Omnix2006

HOSEI UNIVERSITY

Systems and Control Engineering Department
Faculty of Engineering, Hosei University
3-7-2 Kajinocho Koganei, Tokyo 194-8584, Japan
E-mail; ikko@k.hosei.ac.jp

Faculty Advisor Statement

I hereby certify that the engineering design on Omnix2006 was done by the current student team and has been significant and equivalent to what might be awarded credit in a senior design course.

Signed 小林一行

Associate Prof. Kazuyuki Kobayashi

Date May/18/2006

May 18, 2006.

1. Introduction

The Autonomous Robotics Lab (ARL) team of Hosei University newly presents Omnix2006, an innovative vehicle designed for entry for the 2006 Intelligent Ground Vehicle Competition (IGVC). The Omnix2006 is an autonomous intelligent wheelchair of the next generation. A new chassis with a novel omni-wheel mechanism is employed. The name Omnix comes from "Omni-wheels," "Omni-directional camera" and "Omni-directional vision interface." Figure 1 shows the innovative features of the Omnix2006.

Keywords:

- All-wheel Drive (AWD)
- Omni-wheels for zero-radius turn
- Head Mounted Display (HMD) as an intuitive navigation interface for wheelchairs
- Intuitive wheelchair steering interface using a 2-axis accelerometer sensor

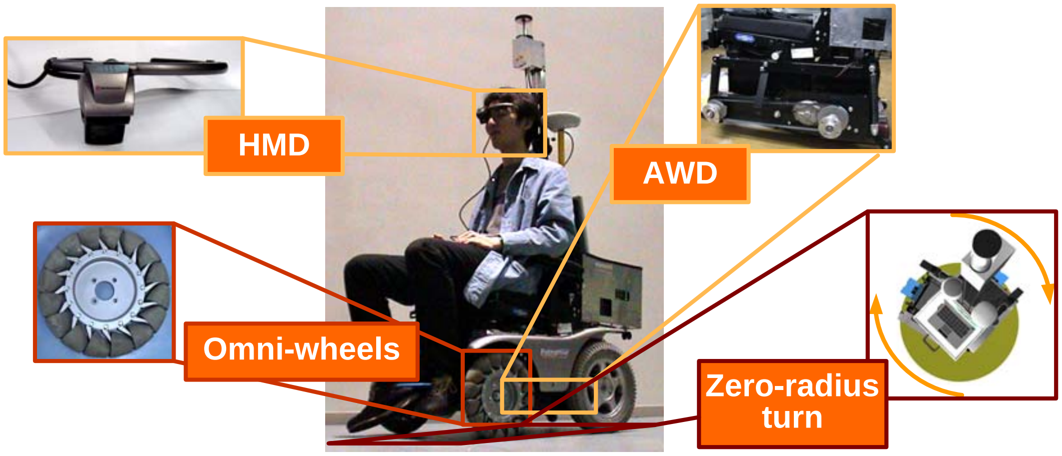


Figure 1: The next-generation intelligent wheelchair "Omnix2006"

We developed an intelligent and safe electric wheelchair by applying the newest technology to the newest chassis. The autonomous wheelchair developed promises to spearhead the next generation of personal transportation vehicles.

2. Design Innovations

The key issues in developing a vision-based, lane-following Autonomous Ground Vehicle (AGV) are safety, accuracy and robustness in navigation. An AGV that navigates undefined and/or unstructured and dynamically changing environments must react quickly and correctly to all situations to avoid danger.

Figure 2: Feedback architecture of Sense-Plan-Act loop

Figure 2 shows the feedback architecture called Sense-Plan-Act (SPA) employed in the AGV development. Signals detected by sensors in the Sense block are processed, feature-detected, recognized and modeled via the reasoning and decision-making process in the Plan block; then, a reasonable control signal is generated to actuate motors in the Act block, and actuation results are fed back to the Sense block. The advantage of feedback architecture of SPA is that any design task can be decomposed into subtasks of the three main functions (blocks) shown in Figure 2; hence, in the development and design process, different team members could specialize in particular decomposed tasks. However, the system itself with the architecture in Figure 2 had a delay time in sensing from slow-sensing devices such as a CCD camera and in reasoning and decision-making from a computer with limited computational power. This delay time would lead to terrible problems in feedback control in real-time situations. To solve the problems associated with delay time, we developed a navigation strategy combining open-loop and closed-loop systems, which we call “vision-based open-loop control navigation”. This new system including an open-loop was developed under the SPA architecture in Figure 2.

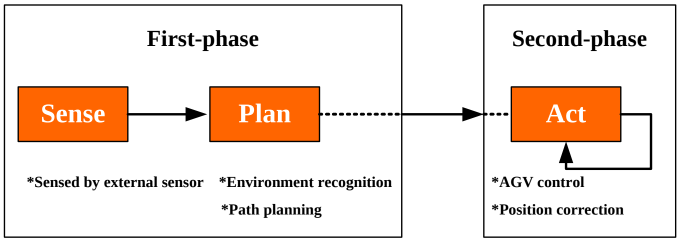


Figure 3: Vision-based open loop control navigation

The “vision-based open-loop control navigation” strategy shown in Figure 3 consists of two phases. The first phase consists of an off-line sensing process by external sensors, an environment-recognition process and a path-planning and path-calculation process. The second phase consists of on-line motor control and position-correction processes by internal sensors. In the first phase, the AGV acquires a time-series of images and completes the image-processing iterations until it has sufficient confidence of success in generating the appropriate path. During image acquisition and calculation, the vehicle does not move. Subsequently, the AGV navigates based on the appropriate path provided by the processing in the first phase. The key strategy in controlling the new vehicle is the separation of off-line route scheduling and on-line navigation control given by the above two phases. This separation yields efficient use of sensor signals, some of which are measured slowly, and quick vehicle control, which leads to safe navigation in a variety of outdoor environments.

3. Design process

Deadlines and roles were allocated appropriately to each member under the accurate recognition of the kinds and levels of their abilities in design processes. After brainstorming by ARL members under advice from faculty advisers, we employed the Rational Unified Process (RUP) based on Unified Modeling Language (UML) as our design process. RUP was also employed in the design of Amigo2005. Use of the RUP approach allows team members to work individually with their own skills, then share their knowledge under a unified total plan. The documents and design information emerging from RUP and written in UML facilitate smooth communication between the team members.

3.1 Unified Modeling Language (UML)

UML standardizes signs and meanings and thus can describe statements clearly. Diagrams (schemes) written using UML provide efficient information for each team member to carry out co-operative work. Schematic representation using the allowed diagrams provides for an intuitive and/or easy understanding of complicated concepts and processes. Moreover, UML provides information about how the project is proceeding as a whole.

3.2 Improved Rational Unified Process (RUP) for the IGVC Project

RUP is one of the well-known methods for developing design processes in UML. Originally, RUP was a method for organizing and managing software developments in software systems that integrate various user requirements. The IGVC project includes not only software development but also hardware and system integration processes, so conventional RUP could not be applied directly. In order to realize the philosophy of RUP, we introduced a new design process called Improved RUP, IRUP. The IRUP specializes and divides a large-scale system or problem into different application areas, different types of organizations, and different competence levels. Figure 4 shows the proposed IRUP for the IGVC project. The IRUP was based on six groups: Mechanical System Group, Electrical System Group, Joint Architecture for Unmanned Systems (JAUS) Group, Autonomous Challenge Group, Navigation Challenge Group and Design Competition Group. Each of these design processes was composed of one or more iterations. Each iteration follows a waterfall pattern containing requirements for gathering, analysis, design, implementation, testing, evaluation, deployment, and a final product, which grows incrementally from iteration to iteration. To iterate the design process appropriately, team members from other design groups confirm whether the developed system satisfies the requirements.

3.3 Team Organization

All of the team members are cross-listed in the team roster shown in Table 1. Figure 5 shows the team organization chart. The estimate for the total human-hours spent on this project was 4600. All works were carried out under the IRUP for the IGVC.

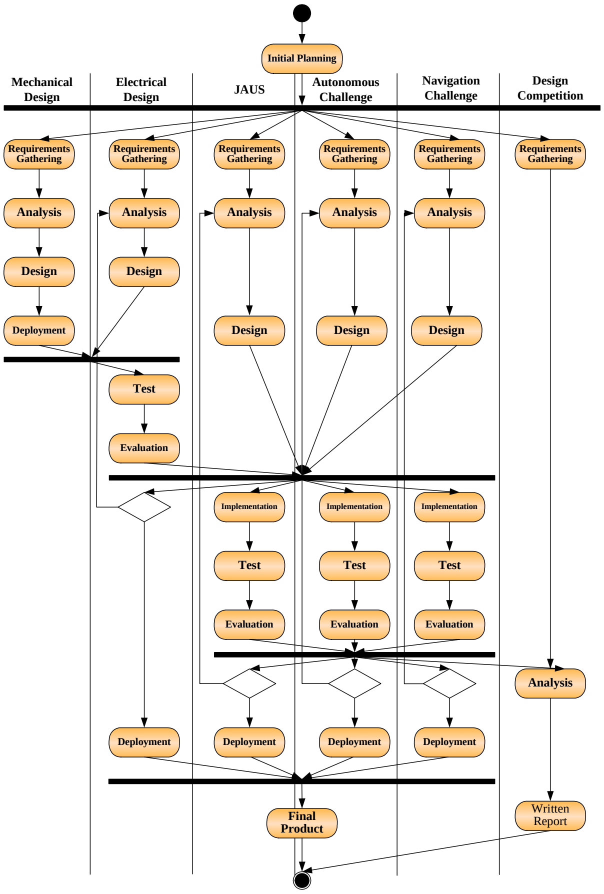


Figure 4: IRUP for the IGVC project

Table 1: Team members

Team Member	Class Standing	Hours Spent
Yuki Tarutoko	Graduate Student, System Engineering	810
Ryotaro Kotake	Graduate Student, System Engineering	790
Manabu Shimizu	Graduate Student, System Engineering	700
Takeyoshi Sasaki	Graduate Student, System Engineering	580
Shin Amano	Graduate Student, System Engineering	430
Zyuniti Kubota	Graduate Student, System Engineering	310
Mikihiro Ando	Graduate Student, System Engineering	230
Yoshitaka Goto	Undergraduate, Systems and Control Engineering	130
Hidenobu Sakazaki	Undergraduate, Systems and Control Engineering	130
Makoto Sugiura	Undergraduate, Systems and Control Engineering	100
Satoshi Sibata	Undergraduate, Systems and Control Engineering	100
Yusuke Misono	Undergraduate, Systems and Control Engineering	100
Masaru Onishi	Undergraduate, Systems and Control Engineering	50
Masataka Kato	Undergraduate, Systems and Control Engineering	50
Syuiti Kubota	Undergraduate, Systems and Control Engineering	30
Humiya Sato	Undergraduate, Systems and Control Engineering	30
Yousuke Torikai	Undergraduate, Systems and Control Engineering	30

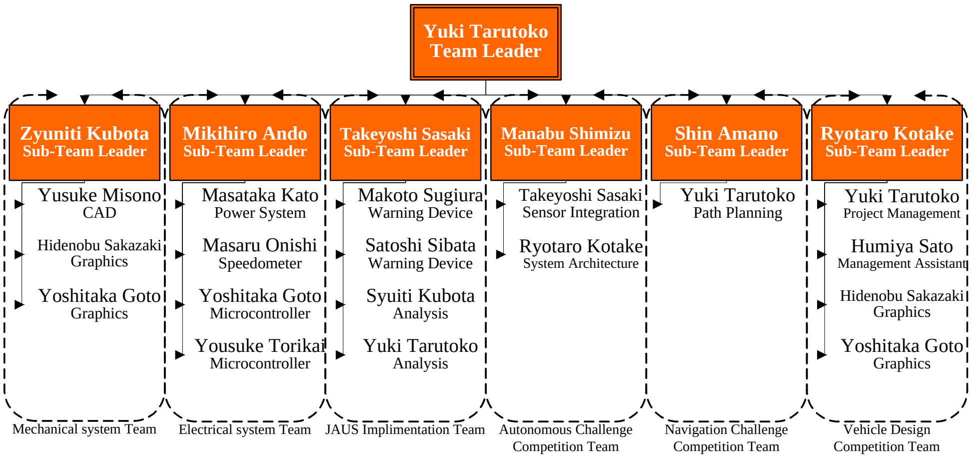


Figure 5: The team organizational chart

4. Mechanical Design

To create a unique autonomous vehicle to respond to the written report requirement, we decided to emphasize the concept “Omni” more than that of the previous Amigo series in the “Initial Planning” process. In repeated discussions through member meetings in the “Requirement Gathering” process, we selected the strategy of building a new vehicle rather than improving the previous Amigo series.

4.1 Base Vehicle

The base vehicle is shown in Figure 6. It is the Patrafour manufactured by KANTO AUTO WORKS, LTD, which can turn with zero-radius. Employment of this commercially available electric wheelchair as the base guaranteed chassis reliability and considerably reduced the mechanical manufacturing time. Omnix2006 is based on this unique mechanism. The keys to the enhanced performance of zero-radius turning lay in the AWD system and in the front omni-wheels.

Figure 6: Base vehicle

4.1.1 Actuators

The actuators to drive the vehicle are two 38Ah 24-volt DC motors originally mounted on the electric wheelchair. The maximum power of the motors is 280 watts. The power for the motors is supplied by two 12-volt batteries. A unique belt system is employed to transmit the driving power from the rear motors to the front wheels. Thus, this system along with the Omni-wheels design enables powerful zero-radius turning as a simple mechanism without a steering actuator.

4.1.2 Omni-Wheels

Figure 7 shows the mechanism of the Omni-wheels. Each Omni-wheel consists of cups around the wheel. Each cup rotates laterally; thus, the Omni-wheel can drive laterally without a special steering control mechanism. When the wheelchair goes straight forward or backward, the cups do not rotate. When the rear wheels are directed to turn, the cups rotate laterally and the wheelchair

Figure 7: Omni wheels

turns smoothly.

4.2 Chassis Modification

Through the team discussion for evaluating the design process of the Amigo series in the "Analysis" part of the IRUP, many problems were identified. The main problem of the previous vehicle was a heat problem in the electric circuits due to housing and outside temperature. We failed to take into account the fact that the intensity of the sun in Michigan is much stronger than that of Tokyo. Through the course of several discussions, we referred back to the design of Amigo2004 instead of Amigo2005. As in Amigo2004, we set a fan at the best site in the electric circuit housing. Also, from the idea of Amigo2004, light-weight materials for the electric housing were selected, which kept the center of gravity of the vehicle at an optimal point in terms of vehicle stability and drivability.

Next, we paid attention to easy assembly as well as durability. Because of the regulations of the airplane industry, the whole vehicle must be capable of decomposing into several elements. Design efforts focused on how to ensure easy assembly while improving the durability of elements, which may incur shock during transportation. In the "Analysis" process, we designed the appearance and analyzed the strength of the elements using the graphic software called Shade7 (e frontier) and the three-dimensional CAD software Autodesk Inventor10 (Autodesk). Shade was mainly used for graphic-appearance design while Inventor was used as the actual "Design" of the Omnix2006. The functional separation of the two different softwares directly correlated to the "Analysis" and "Design" processes, respectively.

Figure 8: Three-dimensional representations of the Omnix2006

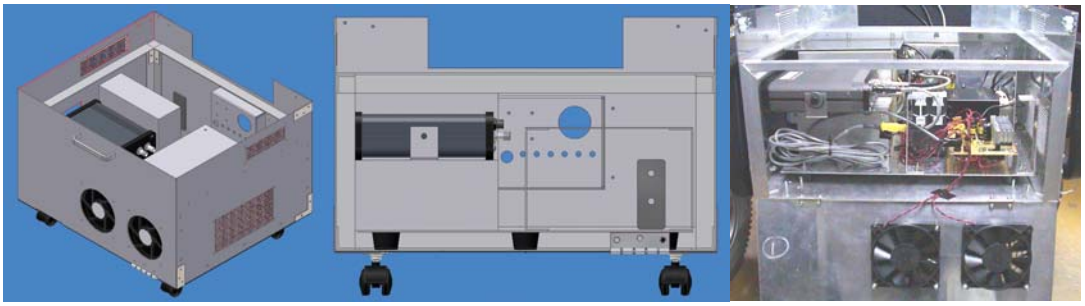


Figure 9: The CAD design of the electric housing box (left side) and the actual design (right side)

5. Electrical Design

As in the mechanical design, we employed the concept of “Omni” in the electrical design. We employed a 2-axis accelerometer sensor and an HMD (Head Mount Display) by which omni-directional images can be shown. These two devices are combined and provide a new omni-directional virtual reality display system by which a hands-free joy-stick functions. When our face turns to the left, the vehicle also turns to the left. This function can be used for debugging of programming and demonstration. In the non-autonomous vehicle application, the employment of this strategy leads to a novel driving scheme for handicapped people which can be actually be used in next-generation wheelchairs.

5.1 Sensors

The sensors incorporated with the Omnix2006 are an omni-directional camera (SONY CCD EVI-370 with hyperbolic mirror), a laser rangefinder (LRF)(SICK LMS-200), two rotary encoders set to control the two motors, an optical fiber gyro (HITACHI HOFG-3) to detect the angle of the vehicle, a digital magnetic compass (HONEYWELL True Point) to estimate the accurate self-position and self-orientation of the vehicle, and a differential global positioning system (Trimble BD950) to locate the position of the vehicle with respect to the latitude and longitude coordinates of the earth. The video frame images are grabbed using a video capture USB card (IODATA USB-CAP2), converted into a digital format using VCAPG2, and sent to the software programmed by MATLAB for image recognition. The average sampling interval of the laser rangefinder is about 20ms. Again, this information is sent to MATLAB through an RS422 interface for range profile recognition. The vehicle velocity measured by rotary encoders and an optical fiber gyro provides accurate dead-reckoning. The differential global positioning system (Trimble BD950) is based on a dual-frequency GPS. It provides the latitude and longitude information of the vehicle’s position.

5.2 Computers

Software in a laptop computer determines the navigation route in an off-line manner and generates on-line control signals from data by the sensors above. The laptop computer is driven by a 1.6GHz Pentium M processor with 512MB of memory. The operating system is Microsoft Windows XP Professional.

5.3 Power System and System Integration

Figure 10 shows how the sensor signals cables and power supply wires are connected and integrated. The image signals from the omni-directional camera are transmitted to the PC via a USB image frame-grabber. The laser rangefinder scans the front plane of the vehicle with 1/2 degree resolution in the 180 degree range. The laser rangefinder signal is transmitted to the PC via a high-speed serial RS-422 with 500 kbps. A DGPS signal is transmitted to the PC via a serial RS-232C to a USB converter; the optical fiber gyroscope is also connected to the PC via a serial RS-232C to USB converter while the speedometer and digital magnetic compass are connected to the PC via USBs.

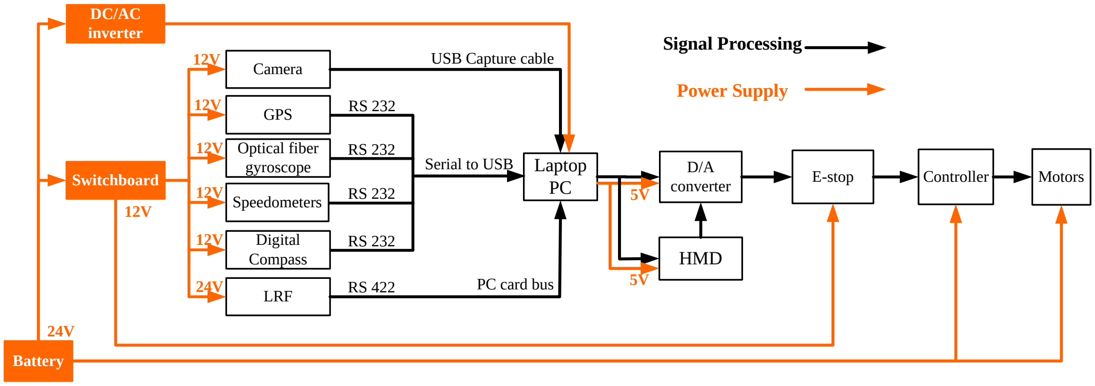


Figure 10: Outline of the electrical systems

5.4 Safety

We designed two different types of emergency stop (E-stop) to follow the rules on IGVC. The first one is a manual push-button switch located on the mast of the vehicle, the position most easily accessible by hand. The second one is an E-stop via wireless transmission. The stop signal is transmitted by an automobile wireless engine starter. It can transmit signals in a wide range with a maximum distance of about 100m (330 feet). In addition, if all of the batteries of the vehicle are off or missing, the vehicle is mechanically stopped and locked.

6. JAUS

Implementation of JAUS is the theme of the new Challenge. We repeatedly read through the Reference Architecture 3.2 during team meetings in the "Analysis" process. JAVA provides a standard TCP/UDP communication class; thus, the documents of the JAVA TCP/UDP class greatly help us to understand the JAUS communication protocol. Figure 11 shows how we integrated the JAUS communication system into the electrical system for vehicle operation. The JAUS message commands from the IGVC developed by the Operator Control Unit (OCU) via an RF data link are received by the wireless Ethernet converter (BUFFALO WLI-TX1-G54) and the information is processed by the Microcontroller (RENESAS H8/3069). Most of our efforts in implementation were focused on analyzing message contents, which we did using UML, JAVA and MATLAB. Adding to the challenge was the fact that JAUS manuals are written in English, which is difficult for us as non-native English speakers to understand for implementing the protocol. We spent about 6 months and several meetings on these efforts.

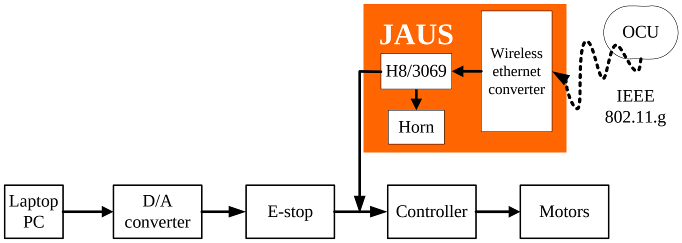


Figure11: The JAUS into the electrical system

7. Software Strategy

The software system was developed based on IRUP with MATLAB script language. One of the advantages of IRUP is the iterative process in software development, i.e., "Analysis," "Design," "Implementation," "Test," and "Evaluation." These processes are closely connected through UML communication between team members. By applying this approach, we were able to develop reliable software within the limited period of development, also because MATLAB script-programming environment enabled rapid prototyping. The combination of MATLAB and IRUP perfectly fit the software development needs and activities of our team.

7.1 Software for the Autonomous Challenge

7.1.1 Selection of soldier or philosopher architecture for basic control software

Amigo2005 placed 14th in the 2005 IGVC Autonomous challenge. After the competition, we analyzed the software system, specifically asking why our vehicle could not navigate with complete accuracy. Through repeated "Analysis" processes, we found two critical problems in the software. The first problem was a failure in lane detection when several obstacles exist; the obstacles in the image were masked to simplify the lane-detection algorithm. This simplification hid the lane and the wrong path was generated. The second problem occurred in the vehicle control due to incomplete image recognition. Incomplete image recognition was used for steering control to avoid slowing down the vehicle. To solve these two problems, we surveyed the following software system architectures: 1) behavioral decomposition architecture (soldier) and 2) functional decomposition architecture (philosopher). The advantages and disadvantages of the two architectures are as follows.

1) Behavioral decomposition architecture

Advantage: Behavior-based architecture is especially responsive to emergent situations. The controller operates independently for different emergency situations, which make the architecture robust and easy to extend.

Disadvantage: The architecture relies only on immediate sensory inputs; it does not permit fusions of signals from different sensors, nor integrate the sensor signals with the a priori knowledge. Thus, the architecture has weak reasoning and planning functions. They work well only in a specific environment with a limited set of behaviors.

2) Functional decomposition architecture

Advantage: Using this architecture, prior knowledge can be used to produce reasonable intelligence. It also facilitates decomposition of design tasks and assignment of subtasks to different team members who specialize in different domains, such as sensory processing, motor control, etc.

Disadvantage: All the sensory data must be fused and all tasks must be coordinated before taking any action; this introduces unnecessary delay time, which is always the cause of trouble in a real-time feedback system. In addition, this architecture is unreliable if any part of the system is functioning incorrectly.

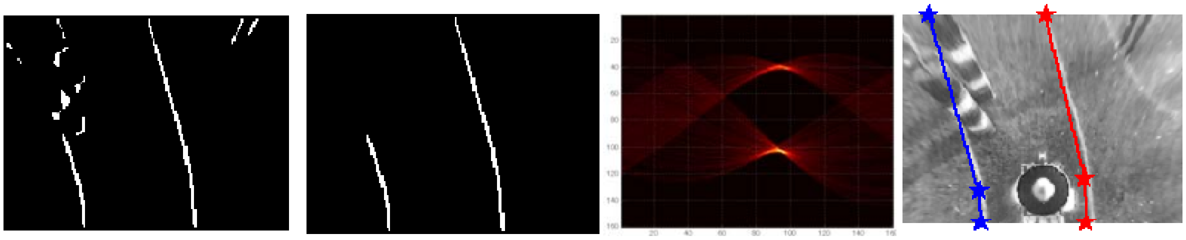
Through the several team discussions about control software architecture, we finally choose functional decomposition architecture, because in our vehicle a variety of sensing devices suitable for vehicle control are mounted. Details of the architecture based on the philosopher type

(2) are described in “2. Design Innovations.”

7.1.2 Robust Lane Detection

In outdoor environments, the position of the sun strongly influences image recognition. The shadows of trees or other obstacles can create false lanes and/or false obstacles. Reconstruction of images grabbed by the omni-directional camera to ground images enhances the lanes so that their determination is not influenced by the shadows in the original image. Figure 12(a) shows an original image grabbed by the omni-directional camera. Figure 12(b) shows the reconstructed ground image. After the reconstruction, we convert a RGB color image to grayscale image using only the B component. Figure 12(c) shows the grayscale image. To suppress high-frequency noise in the grayscale image, we applied median filtering. Figure 12(d) shows the image after median filter application. By using a referenced lane template image prepared ahead of time, normalized template matching is applied to detect the lanes. This technique is robust to noise and sensitive to lanes. The template-matched image is converted to binary image by comparing thresholds. Figure 12(e) shows the binary image. The isolated noise in the binary image is removed by the combined algorithms of the labeling and morphological thinning processes; this is called logical filtering. Figure 12(f) shows the logically filtered image. Finally, the Hough transform, which extracts straight lines in images, is applied to detect lane lines. When the image has a steep curve, the Hough transform algorithm recognizes that there are several lines in the image which correspond to multiple peaks in the $\rho-\theta$ Hough domain. Thus, if multiple peaks are detected in the $\rho-\theta$ Hough domain, the lane curve is approximated by piece-wise linear segments. Implementing such sophisticated lane-detection algorithms, the Omnix2006 proved reliable at detecting lanes even in cases when the lines were hidden by obstacles or drawn only by dashed lines. Figure 12(g) shows the plots in the Hough domain and Figure 12(h) shows the lane detected. The lane lines detected can be stored as sets of starting points and end points and line-crossing points.

(a) Input image (b) Reconstructed image (c) Grayscale image (d) Median filtering image



(e) Binary image (f) Noise elimination (g) Hough domain (h) Lane detection

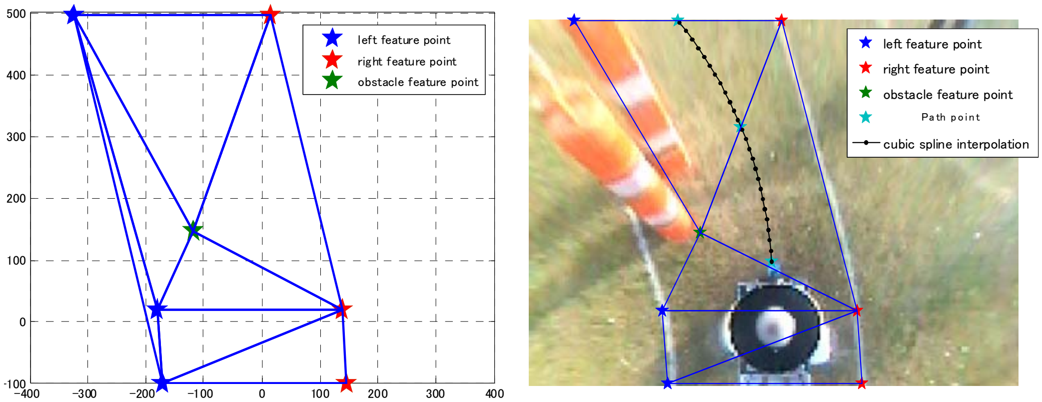
Figure 12: Result of lane detection

7.1.3 Path Generation

To generate an appropriate path, it is necessary to assign path points and the moving direction along the path. The path generation algorithm is organized by the Delaunay triangulation method. The triangle in the method is determined by the lane lines and the edge of the obstacle area detected by both the omni-directional camera and the laser rangefinder.

The lane-line area consists only of the features of the lane lines. The path direction can be easily defined by middle points on the Delaunay edge, which is connected by the left and right sides of the lane-line feature points.

The obstacle area consists of several obstacle points as well as the features of the lane lines. Depending on the position of the obstacles and the lane lines, an allowable navigating direction can be determined and indicated. From the allowable navigating direction, a new Delaunay triangle can be generated according to the positions of the lane lines and obstacles. Modified path points are then generated based on the new Delaunay triangle. The modified path points by the algorithm thus described are shown in Figure 13(a). After the path point generation, cubic-Spline interpolation is applied to paths given by the sequence of path points. Figure 13(b) shows a path generated by the proposed method.



(a)Path point generation (b) Path generation

Figure 13: Result of path generation

7.1.4 Vehicle Control

To navigate the vehicle from the current position to the next position through the generated path, we employed the open-loop control strategy based on the dead reckoning technique. A local two-dimensional positioning sensor consisting of a dual differential speedometer and an optical fiber gyroscope is used for dead reckoning.

7.2 Software for the Navigation Challenge

Through detailed investigations of the navigation software of Amigo2005, we found serious bugs related to obstacle avoidance. These bugs degraded the performance of the vehicle's navigation. In Omnix2006, we totally redesigned and rewrote the software for obstacle avoidance.

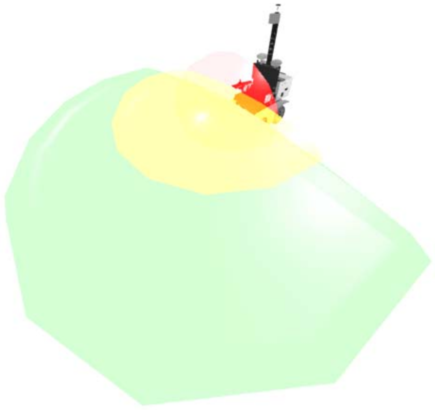


Figure 14: Virtual safety barrier

The basic idea of the new obstacle avoidance algorithm is as follows. In Amigo2005, once an obstacle is identified, the algorithm automatically changes modes from long-term path planning to short-term path planning. In generating the short-term path, virtual waypoints selected so as not to collide with obstacles are assumed from the laser rangefinder information. Short path planning is for stable and safe driving without collision and slows down the vehicle's speed. In Amigo2005, short paths were planned only by direct information from the laser rangefinder, and for certain groups of obstacles, it led to failure. To properly adapt to various types of obstacles, we developed a new concept called "virtual safety barrier." The "virtual safety barrier" takes into consideration the limitations in the resolution of the laser rangefinder. The "virtual safety barrier" consists of three layers. The first layer, category A, is for obstacles such as barrels clearly detectable by the laser rangefinder. The second layer, category B, is for obstacles such as fences and/or nets that may be difficult to detect by the laser rangefinder alone. The third layer, category C, is for emergent situations that may not be detected as either categories A and B. Figure 14 shows a diagram of the proposed "virtual safety barrier."

The resolution of the laser rangefinder limits the detectable angle and the maximum distance as follows.

$$R = \Delta d \cdot \frac{180}{\pi} \cdot \frac{1}{\Delta \theta} \quad (1).$$

r_i : Length to target measured by the rangefinder

Δd : Euclidean distance between r_i and r_{i+1} .

$\Delta \theta$: Angular resolution

R : Maximum detectable distance

For category A, based on the resolution and the size of barrels and eq.(1), we set the maximum distance R_A =10m. For category B, from the resolution or maximum distance at which nets and/or fences are detected by the rangefinder, we set R_B = 6m. For category C, from the resolution at which the range finder detects unavoidable obstacles to stop or reverse, we set the maximum distance R_C = 1m. Detailed navigation strategies for each category follow.

Category A: If the vehicle identifies obstacles such as barrels in the directional path, virtual waypoints are generated from the obstacle position and lane locations to avoid collision. Even if the vehicle detects several obstacles, the vehicle can generate an appropriate path based on generated virtual waypoints without slowing down the vehicle speed.

Category B: The laser rangefinder may fail to detect the nets and/or fences. To overcome this problem, we use information from prior experience to identify the nets or fences.

Category C: When the vehicle detects obstacle of category C, the vehicle must slow down, stop or reverse the moving direction of the vehicle, depending on the detected profile of obstacles.

The new algorithm was extensively tested in both simulation and real field experiments. The results of real field experiments showed that the algorithm is robust. We believe the performance of Omnnix2006 in the navigation challenge will be more reliable and stable than that of last year’s Amigo2005.

8. Analysis of Predicted Performance and Results

The overall performance and quality of Omnix2006 was much higher than those of the Amigo series. Field tests results of Omnix2006 are shown in Table 2.

Table 2: Analysis of predicted performance and results

Performance Measure	Competition	Prediction	Results
Maximum speed		4.25 mph (6.8 km/h)	4.06 to 4.25 mph (6.5 km/h to 6.8km/h)
Maximum swing speed		120 deg/sec	110 to 120 deg/sec
Ramp climbing ability		10 degree incline	8 to 10 degree incline
Reaction times	Autonomous	0.10 to 0.25 seconds	0.10 to 0.28 seconds
	Navigation	0.15 seconds	0.15 to 0.20 seconds
Battery life		5 hours	3.8 to 4.25 hours
Obstacle detection distance	Autonomous	5 meters (Omni-directional camera and LRF) [maximum]	
	Navigation	10 meters (Omni-directional camera and LRF) [maximum]	
Traps, and potholes	Autonomous	Detection of obstacles: (LRF) Detection of potholes: template matching (Omni-directional camera) The obstacles detection methods are same as that of Amigo2005	
	Navigation	Detection of obstacles: (LRF)	
Dead ends		The vehicle performs a near zero radius turn until a suitable path is found	
Waypoint accuracy		0.25 to 0.50 meters	0.25 to 1.00 meters
Remote emergency stop capability		250 meters [maximum]	80 to 100 meters [maximum]

9. Cost

The costs involved in the development of the Omnix2006 are summarized in Table 3.

Table 3: Estimated costs for development of Omnix2006

Item	Cost	Remarks
GPS receiver	$10,000	TRIMBLE (BD950)
Laser rangefinder	$8,500	SICK (LMS-200)
Optical fiber gyroscope	$5,800	HITACHI (HOFG-3)
Electric powered wheel chair	$5,310	KANTO AUTO WORKS (Patrafour)
Hyperbolic mirror	$4,600	
Laptop personal computer	$2,000	FUJITSU (Intel Mobile Pentium M 1.6GHz)
Head mount display	$1,620	SHIMADZU (Data Glass 2/A)
Digital magnetic compass	$1,575	HONEYWELL (True Point)
Isolated analog output module for USB	$660	CONTEC (DAI12-4(USB)GY)
CCD camera	$360	SONY (EVI-370)
Automobile wireless engine starter	$160	SANTECA (RS-1500)
USB video capture cable	$123	I-O DATA (USB-CAP2)
Wireless ethernet converter	$100	BUFFALO (WLI-TX1-G54)
Microcontroller	$60	H8, PIC and PSoC
Power inverter (DC 24V to AC 100V)	$35	CELLSTAR (HG-150/24V)
Rotary encoders	$34	IWATSU
Mechanical parts	$312	
Electronics parts	$298	
Totals	**$41,547**	

10. Conclusions

This report has described the design process, development, and construction of Omnix2006. The Omnix2006 has a novel omni-directional camera as a sensing device and a novel Omni-wheel chassis. Despite the limited period of development, employment of the IRUP and UML design approach facilitated smooth communications between team members. The new design process has facilitated the development of software as well as hardware integration. We believe we did our best to develop the Omnix2006.