


Stony Brook University
Stony Brook Robot Design Team

Wilhelm

2015 IGVC Competition

**Thomas Bundy, Steven Leo, Michael LiBretto, Anthony Musco, Michael Pullman,
Szymon Puzdrowski, Yuka Saijo**

I hereby declare that the mechanical, electrical, and software changes implemented in the design and engineering of the vehicle by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.



Professor David Westerfeld
Department of Electrical Engineering
Stony Brook University

Date: 15 MAY 2015

Table of Contents

1 – Abstract	3
2 – Team Organization	3
3 – Mechanical Design	3
3.1 - Chassis	3
3.2 - Suspension	4
3.3 - Wheels	5
3.4 - Motors.....	5
4 – Electrical Design	5
4.1 – Power Source and Distribution	5
4.1.1 – Batteries	6
4.1.2 – Power Distribution	6
4.1.3 – Charging System.....	6
4.2 – Systems	7
4.2.1 – Motor System.....	7
4.2.1.1 – RoboClaw 2x30	7
4.2.1.2 – Watchdog Circuit.....	7
4.2.1 – Computer System.....	8
4.2.1.1 – Navigation Computer.....	8
4.2.3 – Emergency Stop (E-Stop)	8
4.2.3.1 – Physical E-Stop.....	8
4.2.3.2 – Wireless E-Stop	8
4.2.3.3 – E-Stop Integration.....	9
4.2.4 – Peripherals System.....	9
4.2.4.1 – Camera	9
4.2.4.2 – Pan-Tilt	9
4.2.4.3 – Laser Range Finder.....	9
4.2.4.4 – GPS	9
4.2.4.5 – IMU.....	10
5 – Software Design	10
5.1 – Framework Backbone	10
5.2 – Hardware Interfaces	10
5.2.1 – GPS Interface	11
5.2.1 – IMU Interface.....	11
5.2.3 – Laser Range Finder Interface.....	11
5.2.4 – Pan-Tilt Interface	11
5.2.5 – Motor Interface	11
5.2.6 – Camera Interface.....	12
5.3 – Algorithms	12
5.3.1 – Camera Algorithms	12
5.3.1.1 – Exposure Control	12
5.3.1.2 – Object and Line Detection	12
5.3.2 – Navigation Algorithms.....	13
5.3.2.1 – Kalman Filter	13
5.3.2.2 – Decisions.....	14
5.4 – Error Handling	14
6 – Cost	15

1 – Abstract

Project Wilhelm is the first attempt at a complete rebuild for the Intelligent Ground Vehicle Competition by the Robot Design Team at Stony Brook University since 2011 and we are taking it in a direction that the club has yet to approach. Rather than approaching the competition with a vehicular robot in the conventional sense, the robot was designed to be omni-directional in order to facilitate a finer positioning of the robot within the course. Furthermore, rather than relying on a swing-arm suspension like in previous years we moved toward a more conventional double wishbone suspension found in the rear suspension of older automobiles. In addition to the mechanical overhaul, the electrical system was designed as a split power system with improvements in how power is distributed through Wilhelm. The software utilizes a lot of the same code from last year with improvements in inter-module communication, error detection, and navigational algorithms.

2 – Team Organization

The Stony Brook Robot Design Team is a multidisciplinary club based out of Stony Brook University. There are 3 design sub-teams: the mechanical team, electrical team, and software team. Each team is led by a design team leader, who is in charge of approving the sub-team's design, creating and maintaining timelines, ensuring proper testing is done, and creating the sub-team's budget. The 3 design team leaders report to the Vice-President, where the 3 separate designs are merged together into a single unified design. The President, Vice-President, Treasurer, and Secretary are in charge of managing the club and its assets, including obtaining all the parts needed by the design team leaders. This organizational structure helps to keep the design team leaders focused on the designs and making them as optimal as possible.

3 – Mechanical Design

3.1 – Chassis

In order to fully utilize the position capabilities given by holonomic drive, the robot was made to be symmetrical about its center. This was done by designing the main chassis using concentric octagonal layers to house the motors, the electrical distribution materials, the payload, and the electronics and I/O equipment. See figures 3.1a and 3.1b for the CAD drawings of the chassis.

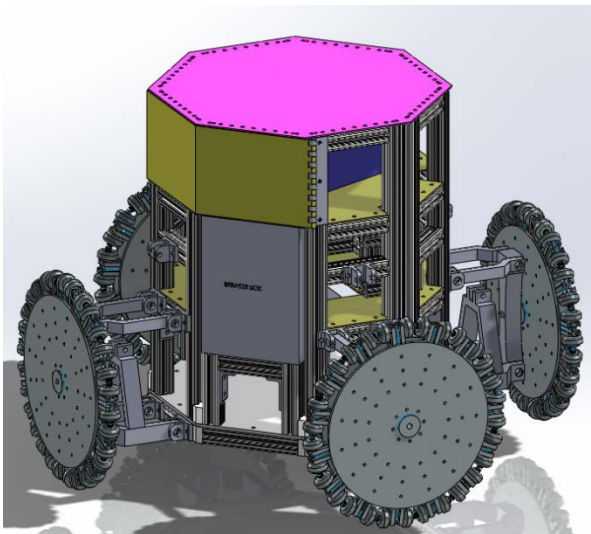


Figure 3.1a: Lower Chassis CAD Drawing



Figure 3.1b: Upper Layer of Chassis CAD Drawing

The bottom layer was constructed with an 80/20 1.5 inch aluminum extrusion frame to provide mount-points for the lower control arm suspension tabs and to provide a rigid skeleton for the upper

layers. The surface is a quarter inch aluminum plate to minimize deflection from the motors. The vertical frame-members are held in place using pem studs that thread into the center hole in the vertical extrusion. The studs anchor themselves in counter bored holes on the underside of the plate and are held in position by the threads from the verticals. The verticals are held upright using L-brackets on two sides. The motors are mounted straight to the plate.

The next layer is supported on 7 flanges cut into the .125 inch aluminum plate. The main chassis structure is interrupted by the breaker box, providing immediate access to the breakers on the exterior of the robot. Two batteries are used for powering the motors and the electronics for the robot. The larger of the two is suspended on a platform that occupies the space over the motors to save vertical space. The smaller of the two is fixed next to it and each has wires routed to the breaker and the rest of the robot. The breaker is held on to the robot by two sets of L-brackets holding it to the second and third layers.

The third layer is mounted similar to the second and holds a box to hold the payload. The box is made from Lexan and is held on using L-brackets. The payload is secured with Velcro straps across the front of the box, and the layer is closed using a door that spans two faces of the octagonal base.

The layer above this secures the electronics equipment and is made with a half-inch plastic. The weight of the electronics should not be large enough to cause major deflection in the plastic. Above this is a gazebo-like structure constructed from steel welded into shape. Steel was chosen for its weldability and the ability to better shape the structure. The particular shape was chosen to allow for ease in future weatherproofing as well as minimize the area of the platform required to be rotated on the slip-ring.

The upper most level contains a rotating ring that is connected to the computer systems through a slip-ring. The slip-ring is turned with an MX 106T servo affixed to the slip-ring through a custom connector and a .5 inch axle. The slip-ring is then affixed to the upper plate and its stator is held in place with a pin connected to the upper platform.

3.2 – Suspension

The CAD drawing of the suspension can be seen in figure 3.2. The suspension tabs are mounted to the exterior of the chassis using U-bracket and aluminum flat stock. The tabs are used to mount the double wishbone suspension to the chassis. This suspension was chosen in order to force our wheels to have contact with the ground because of traction issues discovered last year. The length of the control arms was chosen to fit the robot within the size limits as well as limit the maximum throw of the wheel. The shocks used are go kart shocks with a maximum load of 175lbs, which over four corners of the robot will allow for some spring in the riding height of the chassis. The suspension works as a four bar linkage with the shock providing a counterforce against the chassis of the robot. This allows for limited vertical motion by the upright mounted on the end of the control arm. As the terrain under the robot changes the suspension should extend down and force contact with the ground.

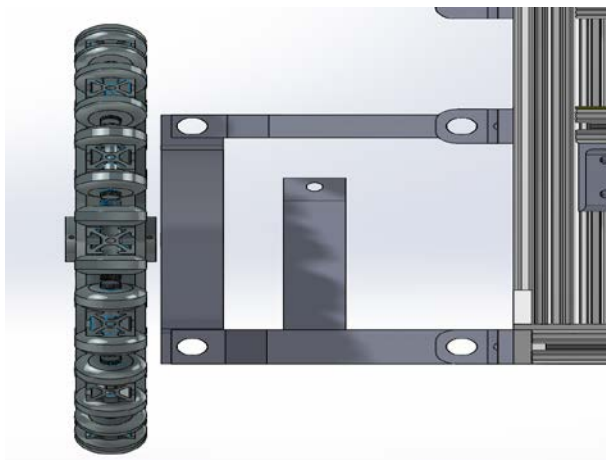


Figure 3.2: Suspension CAD Drawing

3.3 - Wheels

Wilhelm utilizes omniwheels driven by four independently controlled motors. Not being able to purchase wheels with the desired diameter, about 19 inches, plans were made to design and fabricate them in house. Figure 3.3 is a CAD drawing of the omniwheel design. Failure analysis revealed that the axles holding the two rollers will fail first if a significant jarring force is incurred from hitting rough terrain. The wheels are comprised of twelve spokes 30 degrees apart from each other sandwiched by two fastening plates 15 inches in diameter. The spokes extend past the fastening plates and have two inline skate wheels at the end secured on an axle with 5/8th inch washers and hex bolts. An Allen key set screw is screwed into the outward facing spoke hole to lock the roller axles in place. Fully assembled, each wheel are 19 inches in diameter and approximately 12 pounds. The inline skate wheels act as the rollers which allow for translational motion. The hub acts as the axle mount using two spring pins to keep the axle in place. Quarter inch bolts are used and fastened with quarter inch washers and hex nuts to keep the assembly secure.

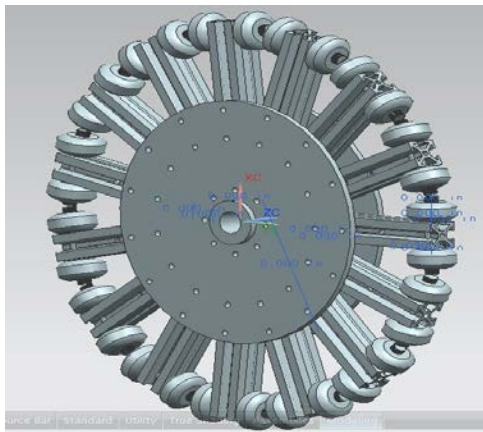


Figure 3.3: Omniwheel CAD Drawing

3.4 – Motors

Wilhelm is approximately 250 pounds and as such needs powerful motors for proper holonomic drive functionality. The motors chosen are 12V DC motors rated for 0.5 HP and 1750 RPM at free-spin. These were selected for their high power, but also because Wilhelm is powered by 12V, not 24V like last year's robot. The reason for this change will be further explained in Chapter 3, section 3.1.1. The motors do not have a gear reduction on them and thus are quite large in order to supply such power. Each motor weighs 30 pounds. The motors are current limited from drawing the maximum of 39A to only drawing 15A due to limitations with the battery supply.

4. Electrical Design

The electrical design of Wilhelm consists of the distribution of electrical power throughout the system and the all electronic systems on board. Figure 4.1 shows how power is distributed throughout Wilhelm.

4.1 – Power Source and Distribution

Wilhelm is capable of drawing a maximum of 1.3KW. For such a high power machine, a power source with the capacity to supply 1.3KW had to be found and a sophisticated power distribution system had to be designed in order to protect both the electrical systems on-board and those servicing the robot.

Project Wilhelm

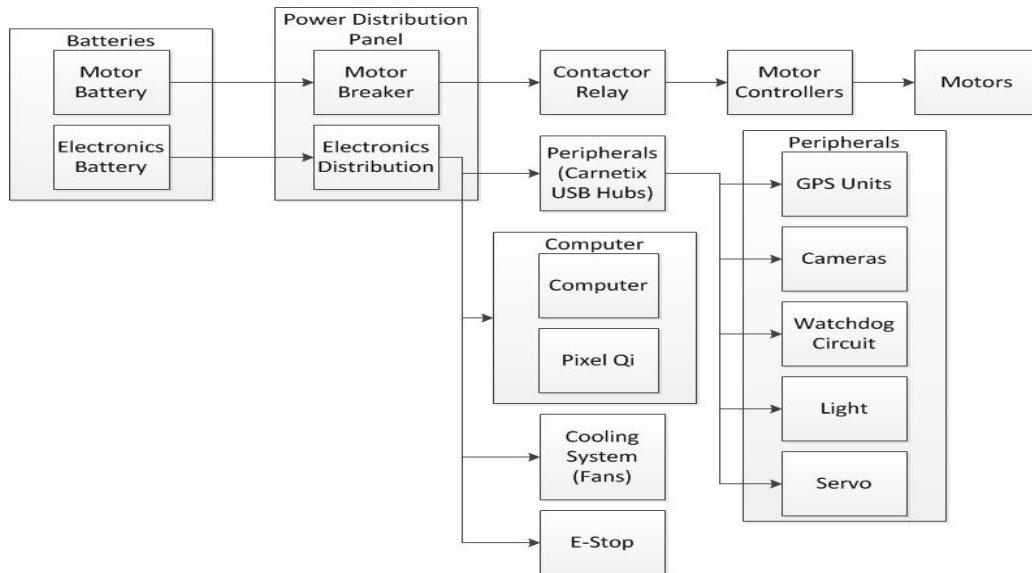


Figure 4.1: Distribution of Power in Wilhelm

4.1.1 – Batteries

Wilhelm is powered using two 12V marine batteries. Battery 1, which from this point shall be referred to as the electronics battery, has a 34Ah capacity at 20 hours. This battery is dedicated to running the logical electronics on-board Wilhelm. Battery 2, which from this point shall be referred to as the motor battery, has a 67Ah capacity at 20 hours. This battery is dedicated to running the motor controllers and all 4 motors. The separation between the electronics battery and the motor battery was made in order to prevent current spikes caused by the motors to adversely affect the logical electronics. Large instantaneous draws on the batteries by the motors can cause temporary voltage drops across the batteries. This creates voltage droops for the lower current drawing logical systems which in turn can cause undesirable results. Thus, the decision to separate the batteries into two distinct purposes was made so that a voltage drop across the motor battery will not affect the electronics battery. Both batteries do share a common ground though in order to maintain proper communication between the logic systems and the motor controllers.

Marine batteries were chosen for the electronics and motor batteries due to the ability of marine batteries to supply a moderately high current continuously as well as drop to near dead with minimal to no damage to the battery. These are very useful traits for marine applications where a sea-faring vessel might spend days at sea draining the batteries unrelentingly. Wilhelm does not need to last on a single charge for very long, but it does need to be able to draw large currents continuously without the risk of potential damage. The model batteries chosen use lead-acid AGM technology, which supports a stabilized voltage even as the battery nears the end of its charge. Li-Ion batteries also have this feature, but the cost involved with purchasing Li-Ion batteries that would be capable of sourcing the currents necessary for proper operation and the specialized charger to charge said batteries was far outside the budget. As such, the marine batteries were the best compromise of charge capacity, voltage stability, and cost. The major drawback is the weight which had to be accounted for in the mechanical design.

4.1.2 – Power Distribution

Each battery feeds independently into a distribution panel. The distribution panel is used to isolate the individual systems from each other. This is necessary in isolating potential problems in a specific circuit, such as a short circuit. The head of each branch at the distribution panel has a circuit breaker rated for the specific capacity of each system. This means that the computer system, which can draw up to 300W, has a much larger breaker than the E-Stop system, which can only draw up to 60W.

The presence of the breakers is two-fold. The primary reason is for protection in case something goes wrong in a particular system, both for the system with the problem and all the other systems that could be affected by a surge. The second reason is for ease of powering on/off each system.

In addition to the circuit breakers for each individual system, a master breaker is capable of shutting down the entire robot at once. The master breaker is electrically in line between the electronics battery and all the logical electrical systems. Despite not being electrically connected to the motor controllers, the master breaker has the power to shut down the motor controllers as well because the E-Stop system discussed in chapter 4, section 4.2.3 disables power to the motor controller via the Kilovac Contactor.

4.1.3 – Charging System

Wilhelm has a protected charging port on-board in order to charge the batteries without having to remove them. This is important as the process of moving batteries introduces risks of damage to batteries due to accidental shorting. By implementing a charging system, the risks are eliminated and the robot has a more finished and professional look. The charging port does connect directly to the battery terminals electrically. This presents a risk of accidental short via the charging port. To mediate this risk, a charging port enable switch has been added which disconnects the charging port from the battery terminals when the port is disabled. In addition, in-line fuses have been added between the enable switch and the battery terminals to prevent current surges through the charging port.

4.2 – Systems

Most of the components within each system are retail devices which were chosen for their power efficiency and the quantity and quality of their features. There are a few custom designed components. These components were designed with efficiency in mind, but with safety as top priority.

4.2.1 – Motor System

The motor system is the only system that operates off of the motor battery. It has a dedicated circuit breaker at the distribution panel. After the breaker, the power passes through a Kilovac Contactor which has the capacity to switch up to 500A. The Kilovac contactor is the control mechanism for the E-Stop to kill power to the motor controllers and ultimately the motors.

4.2.1.1 – RoboClaw 2x30

The motor controllers chosen for Wilhelm are the RoboClaw 2x30 motor controllers. There are two on-board as each supports driving two motors independently and there are four independently powered motors on Wilhelm. These were selected for their encoder support and ease of communication with the computer. The encoder support was a necessity because holonomic drive requires precision when driving the motors and the only way to achieve such precision is with a feedback loop involving encoders.

4.2.1.2 – Watchdog Circuit

Although the RoboClaw 2x30 motor controllers have many features, one feature they lack is a built in watchdog. The watchdog allows the motor controller to shut down if communication from the computer ceases. This is of utmost importance because if the software crashes, the Wilhelm must stop all motion for safety reasons. As such, a watchdog circuit was designed and placed in-line between the computer and the motor controllers. The circuit consists of two 555 timers in mono-stable configuration. This provides a count-down which has been configured for 200ms. This means that if the computer stops communicating to the motor controllers for more than 200ms, the watchdog will shut down the motor controllers.

4.2.2 – Computer System

The computer system includes the navigation computer and the Pixel Qi display. The Pixel Qi display is a specialized display designed specifically for outdoor viewing. The display is a special form of LCD with a backlight that can be toggled on or off. When the backlight is on, the screen appears like any normal LCD screen. However, if in direct sunlight, the backlight can be switched off and the screen turns to grayscale but remains perfectly readable due to ambient light. This is very useful for debugging code while outside in the sun and since the display is part of Wilhelm, there is no need to carry around a screen to debug.

4.2.2.1 – Navigation Computer

The navigation computer is a requirement for any robotic venture. The computer on-board Wilhelm has a 2nd generation Intel i5 processor on a mini-ITX motherboard with 16GB of RAM. The motherboard was selected due to its small form-factor and its reduced power requirements. The i5 processor is a moderately high computational power processor for the navigation program while still maintaining less than a 77W TDP. In addition, a low-power version of the Nvidia GeForce GT 630 graphics card was added in order to utilize the GPU for intensive graphical computation due to the stereoscopic vision which will be described further in chapter 4, section 4.2.4.1. A solid state drive (SSD) is used as the boot drive. This was done because a mobile platform such as a robot cannot have a classical hard drive (HDD) running and expect reliability. The power supply unit (PSU) of the computer is an M4-ATX. The M4-ATX is a specialized 300W DC-DC PSU for low power computers. It will take as input 12V and convert down to all the necessary voltages for the proper operation of the computer.

4.2.3 – Emergency Stop (E-Stop)

The E-Stop is a custom designed circuit with two sub-components in it. The physical E-Stop is activated and deactivated through the press of a physical button on the top of Wilhelm. This will activate a hardware-only circuit that will turn off the motors. The wireless E-Stop is activated and deactivated through the use of a wireless remote. This will activate a microcontroller driven circuit that will turn off the motors. In both cases, when the E-Stop is in the off-state, the motors are prevented from receiving power through the use of the Kilovac Contactor. The E-Stop provides the logic to turn on and off the contactor.

4.2.3.1 – Physical E-Stop

The physical E-Stop operates through the use of the asynchronous inputs to a flip flop. The push-button is actually a DPST momentary button. It is configured such that when the button is not pressed, the asynchronous CLR is asserted and when it is pressed, the asynchronous PRE is asserted. An RC low-pass filter is utilized on each of the asynchronous inputs in order to mediate switch bouncing. The output to the flip flop is directed into the clock input of another flip flop which is configured to toggle on every clock edge. The output of the toggling flip flop is the current state of the physical E-Stop. This creates a physical E-Stop sub-system that is turned on by pressing the button and turned off by pressing the button again. When the E-Stop circuit is powered up, the physical E-Stop has a start-up circuit that ensures that the physical E-Stop starts in the off-state.

4.2.3.2 – Wireless E-Stop

The wireless E-Stop operates by using a microcontroller with an Xbee radio transmitter for wireless communication. On circuit power-on, the microcontroller will start the wireless E-Stop in the off-state. When the “On” button is pressed on the remote, the wireless E-Stop changes to the on-state and when the “Off” button is pressed on the remote, the wireless E-Stop changes to the off-state. The microcontroller will require constant pinging of the remote. This ensures that if the wireless E-Stop is in the on-state and the remote batteries die or the remote goes out of range, the wireless E-Stop will automatically switch to the off-state for safety reasons.

4.2.3.3 – E-Stop Integration

The integration between the two E-Stop sub-systems is done through a DPST relay. The physical E-Stop state provides the signal to be transmitted through the relay while the wireless E-Stop state provides the control of relay state. The idle state of the relay will provide a ground to a TIP-122 npn Darlingtong transistor, thereby turning the transistor off. The only way that the transistor can be turned on is if both the physical E-Stop and the wireless E-Stop are both in the on-state simultaneously thanks to the integration through the relay. The transistor turns on the logical output to the Kilovac Contactor which will result in the Kilovac contactor providing power to the motors. As soon as either E-Stop sub-system switches state, the logical output the Kilovac contactor is cut and the power to the motors is subsequently cut. With this design, it is guaranteed that if the E-Stop system is not powered, the motors cannot be powered. This was an important design decision as it ensures that the motors can't be accidentally run without the protective circuitry being in a ready-state. It is also important to note that the E-Stop system has no interaction with the navigation computer.

4.2.4 – Peripherals System

The peripherals system circuit consists of three powered USB hubs that provide the power to all the sensors on-board Wilhelm and provides the connection between the peripheral devices and the navigation computer. This was done for system isolation as it is not necessary that the peripheral devices be powered whenever any work on the navigation computer is being done.

4.2.4.1 – Cameras

Wilhelm features two on-board Microsoft LifeCam Studio cameras. This provides stereoscopic vision for the navigation program to use for navigation. These were mounted on a Pan-Tilt system that has the capability to pan freely and tilt $\pm 30^\circ$. This was implemented so that the navigation program can control the direction the cameras are facing independently of the direction the robot is moving which is useful for looking ahead or calculating the best path around an obstacle. This also allows Wilhelm to move freely in any direction without ever rotating thanks to holonomic drive and the dynamic cameras.

4.2.4.2 – Pan-Tilt

The Pan-Tilt system was developed using two Robotis MX-106 servos. One servo controls the free panning and the other controls the tilting of the cameras. Thanks to the free panning of the cameras, a slip-ring was used to ensure that the cameras can pan continuously in one direction without the wiring ever presenting an issue.

4.2.4.3 – Laser Range Finder

The laser-range finder is a high-efficiency indoor/outdoor infrared range detector. It is capable of detecting range down to 1cm resolution with accuracy around 10cm in broad daylight. This design choice was made to provide a way of determining the distance of objects more accurately than through stereo vision alone.

4.2.4.4 – GPS

There are two Adafruit Ultimate GPS modules on-board Wilhelm. The use of two GPS modules is done partially for a way to calculate a slightly more accurate GPS position by taking into account the positions each GPS module is reporting relative to where on Wilhelm each physically is and also as a back-up. The backup is helpful if one of the GPS modules suddenly starts sending bad data to the navigation computer or one cannot acquire a GPS fix while the other can. These are adaptable scenarios that will be in the navigation program.

4.2.4.5 – IMU

A 9-axis inertial measurement unit (IMU) will be employed by Wilhelm. The IMU has a 3-axis accelerometer, 3-axis gyro, and 3-axis magnetometer. The magnetometer supplies magnetic heading data although after testing, it was found that the heading data has very low accuracy due to magnetic fields generated by the battery discharging and the motors. The accelerometer supplies information about Wilhelm's current acceleration given its intended acceleration. This allows the navigation program to use accelerometer data as feedback to how quickly the robot can get up to speed. This however is a much larger feedback loop than the encoder feedback loop to the motor controllers and must be used as feedback on the proposed calculation rather than ensuring that the robot is actually moving accordingly. The gyro supplies information regarding whether or not Wilhelm is perfectly vertical or not. This is useful in determining when an incline is encountered or if one of the four wheels becomes caught.

5. Software Design

The object detection, path finding and decision making of the robot are implemented by an embedded system. Please refer to chapter 4, section 4.2.2.1 for more information regarding the navigational computer. A framework developed while programming last year's robot, Project Tina, was utilized extensively this year. As such, this year's program is the next generation of Project Tina's program.

5.1 – Framework Backbone

Wilhelm's base framework is an event driven architecture that allows for modular implementation of hardware. Major sensors and controllers are implemented through modules which are then fused into one working unit by the base module loader. The modules have no knowledge of each other, but they communicate over an event bus controlled by the framework. This approach fosters highly modular and safe programming while decreasing the chance of a complete failure caused by a single module.

Separate logging and diagnostic modules have been implemented as part of the framework to allow the program to detect any problems ranging from hardware failure or disconnect through dangerous voltage and current spikes to getting stuck in the terrain. The decision to make the logging and diagnostic modules statically built in the framework is two-fold. First, both modules need to have knowledge of the other modules. Second, the static nature of both modules helps to reduce the potential for failure within each module. This is very important as both modules help to supply the team with information on critical failures of all the other modules. If a critical failure scenario occurs, the program will stop the robot and perform an analysis of the situation, while logging all of its data for later review by the team. Upon reaching a conclusion on the current state of the robot, the diagnostic module will decide whether to try to fix the problem, advance despite the failure, or stop completely.

The program start is embedded in the framework code which then calls the module loader. The module loader loads each module independently and spawns at least one dedicated thread for each module. The modules will be explained further in the following sub-sections, but the modules are ultimately separated into two categories: hardware interface and algorithmic. The only exception is the camera module which contains two sub-modules: one hardware interface and one algorithmic.

5.2 – Hardware Interfaces

The various hardware interfaces are handled by independent modules. Each hardware interface has a dedicated bus for communication from the computer. This simplifies the program as a queue system of which component is allowed to communicate on the bus at any time does not need to exist. Instead each module operates its own hardware bus and communicates via the appropriate protocol that each module was explicitly designed for.

5.2.1 – GPS Interface

The GPS module communicates via UART to both GPS units. The GPS units send industry-standard NMEA sentences and the GPS module extracts information such as: UTC, latitude, longitude, altitude, ground speed, angle, date, and magnetic variation. The GPS units update the module at 10Hz, but the module only updates the Navigation module at 1Hz. This is done to supply as accurate a location as possible to the navigation module. It also allows for partial responses due to communication errors which can happen and will be discarded. Thus, the GPS module will average together all the valid GPS responses and respond push the current GPS location to the framework the result. The framework handles the pushed data and notifies the Navigation module of an update.

5.2.2 – IMU Interface

The IMU module communicates via I²C to the IMU. The IMU data is updated at 1Hz. The data received is a 16 bit signed number to describe the acceleration in each of the X, Y, and Z axes, a 16 bit signed number to describe the magnetic field strength in each of the X and Y axes, and a 16 bit signed number to describe the rotation in each of the X, Y, and Z axes. The IMU module combines the X and Y axis magnetic field strength into a heading value. It then pushes the data up to the framework which will then notify the Navigation module.

5.2.3 – Laser Range Finder Interface

The laser module communicates to the laser range finder via I²C. Although the I²C bus allows for daisy-chained devices, it was decided that the laser range finder would use its own dedicated I²C bus because separated buses are easier to program and the electrical sub-team could afford to add an additional I²C hardware bus. The laser range finder could take up to 500ms to respond with a valid distance. This is because the laser range finder will make 1000 samples and then returns only the averaged distance to the master. Since the distance is returned to the laser module, the laser module simply pushes the distance to the framework. The laser module, unlike the GPS and IMU modules, is not free running. This means that the laser module, although it has its own thread, does not take constant readings. It will only make a reading when the Navigation module sends the appropriate signal to the laser module. This is because the laser range finder's purpose is to obtain more accurate distance measurements of objects. Thus, the laser range finder should only be used when locked onto the object of which the distance is desired. The laser range finder is locked onto an object through the use of the pan and tilt servos.

5.2.4 – Pan-Tilt Interface

The PanTilt module interfaces with both servos through a proprietary Robotis protocol. The PanTilt module, like the laser module, is not continuous. This module will only update the servos when the Navigation module sends a signal to do so. Since the PanTilt module controls where the cameras and the laser range finder are pointed, the PanTilt module will be utilized anytime a change in direction is enacted or a more accurate distance measurement is needed.

5.2.5 – Motor Interface

The motor module interfaces with the motor controllers. There are two motor controllers on board Wilhelm, as discussed in chapter 4, section 4.2.1.1. The motor module communicates with both through a packet serial protocol on a daisy-chained UART variant. This protocol allows for very simple and easy communication between the motors and the navigation computer. The motor module does not behave like the other modules. The motor module acts more like an extension of the navigation module than an actual module. This design gives the navigation module a direct pipe into the motor module, which does have its own dedicated thread for seamless interfacing, in order to prevent erroneous signals from affecting the motor module. Since the motor module is ultimately the most dangerous portion of the software design, it is important to limit the control to only the one module that has permissions to control it. This design

choice makes the navigation module the only one that can tell the motor module to start the motors and the only one that can tell the motor module to stop the motors. Since this could present a potential danger if the navigation module were to crash, there are fail safes in the framework that can kill the entire program if the navigation module fails to respond to heartbeats in a reasonable time. If the program shuts down, the hardware watchdog, as discussed in chapter 4, section 4.2.1.2, activates and kills the motors.

5.2.6 – Camera Interface

The camera module is also special but because it is actually split into two sub-modules. One sub-module handles interfacing with both USB cameras and retrieving the images at 30Hz, while the other handles the algorithmic side by processing the images received. The algorithmic side will be explained further in chapter 5, section 5.3.1. The camera interface is fairly simple because it takes the image from both cameras when they are available and passes the data on to the algorithmic half. However, the algorithmic portion will respond with how to change the exposure of the cameras. The interface must then change the exposure accordingly.

5.3 – Algorithms

The algorithms that run Wilhelm are stored in just two modules: the algorithmic sub-module of the camera module and the navigation module. The navigation module is in charge of all the decisions based on the data received from the other modules. The camera module is in charge of processing the data from the cameras into a form that the navigation module can use.

5.3.1 – Camera Algorithms

The camera algorithms take two images, one from each camera as input, and output a wealth of information extracted from each. First and foremost, both images are copied multiple times as there are 23 different threads operating concurrently to process the images as quickly as possible.

5.3.1.1 – Exposure Control

An exposure control algorithm borrowed from a paper by Inwook Shim, Joon-Young Lee, and In So Kweon, was utilized to achieve optimal brightness of the images at all times of the day. Since the camera data is so important in the navigational decisions, it is important that clear images could be taken for data to be extracted from. To achieve this, the algorithm looks for the exposure that will generate the most gradient data in the images.

The original images are copied and passed into the algorithm which then proceeds to copy each image an additional 6 times. Each of the now 7 copies of each image has a different gamma correction applied to it in order to simulate different levels of exposure. Following the gamma correction, the images have gradient data extracted. Based on those 7 “exposure” levels and the gradient information extracted, the optimal exposure can be calculated which is then passed to the camera interface to send to the cameras.

5.3.1.2 – Object and Line Detection

Objects are detected using gradients. A Gaussian blur is applied to smooth out the images and lessen erroneous noise in the image. Then the gradients of the images are calculated. After finding areas of maximum gradient change, the CANNY Line Detector is utilized to find both the white lines painted in the grass and the implicit lines of objects beginning and ending. Immediately, all white lines are stored as walls. If a series of objects in a row are detected, such as a series of cones side-by-side, all the cones are stored as a wall. If just a single object is detected, the object is stored as a point object for further processing. During this phase, the algorithm will be looking for general objects, but will also be specifically looking for the two parallel reflective strips on cones to identify an object as a cone. This provides a fast categorization of the object which allows for a better handle of which actions can be performed later on.

The data extracted from the CANNY Line Detector is passed into a SLAM algorithm which will attempt to match an object in both images and calculate the distance of the object based on the distance away from the focal point. The SLAM algorithm will also attempt to approximately map the location of the object, whether it is a point object or a wall, relative to Wilhelm. Wall objects will be populated with information of where they start, where they end, and at what angle they extend. Point objects will be populated with information of where the object is, how far away it is, an approximation of how wide the object is, and an approximation of the objects radius. The output of the SLAM algorithm will be a map of the field of view with all wall objects and point objects detected. This map is then passed to the navigation module for decision making.

5.3.2 – Navigation Algorithms

The Navigation Module ties all modules together and is responsible for the Wilhelm’s decision making based on what is reported by other modules. The general direction is determined from the current GPS position and the destination’s coordinates. An exact path is then calculated based on the detected obstacles. Finally the correct set of actions is determined and executed.

5.3.2.1 – Kalman Filter

A Kalman Filter is used to estimate as accurately as possible where Wilhelm is relative to the origin of the course, which is defined as the start point. A Kalman Filter is used to smooth away noise from the GPS and sensor readings to provide an optimal estimation of the state of Wilhelm. The state of Wilhelm consists of x (meters East of the origin location), y (meters North of the origin location), θ (heading in degrees clockwise from North), and v (meters per second). The state vector must be constructed from the GPS data by converting the latitude and longitude coordinates to N, E relative coordinates before it can be used by the Kalman Filter. The other parts of the filter algorithm include the state transition matrix F , the control transition matrix B , the observation matrix H , and the state covariance matrix P , each detailed below in figure 5.1.

$$\vec{x}_t = \begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix} \quad F = \begin{bmatrix} 1 & 0 & 0 & t \sin(\theta) \\ 0 & 1 & 0 & t \cos(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \frac{t^2}{2} & 0 & 0 & 0 \\ 0 & \frac{t^2}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad P = \begin{bmatrix} \sigma_{xx} & \sigma_{yx} & \sigma_{\theta x} & \sigma_{vx} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{\theta y} & \sigma_{vy} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_{\theta\theta} & \sigma_{v\theta} \\ \sigma_{xv} & \sigma_{yv} & \sigma_{\theta v} & \sigma_{vv} \end{bmatrix} = I_4$$

Figure 5.1: Description of Matrices Used in the Kalman Filter

Input can be directed towards the state vector by applying a control vector u consisting of the x acceleration, y acceleration, change in heading θ , and the change in speed v . The vectors w and v represent the process and observation noise, each having a mean of 0, with covariance matrices Q and R , respectively. The algorithm proceeds according to the equations shown in figure 5.2.

Project Wilhelm

Predict:

$$\begin{aligned}\vec{x}_t^- &= F\vec{x}_{t-1}^+ + B\vec{u}_t + \vec{w}_t \\ \vec{z}_t^- &= H\vec{x}_t^- + \vec{v}_t \\ P_t &= FP_{k-1}F^T + Q\end{aligned}$$

Update:

$$\begin{aligned}S_t &= HP_tH^T + R \\ K_t &= P_tH^TS_t^{-1} \\ \vec{x}_t^+ &= \vec{x}_t^- + K_t(\vec{z}_t^- + H\vec{x}_t^-)\end{aligned}$$

Figure 5.2: Update and Predict Stages

Note that the *predict* and *update* stages are performed continuously as new measurements are made from the sensory devices. Most of the time, u will be zero and the prediction stage will just consider the kinematic transition based on the Wilhelm's current state with additional noise. Q and R represent the covariance matrices for the process and observation noise and must be manually calibrated to achieve optimum filtering results.

The primary obstacle in implementing the Kalman Filter is processing the raw GPS data to allow the predict and update stages to run smoothly. The latitude and longitude coordinates must first be converted to the local N, E coordinate system used for navigation. In addition, the state transition matrix must be updated to reflect the proper trigonometric functions of the heading as shown in its definition above. In general, these are trivial procedures and do not cause any problems.

5.3.2.2 – Decisions

Wilhelm's decision algorithm is fairly simple. It takes the current state and the location of the next waypoint as inputs and outputs the direction that Wilhelm must travel in order to arrive at the destination waypoint. Wilhelm will always attempt to move forward to get to the waypoint. The obstacles detected by the camera algorithms will cause Wilhelm to veer off of a direct path. This form of decision making may not be the most advanced or the most optimal, but it is simple and easy to implement. If faced with a fork, Wilhelm will always choose the direction that will result in being closer to the waypoint.

5.4 – Error Handling

Every program should be able to handle any errors it could produce and Wilhelm's navigation program is no different. All error handling is covered by the diagnostic module, which was briefly discussed in chapter 5, section 5.1. The Diagnostic module is a static module, built into the framework. Error detection is the responsibility of the individual modules. If any module detects an error, no matter how insignificant, the module reports the error to the diagnostic module. The diagnostic module performs a quick analysis of the error and determines whether the error is serious. If the error is found to be serious, the diagnostic module signals the navigation module to stop Wilhelm as the diagnostic module performs a more in-depth analysis of the situation. During this analysis, the diagnostic module analyzes the current state and the raw information presented by the modules. A conclusion is made as to whether Wilhelm should continue, attempt to fix the error, or simply shut down. All information analyzed by the diagnostic module, as well as the conclusion, is logged for review later. The logged information can help with recreating an error situation so a proper fix can be made if need be.

The diagnostics analysis will conclude to attempt to remedy the error only in a few special cases such as performing a soft reset of a hardware device or attempting to detect and get out of ditches. Dangerous power situations, such as high current or high voltage will be looked at very carefully. These situations must be monitored very carefully in order to prevent irreparable damage or safety issues. As

Project Wilhelm

such, most of the time, if a power error is reported to the diagnostic module, the diagnostic module will conclude to shut down in order to allow the electrical team to look over the electronics.

If the diagnostic module concludes to continue, the diagnostic module periodically signal the navigation module to slow down or stop motion entirely in order to monitor the status of the original error status. If the error status ever worsens or is reported more than twice within 20 seconds, the diagnostic module will conclude to shut down due to potentially hazardous situations.

6 – Cost

Sub-Team	Item	Cost
Mechanical	Chassis	\$2,500
	Suspension	\$950
	Wheels	\$1,600
	Motors	\$1,450
Electrical	Batteries	\$500
	Power Distribution	\$200
	Motor Controllers	\$650
	Computer	\$950
	E-Stop	\$250
	Sensors	\$1,000
	Total	\$10,050