level 8 autonomy by the year 2006 if funding is maintained.

The participants were asked again to predict when intelligent skills would be obtained, now thinking specifically about Level 9 capability.

## 6.3 Level of Funding

FCS has budgeted $140 million over the next four calendar years for the development of an Autonomous Navigation System. While only a portion of that will go toward advancing the technology, this is a significant sum. This is in addition to the development of drive-by-wire vehicles, operator interfaces, RSTA and C4ISR.

FCS is only one of many government programs addressing intelligent vehicles. For example, the Unmanned Air Vehicle program is one to two orders of magnitude larger than the Unmanned Ground Vehicle program.

To provide guidance to the participants in the forecast, everyone was asked to assume approximately $500 million over the next ten calendar years. This translates to about 2000 person-years of engineering effort (i.e., 10 teams of 20 professionals working for 10 years) which is a substantial amount of engineering.

## 6.4 Technological Difficulty

The questions that must be answered in order to quantify the effort needed, are:

- What are the perceptual requirements?
- What are the world modeling requirements?
- What are the planning, decision-making, and control requirements?
- What are the system integration and testing requirements?
- What are the requirements for learning?
- What are the software engineering requirements?

The participants were asked to reflect on these questions, and to offer comments and inputs to the report.

## 6.5 State of the Art

**Benchmarks: Current and Past Programs**

Many past and current programs have shown significant success in autonomous driving. Some examples are

- Demo III has demonstrated Level 7 autonomy.
- TARDEC VTI (Vetronics Technology Integration) program (Crew Automation Testbed and Road Follower). Carried out a recent live fire demo at Ft. Bliss.
- Primus C (German version of Demo III) is not far behind Demo III
- Prof. Ernst Dickmanns at Universitat de Bundeswehr in Munich and Daimler-Benz have achieved commercial prototypes of intelligent cruise control which are now in field test; these are based on the German autonomous driving program which achieved hands free driving in highway traffic and 150 km/hr highway speeds.
- CMU NavLab drove across the United States with hands free 97% of the time.
- DARPA MARS researchers have demonstrated substantial autonomous capability
- DARPA PerceptOR is evaluating perception capability for autonomous driving
- The Army Research Lab has funded the Robotics Collaborative Technology Alliance, headed by General Dynamics Robot Systems, as a follow-on R&D effort beyond Demo III.
- The Department of Transportation is funding development and testing of driver assist technologies to improve highway safety. These are generally based on autonomous driving research.

While none of these programs have demonstrated anything close to real human performance in autonomous driving, substantial progress has been made and is being made.

Some details of Demo III and current work, as presented in earlier sections of this report, were provided to the participants. Participants in the Delphi were told to assume, within a decade:

- LADAR with range to 200 meters, depth resolution of 4 cm, foveal resolution near that of human eye, 90 deg peripheral FOV, 3 saccades/sec, 10 frames/sec
- $10^{12}$ ops/sec on-board computing power
- Availability of maps of road networks and terrain features to 3 m resolution
- Access to military or civilian situational awareness reports

## 6.6 Results: Round 2

A second round was conducted after the above discussion, with instructions to:

- Assume FCS Level 9 autonomy (full speed on difficult terrain, city and highway driving)

- Assume hundreds of millions of dollars in funding. Obviously how it is spent will be important
- Assume key enabling technologies under development

The results are shown below. Given that the problem posed was more difficult that in the first round, it is not surprising that estimates are further in the future. Basically all of the short term estimates gone, with nothing remaining less than 10 yrs. All long term estimates remained unchanged, and many were resubmitted with lengthier justifications.

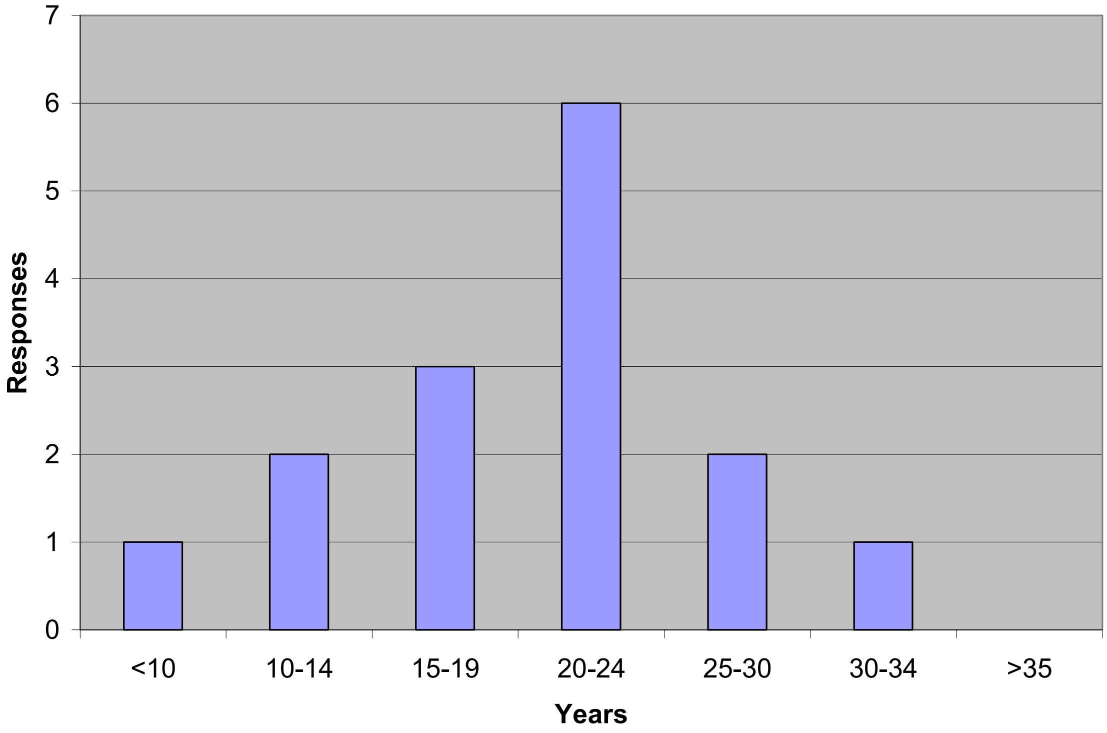


**Figure 6-3: Round Two Forecasts**

The overall result was a compressed range of 5 calendar years for the middle two quartiles (2015 to 2020) instead of 10 (2010 to 2020), and a median forecast of 2020 instead of 2015.

In terms of funding, the range was also compressed and the median further in the future: lengthened: first quartile $360M, Median $500M, third quartile $800M. The span here is a factor of 2.2 (800/360) instead of 10 (1000/100) so there is a tighter agreement among the participants.

The bi-normal distribution between academic and Government/industry participants was much less marked in the second round, although in general Government and industry forecasts, to the extent the participants identified themselves, were more optimistic.

## 6.7 Further Comments

A number of attendees at the San Diego meeting did not participate in the study and several submitted responses that they were unable to make a rational forecast. To draw out their arguments, several researchers were queried in one-on-one discussions and some were asked for written submissions. The comments below by Ron Arkin of Georgia Tech are considered representative:

> The question you ask in your survey is ill-posed.
>
> My basic position can be summed up by recognizing the need for developing the scientific underpinnings of the field before we rush off to establish timetables for implementation. Robotics science is only beginning to be understood and to establish timetables for achieving human level performance seems as foolish to me as establishing timetables to cure cancer, or other basic scientific endeavors.
>
> There are many breakthroughs yet to come in the basic science in understanding human behavior, computational intelligence, and robot-human-environment interaction before such questions can be answered. Funding is required at the basic research level to enable these robotic revelations before robust performance in dynamic and uncertain domains can be guaranteed. Funding enables advances, without it the field will stagnate. But we should not be seduced by the remarkable successes already achieved in such short time frames.
>
> Further, your question seems ill-posed in that it is not even clear that robots should even attempt to achieve human-level performance (why?), or even what or how to characterize human-level performance. Surveys such as this are best left for futurists, not scientists. All they do is to set up expectations which can perhaps devastate the field (e.g., the AI winter) if they are not met. Hopefully our lessons from history will prevent us from making the same mistakes.

Several responses were along the same lines, arguing that robots should not attempt human-level performance and that the basic research issues were so substantial that forecasting engineering success was futile.

An extreme position was taken by one respondent that (1) we would probably achieve near human level performance fairly quickly (5 calendar years) and (2) that we would never achieve fully human level performance.

Again, when queried on human level driving skills, or at least achieving a militarily useful level of driving skill, Alan Schultz of the Naval Research Lab responded:

> Ah, but those are two very different things. When I think of achieving human level performance in driving, I believe the single major problem is perception.

> And I believe that this will continue to be a problem for a long time. However, a militarily useful level of performance is achievable in a much shorter time from.
>
> The difference is in the scope of capability needed. For human level driving performance I include everything from detecting and interpreting signs, road conditions (e.g. spotting ice) etc. The system must be able to handle all contingencies and unexpected conditions and above all, must do so with an extremely high level of reliability and safety.
>
> A military system is more constrained in the environment to be used and most importantly, in most operational situations can operate with a lower level of reliability and safety.
>
> In summary, I project a higher cost and longer time to reaching human-level performance because of the extreme difficulty in obtaining reliable and robust perception.
>
> For militarily systems, I would have picked the middle two quartiles.

Several researchers commented on specific technical issues that needed to be addressed and that were felt to be particularly difficult and that would take substantial time to resolve. John Feddema of Sandia comments:

> I think human-level driving performance could occur much earlier in ideal weather conditions and very structured environments. I do not believe we have a sensor that will reliably work in rain, snow and fog. I also do not think that we even know how to handle the combinatorial explosion of conditions that occur in unstructured environments.

Johann Borenstein of the Universtiy of Michigan wrote:

> My concern is that the performance criteria of "passing a driver's license test" is not sufficient for the safe operation of a vehicle in traffic. Specifically my point is that a human driver's license test assumes correctly that the driver has the inherent ability to do human reasoning and applying human commonsense. I argue that equipped with these skills humans are capable of predicting and dealing with an unlimited number of exceptions. Lacking these skills, a robotic driver will be unable to predict and deal with these exceptions. I agree that many of these exceptions can be anticipated by the robot designers, and appropriate responses can be preprogrammed. However, it is also my contention that there are infinitely many possible exceptions and not all can be pre-programmed.
>
> Based on my slight disillusionment with the capabilities of technology to compete with nature I stated my original opinion that it will take 20 <calendar> years or even more before a robotic driver is feasible. I continue to stand by this opinion

> despite the more optimistic views of many of my peers.

The author, having trained three children to drive and supervised each one of them for some 50 hours of driving experience beyond basic Drivers Ed, was of the opinion that exceptions were indeed critically important, and that getting a driver's license did not make one a competent driver, but that there did not seem to be much reasoning or common sense exhibited in those hours of additional training. Instead, each child had to actually experience examples of problems that are encountered in driving and had to be specifically instructed in how they should be handled.

Clearly some generalization occurs in such training and instruction is at a high level; this brings up the whole issue of learning and human-machine interface. Jean Scholtz of NIST comments:

> Currently human interaction with robotic driving platforms consists of two modes: autonomous or tele-operation. There are instances where a few commands such as back-up and try again are available to the operator. Tele-operation may suffice as a fallback mode of operation for off-road driving or other types of robotic tasks, such as search and rescue, but tele-operation is of limited use for on-road driving in urban terrain. The urban situation has numerous vehicles, pedestrians, traffic controls, and road obstructions such as detours, potholes, and parked cars that an on-road driving vehicle must sense and react to quickly. It is unlikely that a remote operator can react quickly enough to safely navigate through urban situations.
>
> There are a number of roles that HRI needs to support. For on-road driving, a supervisor might oversee a number of different vehicles operating in the same general geographic area. An operator might be called on to provide support for a vehicle that is having problems navigating in a particular situation. A team mate might be the driver of a manned vehicle that is operating in conjunction with an unmanned vehicle to accomplish a particular task. A mechanic might be needed to fix sensory equipment or other mechanical problems and would need to issue some commands to the robot to ensure that the problem had been fixed. In the on-road driving domain there are likely to be a number of bystanders; that is, people who have not been exposed to any robot training but who will be driving or walking in the same environment that the robot is navigating. In addition, there are information consumers. These are the people who are interested in the information provided by the robot. That information might be surveillance information or medical information provided by search and rescue robots. The consumers of information might be allowed to interaction with various sensors (such as cameras) on the robot or they might have to make requests through the supervisor or operator to obtain information.
>
> Another issue is that of HRI awareness. In situations where there are multiple people and multiple robotic platforms, teams will function effectively only if user interfaces provide for awareness between the various team members. Humans

> must be aware of what robots are doing but in addition robots need to be aware of what other robots are doing and what the humans are doing. As with any team, humans need to be aware of what each other is doing. In particular when a number of humans are interacting in different roles with the same platform, the user interface needs to provide this awareness.
>
> Basic research issues for HRI include:
>
> - Determination of the information and level of abstraction necessary to provide the situation awareness for each interaction role.
> - Interactions to support adjustable autonomy
> - Platform independent interaction vocabulary
> - Fusion techniques for providing sensory information to maximize situational awareness and minimize user's cognitive load
> - Robot awareness of user's cognitive and physical workload
> - Smooth handoff or switching strategies between roles and platforms
> - Interactions with teams of robots
> - Interaction architectures integrated with real-time robotics architectures
> - Metrics and methodologies for evaluation of HRI
>
> Underlying all these issues is the premise that the current robot platforms and the current interaction modalities and platforms will evolve. HRI needs to be designed for the robots and interaction modalities of the future. A research program devoted to HRI issues is needed to make significant progress in these areas. A five <calendar> year, $50 Million interdisciplinary program (cognitive psychologists, HCI researchers, and robotics researchers) would produce good results for HRI as there currently exists additional funding in modality research as well as in augmented cognition. The results from these efforts could be integrated into a more specialized program in HRI.

Jim Keller of UPenn believes the most important issue is knowledge management:

> I think the biggest challenge is management of the knowledge base that constitutes a good driver and not necessarily the navigation, perception, command and control aspects that typicall come to mind in a robotic application.
>
> Perhaps it would be better to qualify the level of expertise to the following levels:
>
> - Just received drivers' license (requires <10 years to get there): this is the robotics part of the problem.
> - Approximate expertise after human has been driving one year, five years, etc.
>     - As the level of expertise is increased, the knowledge base management is more the issue. In this regard, until the robot becomes conscious, I do not think it will ever exceed human performance. The solution is more complex than other knowledge

base issues like computer chess because of the real time representation.

- Another way of making the problem tractable would be to limit the speeds expected (i.e. type of road).

Similarly, John Weng of Michigan State University comments:

> Human level performance requires a highly integrated driving system. Human designed domain knowledge tends to leave many holes, which are in fact infinite or unbounded in possibility. Real world “living” experience and learning while “on the fly” is a powerful way of filling these holes with skills of “interpolation” between known cases and new exceptions.

Finally, there were those that thought that producing useful military technology should be addressed as an engineering problem rather than one of basic research trying to achieve an abstract (and unjustified) goal of matching human performance. Alan Schultz of NRL makes that point above. Sebastian Thrun of CMU notes:

> To me, it is *not* a question of human level computation to achieve human level driving.
>
> If we want vehicles to drive people autonomously, I believe the technology mostly exists, but it would require instrumenting our roads. We already have instrumented our environment to facilitate human driving. The steps necessary to facilitate autonomous driving would be minor in comparison. I believe the most important hurdle towards autonomous driving is not technical, but societal (and to a minor extent: legal).
>
> Autonomous driving on roads designed only for human driving is a different story, one with great importance for the military. Again, I believe we don't need human level cognition, perception, or reasoning. But we do need significant advances towards reliable perception. I personally believe some of these advances will be tied to computational power, but the computational metaphors will be quite different, in that probabilistic computation will play a pivotal role in the design of autonomous driving systems.

## 6.8 Results

While the results are not considered definitive, particularly because of the change in the problem definition between rounds one and two, it is clear that ***researchers generally felt that it would take at least ten calendar years and probably closer to twenty calendar years to achieve the capabilities of autonomous driving desired for Future Combat Systems, and that funding of the order of $500M would be needed.***

***It was further clear that setting general human levels of autonomy is not the correct approach, that specific military needs and modes of driving need to be addressed and solved, and that this involves continued research in sensors, perception, knowledge management and planning.***

# 7.0 Conclusions

Useful and practical autonomous driving is in its infancy. As such, there will certainly be unforeseen challenges and periods of both pessimism and over-optimism. Nonetheless, a review of the accomplishments to date, and a survey of current views of experts in the research community is useful, and has provided a basis for a best-estimate at this time of the nature and size of the challenge. While not unanimous, the most prevalent views lead to these overall conclusions:

- ***Militarily useful autonomous driving capabilities can be developed in approximately ten to twenty calendar years on continued research. The time scale will depend upon the level of funding available.***
- ***The cost will be in the range of three to five hundred million dollars, which is consistent with current funding levels of Army autonomous mobility programs extended over twenty calendar years.***
- ***The biggest single problem is perception. The attack on the problem should start with development of a new generation of sensors designed specifically for autonomous driving.***

The conclusions of the different approaches to estimating time and cost for achieving intelligent on-road driving, which support the overall conclusions above, are summarized below.

First: Based on extrapolation from the Demo III experience, it will take another fifteen calendar years of work at the current level of effort to achieve intelligent on-road driving capability.

Second: Based on the Task Decomposition of driving tasks using the DoT manual, it is estimated that approximately $300-400 Million in funding will be needed to achieve intelligent on-road driving skills. Over a twenty calendar year period, this is $15-20 M per year, roughly the level of funding now provided under the ARL and TACOM programs. Increased funding would reduce the time scale.

Third: A new generation of sensors designed specifically for autonomous driving is needed to provide the necessary visual acuity. This is critical because perception emerges as the largest problem in autonomous driving.

Fourth: Engineering attention needs to be paid to providing adequate processors with adequate inter-processor communication to researchers along with software development and debugging tools. Adequate computing power using cluster computers is now or will soon be available, making it possible to address these engineering issues in the near future. Computing power should not be a gating element.

Fifth: Based on the Delphi Forecast of MARS researchers, it will take 15-20 calendar years and of the order of $500M to achieve intelligent driving skills.

Sixth: Several MARS researchers emphasized that setting intelligent driving skills as the goal was not the correct approach, that militarily useful capabilities would be achieved short of that goal

Seventh: Continued research in sensors, perception, knowledge management and planning, at a level at least equal to current funding is essential, even if the scope is reduced to targeting specific military driving modes to be solved in the near term.

## References

1. Albus, J., Huang, H., Messina, E., Murphy, K., Juberts, M., Lacaze, A., Balakirsky, S., Shneier, M., Hong, T., Scott, H., Proctor, F., Shackleford, W. P., Michaloski, J. L., Wavering, A., Kramer, T., Dagalakis, N., Rippey, W., Stouffer, K., Legowik, S., Bostleman, R., Norcross, R., Jacoff, A., Szabo, S., Falco, J., Bunch, B., Gilsinn, J., Chang, T., Meystel, A., Barbera, A., Fitzgerald, M., DelGiorno, M., and Finkelstein, R., "4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems," NISTIR 6910, Gaithersburg, MD, 2002.

2. Balakirsky, S. and Lacaze, A., "Value-Driven Behavior Generation for an Autonomous Mobile Ground Robot," *Proceedings of SPIE's 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, 2002.

3. Balakirsky, S. B., *A Framework for Planning with Incrementally Created Graphs in Attributed Problem Spaces*, Akademische Verlagsgesellschaft Aka GmbH, Berlin 2003.

4. Barbera, A., Horst, J., Schlenoff, C., Wallace, E., and Aha, D., "Developing World Model Data Specifications as Metrics for Sensory Processing for On-Road Driving Tasks," *Proceedings of the 2003 Performance Metrics for Autonomous Systems (PerMIS) workshop*, 2003.

5. Camden, R., Bodt, B., Schipani, S., Bornstein, J., Phelps, R., Runyon, T., French, F., and Shoemaker, C., "Autonomous Mobility Technology Assessment Interim Report," *Army Research Laboratory (ARL-MR-565)*, 2003.

6. Churchland, P. S. and Sejnowski, T. J., *The Computational Brain*, MIT Press, Cambridge, 1999.

7. Dickmanns, E., "multiple papers," *Proc. IEEEE Intelligent Vehicles Symposium*, Dearborn, MI, 2000.

8. Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol. 1, 1959, pp. 269-271.

9. Ferlis, R. A., "Improving Urban Mobility Through ITS," *To be published in City Matters (http://www.urbancity.com),* 2003.

10. Grossberg, S., *Boston Universoty, private communication*, 2003.

11. Hong, T., Balakirsky, S., Messina, E., Chang, T., and Shneier, M., "A Hierarchical World Model for an Autonomous Scout Vehicle," *Proceedings of SPIE's 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, 2002, pp. 343-354.

12. Kurzweil, R., *The Age of Spiritual Machines*, Penguin Books, New York, 1999.

13. Lacaze, A., Murphy, K., and DelGiorno, M., "Autonomous Mobility for the DEMO III Experimental Unmanned Vehicle," *Proceedings of the AUVSI*, Orlando, FL, 2002.

14. McKnight, J. and Adams, B., *Driver Education Task Analysis. Volume 1. Task Descriptions*, Human Resource Research Organization, Department of Transportation, National Highway Safety Bureau 1970.

15. Moravec, H., *Robot, Mere Machine to Transcendent Mind*, Oxford University Press, New York, 1999.

16. Shoemaker, C. and Bornstein, J. A., "Overview of the Demo III UGV Program," *Proceedings of the SPIE Robotic and Semi-Robotic Ground Vehicle Technology Conference*, Vol. 3366, 1998, pp. 202-211.

17. U.S.Department of Transportation, F. H. A., *Manual on Uniform Trarffic Control Devices (MUTCD 2000) Millenium Edition* 2000.