THE UNIVERSITY OF MICHIGAN - DEARBORN

WOLF

Intelligent Ground Vehicle Competition 2009

TEAM MEMBERS

Alan Akroush, Gregory Czerniak, Ed Klacza, Gierad P. Laput, Anthony Lucente, Ross Marten, Jason Smith

Chapter 1

Design Report

1.1 Introduction

The University of Michigan - Dearborn presents *Wolf*, a contender for the 17th Annual Intelligent Ground Vehicle Competition.

For 2009, Wolf received changes to its sensors, which essentially transformed it into a less expensive version of Raptor. Although its sensors had lower resolution, lower refresh rates, and higher error ranges, the team believed that Wolf could run at performance levels comparable to its big brother. To do this, the team employed the Next-Generation Robot Platform software framework, which is identical to the software on Raptor. To account for the lower-quality sensors, filtering algorithms and more advanced vision processes were employed.

1.2 Design Innovations

SICK Laser Range Finder

Wolf's obstacle avoidance system has been upgraded to use the SICK laser range finder. The previous version of Wolf used ultrasonic and infrared sensors to detect obstacles. Although these sensors were cost-effective, there was an obvious trade-off in accuracy. With the implementation of the SICK laser range finder, Wolf is able to perceive obstacles with a 180-degree field of view at a resolution of one degree. This significant improvement allowed Wolf to detect obstacles more effectively.

NGRP Software

This year's version of Wolf sports a fully revamped sofware architecture. The Next-Generation Robot Platform (NGRP) was implemented in this year's version of Wolf. Apart from the previous version, NGRP allows Wolf to easily incorporate software modules and add new features. With NGRP, hardware or software upgrades will only require writing separate modules which can be tested and simulated independently.

The main motivation for implementing NGRP in Wolf was to test the robustness and flexibility of the platform. Raptor, another unmanned system from the University of Michigan - Dearborn, uses NGRP. The goal of NGRP was to provide a universal platform for intelligent ground vehicles, and a successful implementation of NGRP in both Raptor and Wolf will be a major achievement.

1.3 Team Organization

The development team for Wolf comprised of both undergraduate and graduate students from the Intelligent Systems Club. Weekly meetings provide an avenue for members to present their newest work or findings. The team operated under a democratic structure wherein major decisions are subject to the approval of the majority. Teamwork was further enhanced by using effective collaboration tools such as version control systems and a centralized knowledgebase on UMich CTools. Better design ideas and the prevention of early mistakes in the design process were the primary benefits that emerged from the extra collaboration.

- Lead Developer: Gierad P. Laput, ECE Graduate
- Software Platform Developer: Greg Czerniak, ECE Graduate
- Sensor System Developer: Anthony Lucente, ECE Graduate
- Simulation Developer: Ed Klacza, ECE Graduate
- Vision System Developer: Jason Smith, ECE Master's Student
- Contributing Developer: Alan Akroush, ECE Graduate
- Contributing Developer: Ross Marten, ECE Senior

1.4 Development Process and Systems Integration Plan

Development Process

A continuous integration process was enforced by the development team to ensure that each newly added feature was working as expected. Whenever a new function was added, it was immediately integrated into the larger system and then tested exhaustively. This development methodology made the debugging process easier because errors were easily isolated.

Systems Integration Plan

Implementing the hardware changes on Wolf was a straightforward task since it only involved replacing the old sonar sensors with the SICK laser range finder. The ultrasonic and infrared sensors were removed from the chassis and the wire connections between the sensors and the host computer were disconnected. Consequently, the SICK laser range finder was then mounted in front of the chassis. Finally, the cables and software drivers were connected and installed to the host computer.

On the other hand, implementing the new software platform on Wolf demanded a few critical adjustments. An overview of the software transition process is listed as follows:

- First, preqrequisite software required to run NGRP such as Python and Visual Studio was installed on the host computer.
- Second, the complete NGRP platform was installed into the Wolf's host computer.
- After installation, NGRP parameters were configured to precisely calibrate the sensors on Wolf.
- Lastly, a quick sanity check was performed to ensure that all critical components worked correctly.

1.5 Hardware

1.5.1 Mechanical

The Wolf robot has undergone both mechanical and software improvements for the 2009 IGVC competition. The more experienced members who have been to at least one competition made these modifications.

Wolf's mechanical design consists of a rigid chassis, two independent drive wheels and front mounted casters. The robot uses differential steering. The electrical system is powered by two twelve-volt marine deep cycle batteries connected in series providing 24 volts. A 24 to 12 volt converter supplies power to the subsystems.

As in previous competitions, the electronics are accessible opening the robot like a clam shell using hinges and a trunk strut to hold it open.

The laptop is located on the back of the robot such that it is easy to see the screen for testing purposes. The laptop lid can also be closed and tucked into the robot for competition.

1.5.2 Electronics and Electrical System

Motors

The motors are a pair of widely-used wheelchair motors chosen for their high torque and carrying capacity. These motors accept a 12.5 inch wheel directly on the output shaft simplifying the mechanical design. The rear of the motors have an electric brake which is removed to allow the quad optical encoders to be attached.

Motor Controller

A Roboteq motor controller was used to provide control to the motors. This controller was chosen because of its rich feature set that would have been time and cost prohibitive to design and build in-house. These features include a simple serial interface to accept commands, encoder inputs, high amperage ratings and over voltage protection. The controller can be commanded to move the robot at a specified speed and radius, and the controller will rely on the encoders to ensure that speed is met. This was essential because the power required to move the robot changed depending on the incline and terrain.

Sensors

A SICK laser range finder has been added for this year's competition and replaces the ultrasonic sensors used in previous competitions. The laser provides much more information more accurately than the set of ultrasonic sensors essential for obstacle detection. The five ultrasonic sensors provided just five zones of measurement providing straight-ahead, left and right obstacle detection. The SICK laser makes a measurement every degree for a total of 180 measurements. This resolution allows for identifying the exact placement of obstacles.

The camera mount has also been redesigned and replaced since it became weak and unstable from use. The mount is now taller and moved forward to prevent the camera from seeing the front of the robot. Large diameter steel pipe was used in order to comfortably run the wires down to the main body. This head also holds the safety light, wireless emergency stop and GPS unit.

- Vision
  - Unibrain Fire-i Digital Camera
  - Used to detect lanes and other obstacles
- Odometery
  - Quad Optical Encoders
- Positioning
  - Garmin 16A 5Hz GPS
  - Used in the Navigation Challenge
- Obstacle Avoidance
  - SICK Laser Range Finder
  - Provides 180 data points at 180-degree field of view (1 degree resolution)
  - At each specific angle, the SICK returns the distance of the closest object
  - Used in autonomous mode to detect and avoid obstacles

1.6 Software

1.6.1 New Platform

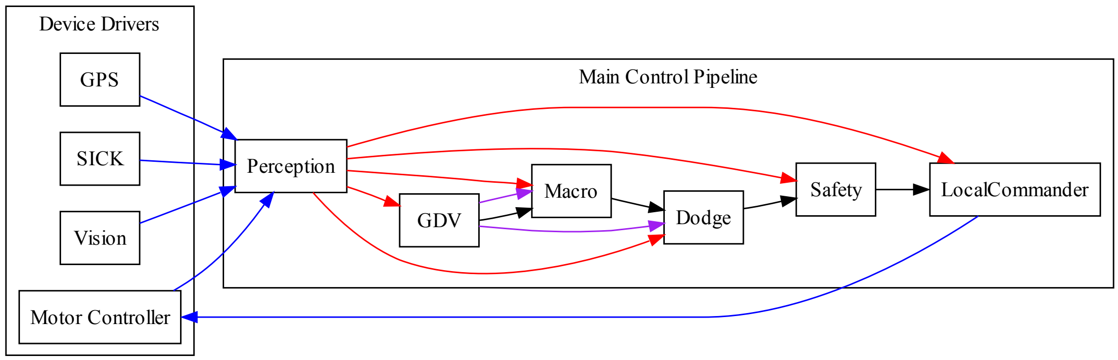

Figure 1.1: A data flow graph of NGRP. Blue arrows are UDP, red arrows are map data, black arrows are differential drive commands, and purple arrows are artificial obstacles.

This year, in an attempt to test the robustness of the Intelligent Systems Club's Next-Generation Robot Platform (NGRP) software framework, the team decided to convert Wolf into an NGRP-compatible unmanned vehicle. The new platform offered several benefits:

- Expandability: NGRP was built to be modular and flexible, allowing new technologies and algorithms to be added with relatively little systems integration effort.
- Configurability: Through its powerful configuration file system, modules in NGRP can be configured for different robots without modifying or recompiling code.
- Simplicity: Modules in NGRP perform one action and do it well, making individual modules readable and understandable.

1.6.2 Strategy

Pipelined Control System

By switching the software to the Next-Generation Robot Platform (NGRP), Wolf acquired the pipeline-based control system built into NGRP's codebase. This pipeline starts with a General Direction Vector (GDV) and performs a series of increasingly-reactive behaviors to the trajectory of the robot. The effective range of these behaviors are inversely related to each algorithm's reactiveness.

The first algorithm in the pipeline is Macro, which performs a sweep of 180 rays from right to left in front of the robot. These rays stop when they intersect an obstacle such as a line or a barrel. Once this sweep is performed, the algorithm attempts to find the dot product of the GDV and each line segment, and chooses the direction of the line with the highest scalar value as the best direction. This is under the assumption that the best direction for the robot to go is the one that leads to the farthest distance toward the General Direction Vector.

Macro feeds directly into the next module called Dodge, which is a very reactive algorithm. Simplified, the algorithm reduces to "if there is more stuff on the left, turn right, and if there's more stuff on the right, turn left." Since following this algorithm is clearly not in the best long-term interest of the robot, Dodge's influence is restricted to obstacles that are less than or equal to one meter in front of the robot. That way, Dodge can prevent Wolf from crashing into obstacles, but it cannot dominate the control pipeline.

Finally, once Dodge has performed its influence on the control system, the differential drive signal is sent to the Safety module, where it is subject to multiple sanity checks. These checks include:

- Is the robot about to run directly forward into an obstacle?
- Is the robot about to make a turn into an obstacle?
- Is the robot trying to make a turn too sharp for its reaction time?

If the Safety module determines any of these conditions have occurred, the module will prevent the robot from making a false move.

Navigation Challenge

For the Navigation Challenge, the software uses the same basic control pipeline, but rather than setting the GDV through an algorithm, it sets the GDV to be the as-the-crow-flies line between the robot's current position and the next waypoint. That way, the rest of the control pipeline will naturally cause the robot to pursue that direction and reach the waypoint. Once the robot has detected that it reached a waypoint, it simply modifies the GDV to point to the next waypoint on the list provided by the competition officials.

GPS Filtering

Instead of the differential GPS present on Raptor, the Wolf uses an inexpensive conventional GPS manufactured by Garmin. With WAAS, this GPS is accurate to approximately one meter, and without WAAS the GPS is accurate only to ten meters. This amount of error can be fatal in the Navigation Challenge.

To mitigate this level of error, the team used a basic Kalman filter to build what amounted to a self-adjusting low-pass filter. By doing this, the system would filter the incoming GPS coordinates. More precise relative movement information was supplied by the Roboteq's odometry system, while the Kalman-filtered GPS coordinates would supply absolute coordinates. By combining the strong points from both of these sensors, it is possible to participate in the Navigation Challenge using inexpensive equipment.

1.6.3 Image Processing

Lane Detection

Most of the image processing for lane detection is performed on the GPU. The code uses the following sequence of operations to extract white lines from the image:

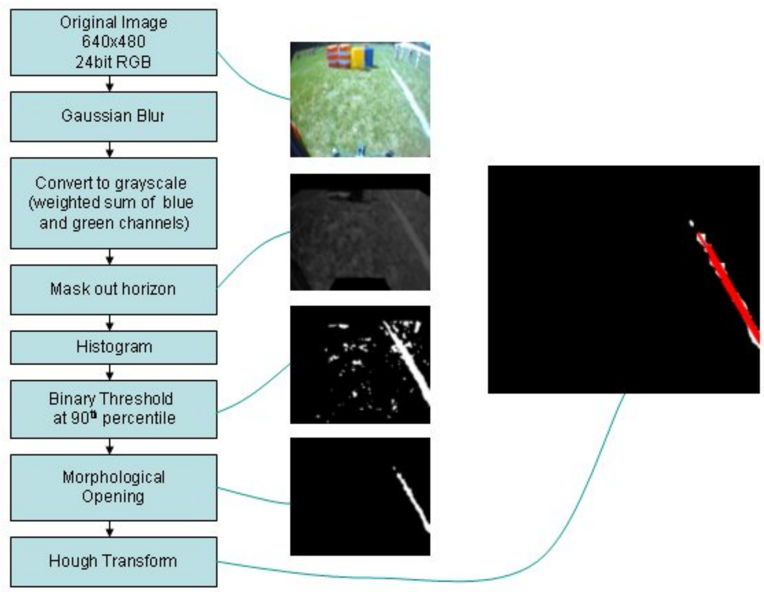

Figure 1.2: An illustration of Wolf's vision pipeline.

Each operation is described in detail below:

Gaussian Blur

After experimenting with a number of filters, the team found that a 3x3 Gaussian kernel with $\sigma = 0.95$ achieved the most acceptable reduction in noise without sacrificing too much detail from the image.

Grayscale Conversion

To convert a color image to grayscale, the red, green, and blue color channels must be combined to form a grayscale channel. The most common approach to grayscale conversion is to use luminance. This technique was designed to make black and white images more attractive because of characteristics in the human vision system, and is not necessarily appropriate for this specific application.

The team found that by subtracting 1/3 of the green channel from the blue channel, white lines could be isolated and brightness from dead and brown grass could be reduced.

Binary Threshold

After the image has been converted to grayscale and the horizon and other non-useful features have been masked out, a binary threshold is performed. Since the white lines on the grass are often the brightest features in the image, the threshold is determined by calculating a histogram and finding the 90th percentile. This permits only the brightest 10% of pixels to pass through the filter.

Binary Morphology

Sometimes small, disconnected blobs of pixels appear in front of the robot because of dead grass. Morphological erosion can be applied to reduce the effects of this problem. The team found that using multiple iterations with a 3x3 square structuring element to be the most effective.

Hough Transform

The Hough transform is used to extract straight lines from the processed image. The code uses the OpenCV probabilistic Hough transform.

Barrel Detection

An experimental component of Wolf's vision system uses object detection to localize barrels. Part of the intent of this system is to pave the way for future systems that rely more heavily on machine vision than active proximity sensors. Barrel locations can also be used as markers for constructing an internal map of the course.

After surveying several image-based methods of object detection, the Viola-Jones algorithm was selected for its speed, robustness, and free availability as part of the Intel OpenCV library. The Viola-Jones method is an ensemble-based learning system. This algorithm is distinguished from other object-detection methods by four key components:

- Feature detection using rectangle features that are reminiscent of Haar-basis functions.
- The use of an integral image, which heavily speeds the calculation of average pixel intensity over rectangular regions in the image.
- The use of AdaBoost so that several weak classifiers are combined to form an overall strong classifier.
- Arrangement of classifiers in a cascade, where the highest performing classifiers appear at the head of the chain so that negative images can be quickly rejected.

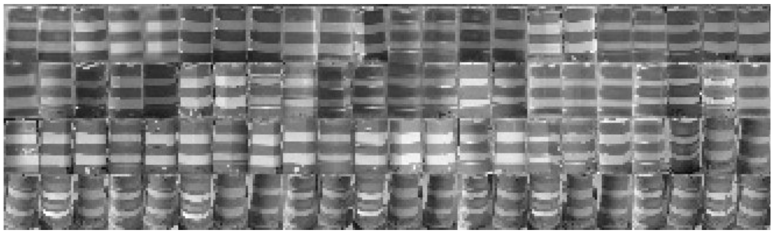

Figure 1.3: A small subset of barrel images used for training.

The detector was trained using the provided HaarTraining utility with 992 positive examples and 4,165 negative examples. A small portion of the positive examples are shown below:

A base resolution of 15x24 pixels was used for the detector. The detector was trained for 10 days using a 2.6GHz quad-core CPU which led to a total 23 classification stages. On a test set of 733 images, the detector was able to achieve a positive detection rate of 0.726 with a false positive rate of 8.19×10^{-3}. The test set was intentionally biased with hard-to-classify images (with barrels rotated outside the limit of what is typically encountered). In practice, the barrel detector is functional with very few occurrences of false positives, but will need to be re-trained with new images taken from the 2009 IGVC competition.

Figure 1.4: A screen shot of a detected barrel.

1.7 Safety

Safety was a major concern when developing both the hardware and the software for Wolf. Wolf featured a mechanical emergency stop button on the rear that cut all electrical power when pressed. A wireless emergency stop effective to 50 feet was also on Wolf, which prevented catastrophes associated with "runaway robot" scenarios.

The NGRP software required a dead-man button to be pressed for all movement commands, which prevented accidents from unintended movement commands. Also, as mentioned in previous sections, the software has automatic safety checks made before sending every motor controller command to prevent accidental collisions.

1.8 Performance and Cost Analysis

Tables 1.1 and 1.2 at the end of the paper contain the information for cost and performance characteristics, respectively.

1.9 Conclusion

Wolf took approximately 300 man-hours to prepare for the 2009 IGVC. Although it is about half as expensive as Raptor, the team believes that with the NGRP software and the specialized vision software, it can perform at a level comparable to Raptor. The team is very eager to see Wolf's performance in this next competition.

1.10 Acknowledgements

The team would like to thank Larry Sieh for his help and support.

1.11 Signature

I certify that the design and creation of Raptor has been significant and is equivalent to what might be awarded credit in a senior design course.

Dr. Narasimhurthi Natarajan

Professor of the Department of Electrical and Computer Engineering, University of Michigan - Dearborn

Date

Description	**Cost**	**Quantity**	**Total**
Chassis	$250.00	1	$250.00
Wheels	$40.00	2	$80.00
Casters	$20.00	2	$40.00
Laptop Computer	$1,000.00	1	$1,000.00
Embedded ARM Microcontrollers	$555.00	1	$1,110.00
SICK Laser Range Finder	$5273.00	1	$5273.00
Motor Controller	$700.00	1	$700.00
FireWire Camera	$173.45	1	$173.45
Mounting Supplies	$49.47	1	$49.47
Batteries	$159.95	2	$319.90
Painting / Supplies	$28.90	1	$28.90
		Total	$9,024.72

Table 1.1: The table of expenses for Raptor.

Attribute	**Design Prediction**
Maximum Speed	5.0 mph
Climbing Ability	30 Degree Ramp
Nominal Power Consumption (Watts = Amps x Volts)	240 Watts
Battery Operating Time (24v 55AH Battery System)	6 hours
Distances at which objects can be detected	5.5 meters
Waypoint Accuracy (DGPS)	<3 meters WAAS
Reaction Times	100ms
How Vehicle Deals with Complex Obstacles	Reactive Fuzzy Logic

Table 1.2: The table of expenses for Raptor.