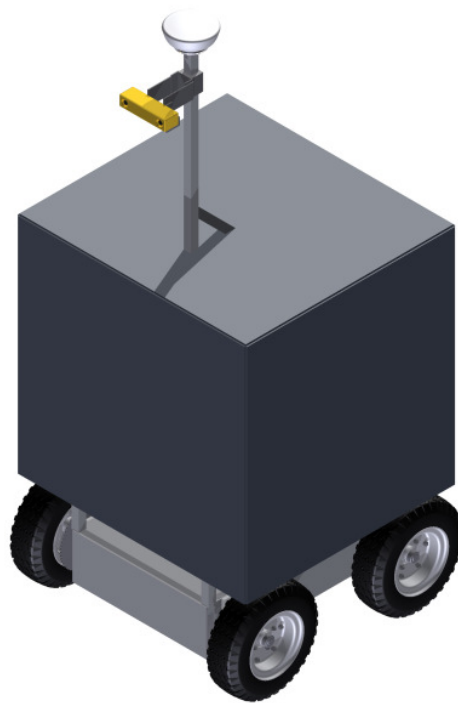


Kratos



Chicago Engineering Design Team

The engineering design in this vehicle by the current student team is significant and equivalent to the credits that would be awarded in a senior design course.

Changes from the Previous Year

Last year, the team competed with two vehicles; Enyo and Kratos. However, the team decided that Kratos was a better candidate to for the competition due to its smaller bottom chassis. Since the wheels are closer to the center of rotation, the amount of power needed to turn was well as the strain on the drive-train would be reduced. Therefore, Kratos was redesigned to make use of Enyo's electrical system.

The electrical system's main design has stayed the same. However, many of the layouts for the boards made it difficult to etch and assemble. This year, the boards have had their layouts optimized in order to ease assembly and reduce failures.

In the past, software design utilized pre-existing frameworks such as Orca and Gearbox to operate the vehicle at a lower level. By designing a system without utilizing any pre-existing framework, greater control of all parts of a system can be achieved. The system is no longer limited by the design choices made by third parties. As a result the system is very agile and can be extended and molded to the specific needs of the vehicle.

Design Strategy

The team is composed of three main groups; the mechanical, electrical, and software groups. Each group has its own team meetings but all groups meet together once per week to keep everybody informed as well as to provide feedback across the groups. All main decisions are made during these meetings. Figure 1 highlights the main process used to design the vehicle. Objectives are taken from the rules of the competition and used to create the problems to be solved. Once designed and tested, the problems or the solutions may need to be modified based off of testing data.

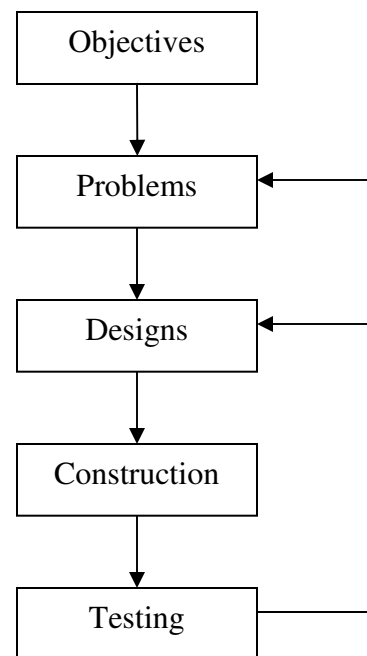


Figure 1: Team design process

Mechanical Design

The mechanical aspect of Kratos is designed to be rugged, easily maintained, and provide ease of accessibility to internal components. All materials aside from drive-train components were machined in house. The following sections highlight the main mechanical components.

Chassis

Kratos' chassis is divided into two sections, a welded steel bottom chassis and an aluminum T-slot top chassis. The bottom chassis houses the drive train and wheel encoders. Welded steel provides structural support to the drive-train which can produce significant forces during operation. The top chassis houses the batteries, computer, electronics, and sensors. T-slot is used because it allows for quick fabrication and access to internal components for maintenance. All components are placed in order to keep the center of gravity as low as possible to the ground and centered in the vehicle. The electronics are placed on a single shelf which can be easily slid out from the T-slot frame. Since the batteries are heavy and difficult to remove, a drawer was designed to allow the batteries to be pulled out from the chassis so the user can easily replace them.

Drive-train

A differential drive system is used for Kratos where two 24V motors are used, one for each side. This system allows the vehicle to turn in place or turn in a radius if moving forward and backwards; which gives the software more flexibility in maneuvering around objects. The radius of motion can be controlled by adjusting the ratio of speeds between the two sides of the vehicle.

Electrical Design

Kratos contains a robust electrical system rooted in the tenets of reliability, safety, and efficiency. These tenets are realized by designing all components to safely handle and recover from a multitude of error conditions. To increase reliability, each electrical component is securely mounted with clearly labeled ports and is modular. Furthermore, all microcontroller systems enter idle mode when they are not doing work, thereby reducing power consumption,

and increasing efficiency. All boards were etched and soldered in house. The following sections highlight the electrical components of Kratos.

Power System

Kratos is powered with a set of six 12 volt 26 amp-hour sealed lead acid batteries. Four of them are arranged in parallel in order to form a 12 volt bank to power the electronic systems and the DC-DC ATX pico power supply for the computer. The remaining two batteries are arranged in series with each other in order to provide 24 volts to the motor-controllers that control the motors. All electronic module voltages are regulated using 85% efficiency switching regulators.

E-Stop

The emergency stop system (E-stop) is designed to disable the vehicle in emergencies. The E-Stop may be activated either by remote control or by pressing the onboard switch. The E-stop is composed of two modules: a handheld transmitter unit that wirelessly transmits a signal to the vehicle and a receiver unit that is onboard the vehicle. The transmitter unit has a highly visible red pushbutton switch that when pressed, activates the E-stop. The behavior of the E-stop is that if either the onboard switch or the wireless transmitter provides the STOP command, the vehicle will stop moving. The vehicle will resume movement only when both sources provide a GO signal. The radio module uses spread spectrum technology to ensure data encryption and prevent interference or jamming from external sources, thus improving system reliability. In addition, the radio module has a range of 5 miles in open-air line of sight.

Embedded Drive Train Control System (EDCS)

The EDCS is designed to interface the software controls system to the motors that drive the vehicle. The system consists of a module for each side of the differential drive. Each module is responsible for measuring the wheel speed and controlling the power level of the motor for its side. A microcontroller is the heart of each module and communicates with the software control system via RS-232 serial communication. The communications protocol between the EDCS and the computers is designed to inherently detect communication errors, leading to improved system reliability and safer operation.

Sensors

Kratos has 5 types of sensors that provide data feedback to the computer software.

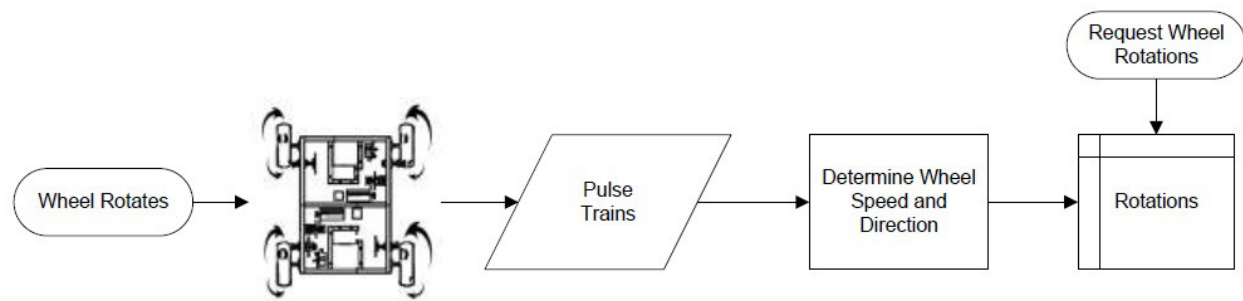
1. Two wheel encoders are used to measure the approximate speed of each side of the vehicle (Fig 2A). The encoder produces two pulse trains with frequencies linearly dependent on wheel speed. The encoder sends the two pulse trains to the EDCS, which then determines the wheel speed and direction of rotation.

2. Eight sonar units are used to measure approximate distances to objects; four sonar units are placed in the front of the vehicle, and four in the back of the vehicle (Fig 2B). The sonar units produce a pulse train with frequencies linearly dependent on the distance to the object. The sonar units then send the pulse train to the sonar control module that measures the pulse width and sends it to the computer.

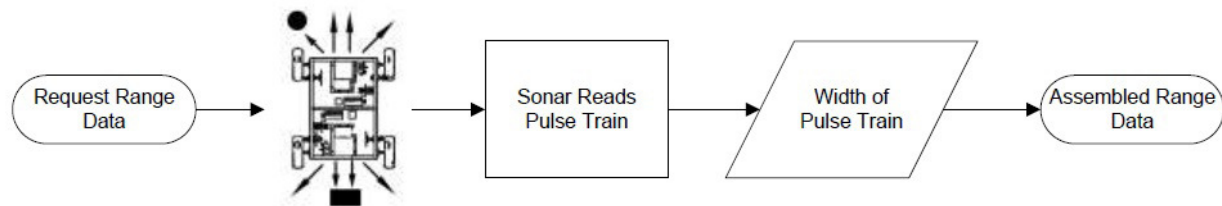
3. An Inertial Motion Unit (IMU) is used to track the vehicle's accelerations in three dimensions. The IMU control board measures the readings from the IMU module and communicates the data to the computer.

4. A stereo vision camera is used for line detection as well as object detection. Two cameras mounted a fixed distance apart allows a three dimensional reconstruction of the environment and obstacles.

5. A global positioning system (GPS) is used to give the computer coordinate information on the vehicle's location. This information can be used to give commands to the vehicle.



A



B

Figure 2: Data Flow for the EDCS/Wheel Encoders (A) and Sonar devices (B).

Control Selector

The Control Selector allows the user to choose the manner in which Kratos is controlled. This system has four modes of operation that are easily controlled by the user through switches on the Control Panel. The first mode is a safety mode which causes the vehicle to remain stationary; it is also the default mode to ensure safety by not allowing the vehicle to move until a user deliberately allows it. The second mode allows the vehicle to be exclusively autonomously controlled. The third mode allows the vehicle to be exclusively controlled by remote control. The final mode allows the vehicle to switch between autonomous and radio control by the switch on the remote control. In addition, the system recognizes the absence of a signal from the remote control and stops the vehicle when no signal is detected.

Control Panel

The Control Panel interfaces with every electronic system on the vehicle through the I2C interface, allowing it to send commands and receive data. In this sense, the control panel acts as a medium between devices, as it is the only device capable of controlling others; if one device needs access to another, it must indirectly access it through the control panel. The secondary

ENYO
Control Panel

SYSTEMS		EDCS 1		EDCS 2		STATUS	
ESTOP	GOOD	80%	2m/s	-20%	-1m/s	WiES	GO
SigMu	GOOD	SONAR 1				OnES	STOP
SysMo	ERR1	500cm	206cm	198cm	514cm	SigMu	Auto
EDCS1	GOOD	SONAR 2				CB1	GOOD
EDCS2	GOOD	600cm	600cm	600cm	600cm	CB2	POP
Son1	ERR2						
Son2	ERR2						
						12V	GOOD
						24V	CRIT

Figure 3: Control Panel Display

function of the control panel is to provide status information from devices and forward the information to an LCD screen (Fig 3) and to the computers for feedback. This feedback greatly increases vehicle reliability and safety since it informs the software of possible hardware system errors. Status information includes Signal Multiplexer status, E-stop status, EDSCS data, battery status, and fan status. One can adjust settings for the Signal

Multiplexer and fan operation using buttons located on the Control Panel. Other features include ports for connecting monitors, keyboards, and mice to the computers, permitting quick access to the computers. Other features include ports for connecting monitors, keyboards, and mice to the computers, permitting quick access to the computers.

System Monitor

The System Monitor is responsible for keeping track of vehicle resources, performance, and malfunctions. If malfunctions occur, the System Monitor notifies the Control Panel, which in turn notifies the software and the user. One key task of the System Monitor is to continuously measure remaining battery life and warn if the battery capacity is becoming low (Table 1). The system has two types of battery warnings for both the 12 volt and 24 volt systems, and another type of warning for each motor. Based on these warnings, the user or software can take action to correct these faults, allowing the vehicle to run in the safest manner possible.

Event	Electrical Response	Software Response
Battery Low	Msg: Change Batt. Soon	Limit Speed
Batt. Critically Low	Msg: Change Batt., Switch on E-stop	None
Motor Pwr. Lost	Msg: Motor Pwr. Lost	Wait

Table 1: System Monitor Warning Chart

Computer Systems

The vehicle contains one high-performance computer to do image analysis, sensor data analysis, and control of the vehicle. The computer uses an Intel Core 2 Quad Q8200 CPU, giving excellent performance due to a quad core processor that allows four different software components to run concurrently. The image processing is performed on an NVIDIA GeForce GTX 580 video card which supports CUDA. The computer uses a high performance ASUS P5E motherboard due to their resistance to overheating. It has 8GB of RAM to handle large amounts of data without having to rely on slow hard drive caching, letting the computer process data continuously at high speeds. A 32GB solid state drive (SSD) is used to store data. SSD's were chosen for their resistance to vibration damage during vehicle operation.

Summary of Electrical Components

All the components can be broken down into three categories; sensors, control circuitry, and plant (Figure 4). Most sensors give information directly to the computer which acts as the primary controller for autonomous mode. All control is sent to the EDCS where it is converted into movement information and sent to the control selector. The control selector is used to determine whether to send the remote control signal or EDCS signal to the motors. The control panel is what gives the user the information from the computer and the status of the robot.

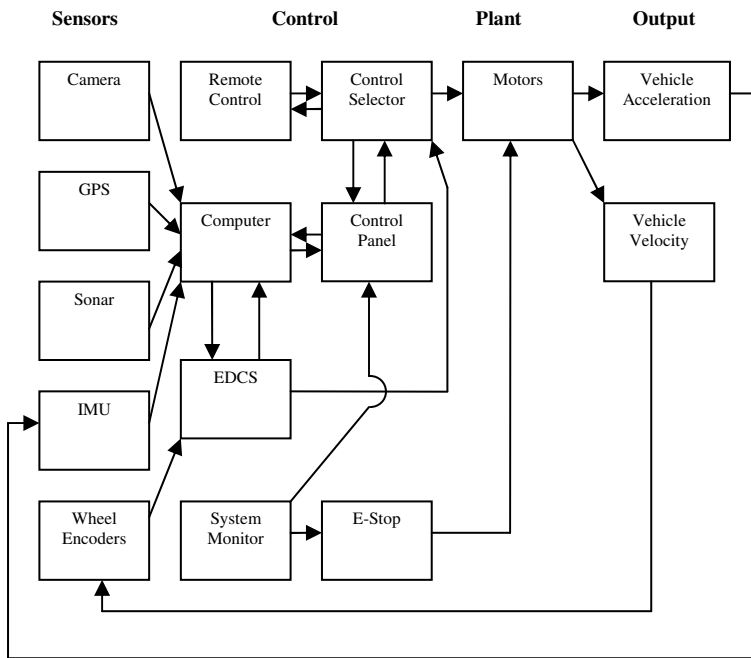


Figure (4): Electrical Flow Diagram.

Software

The robot utilizes a system written in Java, called Lobodeimos. This is a “proof-of-concept”/prototype design of the Deimos system which is in development currently planned for use in the 2013 IGVC. The features and design implementations utilized in Lobodeimos are a small subset of those to be present in Deimos. The focus for Lobodeimos is a minimal subset to achieve the required objectives.

The minimal requirements breaks down as follows:

1. GPS location awareness – Required to navigate to provided waypoints
2. Heading awareness – Required to accurately home in on a specified waypoint
3. Disparity – Required for obstacle avoidance
4. Line Detection – Required for line avoidance
5. Wheel Controllers – Required to control location

This set of requirements leads to a naïve navigation system. This appropriately represents the minimalistic design of the Lobodeimos prototype. This system is displayed in Figure 5.

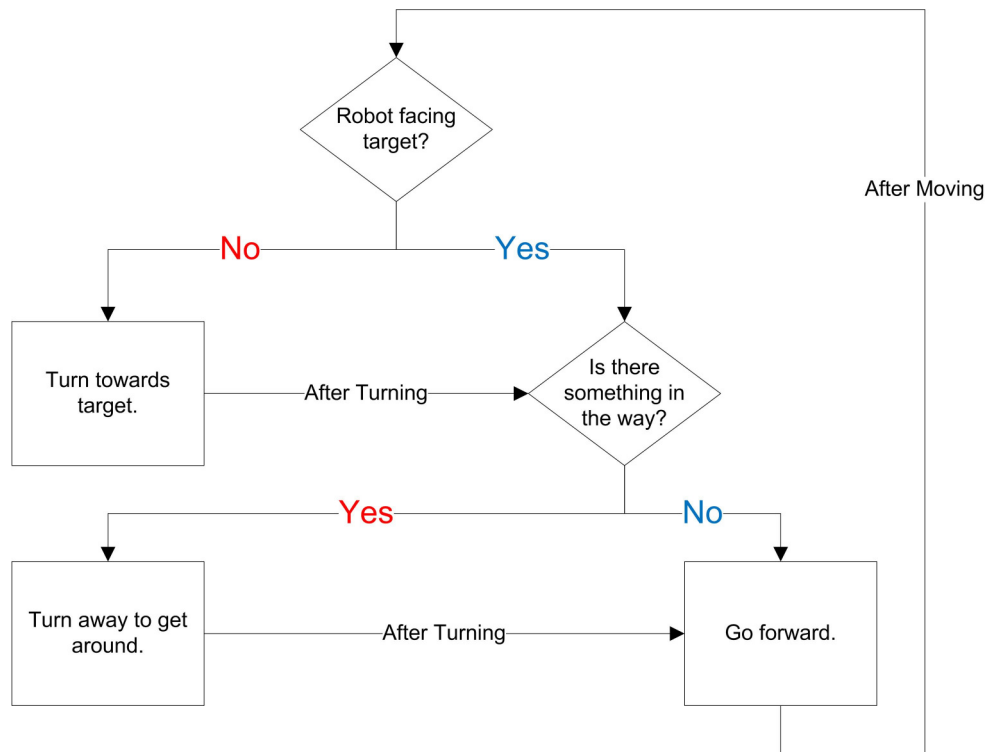


Figure 5: Decision making algorithm

Java was chosen as the high level language to work in primarily for its ease of implementation and increased familiarization by the majority of the developers involved with the project. Though efficiency is traditionally viewed as lower in a Java implantation than a natively compiled language like C++, much of this has been overcome through a combination of CUDA and the Java HotSpot JVM (introduced in Java 1.3) which provides on-the-fly compilation to native code of frequently executed code blocks.

The Deimos/Lobodeimos system is designed to be modular and configurable allowing for minimal time to changing working parameters for operation as well as decreased lag time in integrating new hardware systems. This helps decouple the hardware from the software, allowing more flexibility in design of hardware systems moving forward.

The current implementation uses the following:

1. Java – Primary language used to program vehicle.
2. jCuda – For running disparity using the GPU drivers.
3. javaCV – Open source library for line detection.
4. JNA/ javacpp – Library to convert camera image drivers to be used in Java.
5. jSSC – Serial port communication for USB connections.
6. junit – Used for running unit tests

Plans for the full Deimos system include integration of:

1. Sonar – To detect objects closer than camera can detect.
2. Inertial Measurement Unit (IMU) - To determine vehicle positioning.
3. Dynamic mapping – Mapping and path finding of course during operation.

Performance Characteristics

- **Actual Speed** – Max speed is 7.3 mph
- **Actual Ramp Climbing** – Vehicle is tested up to a 25 degree slope
- **Actual Battery Life** – 26 minutes with motor power supply as limiting factor
- **Actual Obstacle Detection** - 2 ft minimum, 15 ft maximum with camera
- **Actual Waypoint Accuracy** – 2.5 ft diameter

Vehicle Costs

Component	Price
Lead Acid Batteries	\$600
GPS	\$1600
Stereo Camera	\$2000
Raw Materials for Chassis	\$400
Drive-train Materials	\$300
Motors	\$1100
Electronic Components	\$800
Sonar Units	\$300
Wheel Encoders	\$750
Computer	\$1300