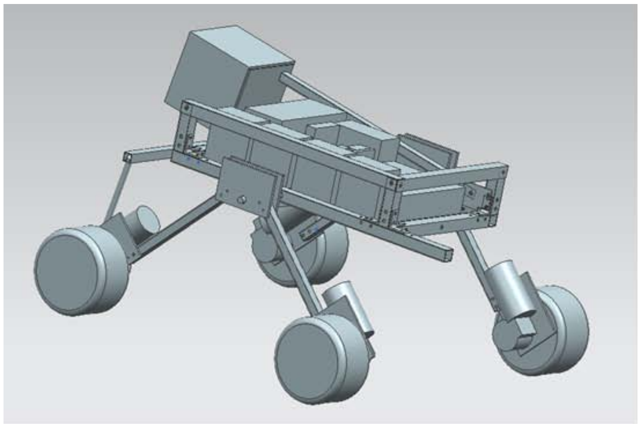
Frigg Remote Exploration System

Final Report for RoboOps 2015
University of Maryland
Department of Aerospace Engineering

Project Advisor:	Dr. David L. Akin Associate Professor of Aerospace Engineering Director, Space Systems Laboratory
Graduate Mentors:	Chris Carlsen Nicholas D’Amore Kevin Davis Donald Gregorich Nicholas Limparis Kate McBryan
Student Team Members:	Daniel Arble Marlin Ballard William Beckman Robert Brown Lemuel Carpenter Joshua Drubin Zeyad Emam Joshua Kaffel Shaheer Khan Joshua Pugh Elena Shrestha



Introduction

The University of Maryland is pleased to participate in the NASA RASC-AL RoboOps competition for the fifth straight year, and submits this final report to NASA and the NIA contest staff to document the design, development, and testing of the system leading to the 2015 competition. Although the original concept for this year was to make a number of selective modifications to the Persephone robot from the 2014 competition, as development progressed almost everything had to be redesigned and built from scratch, other than the core electronics hardware. Since the 2015 University of Maryland vehicle is essentially a new system, it was given a new name: Frigg, wife of Odin in Norse mythology, and yet another goddess of the harvest.

System Description

The strategy for the 2015 University of Maryland RoboOps entry was to maintain the basic vehicle configuration: four-wheel skid-steer, rocker suspension, with a dexterous (four degree-of-freedom) sampling manipulator to maximize collection volume (including reliable sampling at a variety of target heights) and minimize the need for fine adjustment of vehicle position during sampling. There were three overarching issues that led to our poor performance in 2014: heavy weight (44 kg), long communications latencies (up to 90 seconds during the competition), and poor manipulator performance. All of the design items detailed below were focused on remedying these deficiencies. An overall system block diagram for the Frigg rover is shown in Figure 1.

Mass

The final masses for the vehicle are shown in Table 1. These masses are based on weights of the final systems; it does not include miscellaneous items such as dust covers, mechanical fasteners, or incidental wiring (over and above the main wiring harnesses, which are accounted for under "electronics"). For this reason, a 30% margin has been added after the measured masses to conservatively estimate these incidental additions to the rover. This analysis shows that the original design goal of a 30-kg rover should be readily achievable.

Table 1: Mass allocations for Frigg rover

Component	Mass (kg)	% Total Mass
Wheels	3.2	9
Battery	3.6	10
Drive motors	4.7	13
Manipulator	1.0	3
Chassis	3.4	9
Electronics	5.0	14
Cameras and sensors	0.5	1
Main camera gimbal	0.8	2
Margin	6.8	30
Totals	29.4	100

Chassis Design and Drive Systems

Capabilities

At the beginning of the 2015 design process, the UMd advisor recognized that, over the past several RoboOps competition, there have been more and more high-value targets in the Mars area, as well as a reluctance on the part of all teams to enter this field due to the potential for getting the vehicle stuck. One of the UMd

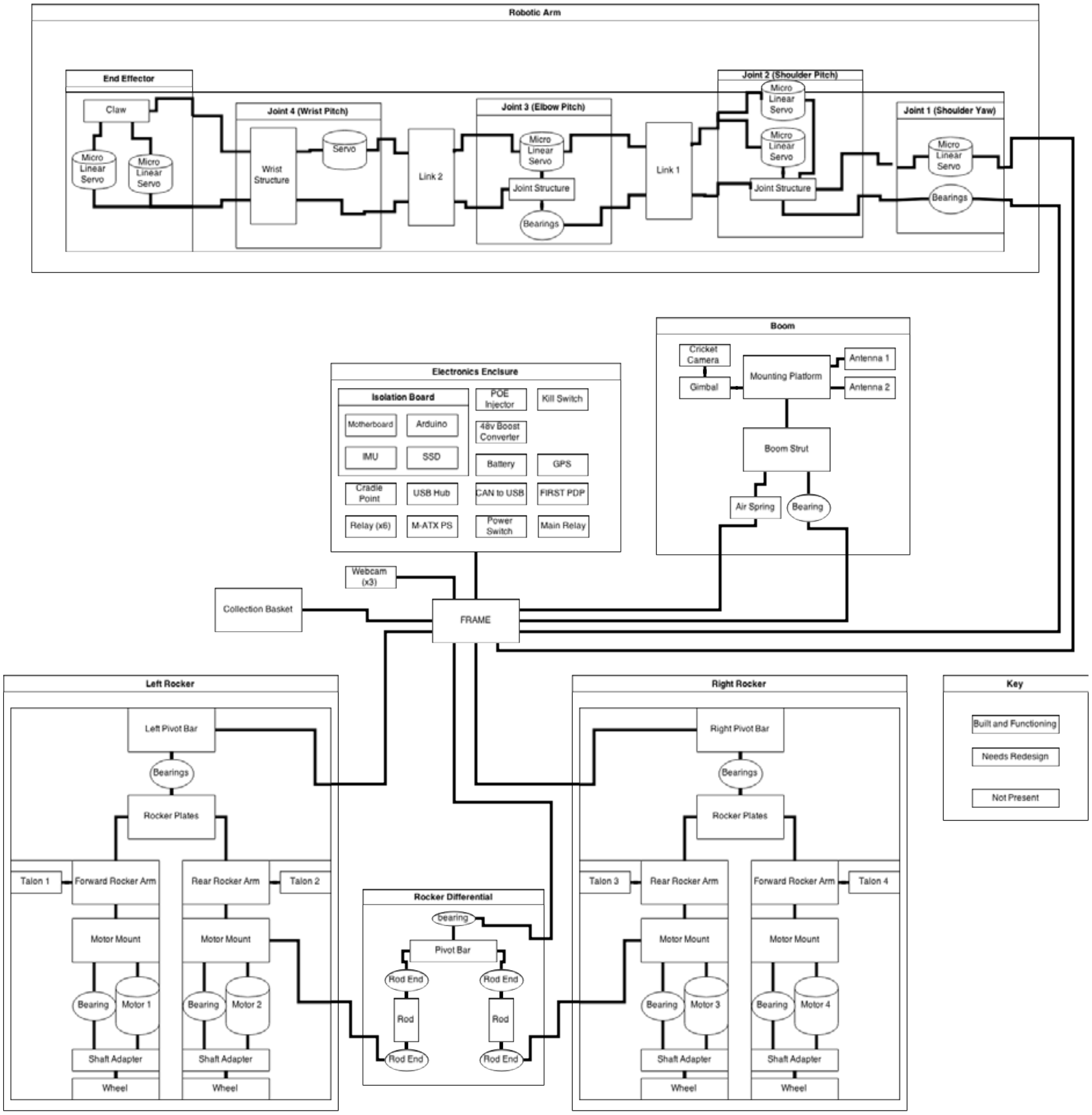


Figure 1: System block diagram for Frigg rover

design goals for 2015 has been to develop a vehicle capable of “running the field” in the Mars area, collecting rocks from even the most congested regions of the field. This led to a set of design requirements to traverse obstacles up to a foot high, travel routinely across loose sand without undue risk of getting bogged down, and climb a 30° slope (to provide margin for the 15°-18° slopes on the navigable portions of Mount Kosmo.)

Wheels

For the first three years, all University of Maryland RoboOps entries used the same set of custom hard-rubber wheels with deep (1 cm) grousers. In 2014, there was a desire to increase the contact patch of the wheels to enable more reliable motion in deep sand. A set of commercial off-theshelf pneumatic wheels were procured, but still had high bearing pressure and sank deeply in loose sand. To increase terrainability, the wheels were doubled; the “dualies” provided good maneuverability, but weighed over 3 kg per dual-wheel assembly. Despite removing more than 6 kg by reducing the size of the battery, the 2014 entry came in at the same weight as in 2013, consigning the UMd entry in 2014 to go in the unenviable starting position.

As part of the “split development” cycle adopted by UMd this year, a team of five freshmen in in ENAE 100 during Fall 2014 conceptualized and developed a lightweight wheel concept, based on the use of a thick cardboard cylinder taken from a concrete casting form. This cylinder, 10” in diameter and 6” wide, was coated with a commercial rubber tread pattern to form the contact surface of the wheel. Plywood disks reinforced the rims and provided the hole pattern for mounting to the drive motor. This prototype was tested in a terramechanics test cell using loose sand as the terrain, and worked well enough to baseline its use for Frigg.

In the Spring of 2015, four of these wheels were fabricated and tested on Persephone, which was kept in service as the system test vehicle while Frigg was under construction. These tests demonstrated that the large wheels worked too well; the high-friction contact surface proved to be almost impossible to reliable skid-steer. A second set of similar wheels was fabricated using 8” diameter cardboard forms 4” in diameter. These wheels worked better, but extended testing showed that the plywood hubs were inadequate to the task of mating to the drive motors. The plywood hubs were replaced by 1/8” aluminum plates, CNC milled for both wheel mounting and with extensive lightening holes.

These wheels were used for extensive testing, including navigability over rocks up to 12” tall, but showed the limitations of the cardboard tread support. Rough terrain maneuvers, such as overtopping rock obstacles, produced large local loads which deformed the tread and threatened to collapse the wheel. While this never happened, it was found to be necessary to further reinforce the tread structure with an outer wheel hub. At this point each wheel was well over a kilogram in weight, and increasing.

An extensive online search revealed a small market for giant (1/5) scale radio control trucks, and a set of commercial wheels were bought and tested. These wheels are 7” in diameter and 4” wide, with an aggressive tread pattern of flexible rubber backed by a sponge rubber interior to alleviate the need for pneumatic inflation. A three-piece hub was designed and 3D printed to mate to the drive shafts of the motors, and the wheels installed on Persephone for testing. These wheels (shown in Figure 2) weigh only 750 gms each, and have excellent traction in both soft sand and gravel.


Figure 2: Frigg final wheels

Chassis

As in past years, the UMd chassis is an assembled structure of aluminum extrusions with sheet-metal gusset plates and shear webs.

To lighten the structure over past years, areas which do not take loads are covered with a thin semi-rigid polycarbonate sheet, which provides environmental protection for dust, rain, and direct sunlight at minimal additional mass. Structural modeling was accomplished this year to verify that extrusions could be reduced in wall thickness to save weight without compromising structural integrity. Rather than fabricate a full bus structure, as in years past, the chassis was designed as a planar structure with a few support structures cantilevered upwards or forwards to support unique hardware such as the manipulator or deployable sensor mast. The 2013 rover demonstrated the fragility of using commercial consumer-grade computer motherboards on "off-road" platforms such as these robots, as three separate systems failed over the course of that year. In 2014 the electronics were built onto a rigid composite board and structurally secured in the vehicle on vibration isolation mounts; in conjunction with the procurement of a hardened motherboard, this eliminated any electronics problems in 2014. The same approach was continued for 2015, other than the selection of different vibration isolation mounts to more accurately fit the vibration spectrum, and to reduce overall volume and weight.

Cameras

Cameras for RoboOps vehicles are tasked for a multitude of requirements. There is a need for high-resolution coverage of the vicinity of the rover to spot potential sampling targets. There is a need for monitoring the field ahead of the rover to assess trafficability and to steer paths over, around, or through potential obstacles. There is also the need to exercise fine control of the sampling manipulator, placing the end effectors on target quickly and efficiently.

To accomplish all of these requirement, the Frigg rover incorporates four cameras, each of which communicates with the router over Ethernet. A Point Grey Cricket camera is used for navigation, mounted on a motion-stabilized gimbal on the sensor mast. This camera is used to find potential targets; the image stabilization provides better views to Mission Control even while the rover is traversing the Rockyard at its maximum speed.

The other three cameras are Logitech LC920/LC930 web cameras, which have been found to provide excellent images even in bright sunlight. These cameras are small enough to be mounted in fixed orientations around the rover, and are nominally used for forward and aft hazcams (observing the ground the rover is rolling over) and a camera on the manipulator lower arm segment, imaging the end effector to improve sampling effectiveness.

Manipulator

The rover is required to pick up irregularly shaped rocks and other targets with diameters ranging from 2 to 8 cm and masses ranging from 20 to 150 gm, which can be partly buried in the terrain or placed on top of other rock formations. The baseline for this year was to adapt and reuse the dexterous manipulator used in 2014. Since there were multiple issues and concerns about that design, a parallel effort was undertaken to design an entirely new dexterous manipulator using linear, rather than rotary, actuators.

The baseline design was a 4-DOF arm consisting of a two-link manipulator with yaw-pitch-pitch kinematics. The arm's motion is controlled by six Dynamixel servomechanisms laid-out as follows: one MX-64 servo controlling shoulder-yaw, two MX-106 servos ganged together controlling shoulder-pitch, one MX-106 servo controlling elbow-pitch, one MX-64 servo controlling wrist pitch, and one MX-28 servo opening and closing the end-effector. The Dynamixel MX-series servos were chosen by the UMD RoboOps team for 2014 as an advance over more traditional servo systems, due to their ability to use a modern digital communications protocol in a single-string "daisy chain" to control all of the servos on the arm with a single channel. Also, the MX- series incorporated an angular position sensor, which for a direct-drive system such as the architecture chosen by UMd provides sufficient precision and repeatability to allow resolved-rate control and autonomous motions to preselected way points.

The manipulator was originally equipped with an interlacing-finger end effector designed by freshmen in a ENAE 100 project team. This was the version shown in the mid-term report videos, which worked well for grasping and controlling payloads. After that point, it was decided to develop an alternative end effector, based solely on a single servo moving half of a two-finger gripping end effector. This was lighter in weight and shorter in distance from the wrist to the grasp point, which increases the dexterity of the manipulator.

During the 2014 development, one of the Dynamixel servos died and had to be repairs at the internal board level. It was found that the servos, despite their extreme price (almost $4000 in 2014 for the entire arm) were unable to pass full current for a serial manipulator through the daisy-chain, despite the manufacturer's claims. The servos also "safed" following an over torque condition in such as way as to require a hard reset of vehicle power, which was the proximal cause of the Persephone arm failure during the 2014 RoboOps competition run.

During testing, the MX-106 in the elbow of the manipulator failed. During diagnosis and repair, it was found that the entire internal control board had fried, making it infeasible to repair at the SSL. The MX-64 was moved from the wrist to the elbow, and the only spare Dynamixel in the lab (an MX-28) was inserted into the wrist pitch position. (This was the original design configuration for the arm in 2014, but was modified when found to be marginal on joint torque.) In subsequent testing, one of the paired MX-106s in the shoulder failed. The vendor was out of stock of the MX-106, and with two of them failed, the baseline manipulator was dropped from Frigg.

Due to concern over the robustness of the Dynamixel servos, in parallel an alternative manipulator concept was designed and prototyped. This arm (Figure 3) was based entirely on the use of Firgelli L-16 series miniature linear actuators, based initially on the success of similar designs with linear actuators by other teams in the 2014 RoboOps competition. The approach to this manipulator was to keep the system as lightweight as possible while replicating the work envelope of the baseline manipulator, and to maximize the use of 3D printed parts

Figure 3: Frigg sampling manipulator

as much as possible for considerations of weight, cost, and rapid modifications during development. This manipulator utilized the same Y-P-P-P 4-DOF kinematics as the baseline manipulator, but all joints were driven via three-bar linkages by linear actuators, except for the elbow, which adopted a five-bar linkage to allow a single actuator to provide nearly 180° of motion for the elbow joint. Except for two 16.5” 0.75in square extrusions forming the two arm links, all other structure for the arm is PLA 3D printed in a fused deposition printer. This includes the bilateral end effector, driven by dual L-12 micro linear actuators.

Control and Communications System

The electronics system block diagram is shown in Figure 4. The heart of any electronic system is its source of energy; in the case of Frigg, the power source consists of a 12 Volt, 20Ah LiFePO4 battery pack. This battery feed into a main system cutoff relay that is triggered by a rocker switch on the vehicle. The main cutoff relay then feeds into a power distribution panel (PDP) that provides circuit protection in the form of snap action circuit breakers for each of the branched circuits. In addition to providing a central point of power distribution for the system, the PDP provides a star ground to help prevent ground loops within the rover. The PDP provides power to the following devices, subsystem cutoff relays, M-ATX DC-DC power supply, POE boost converter, and regulated 5 Volts for the radio modem. The subsystem cutoff relays act as power path interrupters for the four motor controllers, the robot arm power, and the camera gimbal power. A mushroom head switch acts as the emergency power cutoff for the rover and kills power to all of the subsystems that pass through the subsystem cutoff relays. The M-ATX DC-DC power supply sources power for the quad-core Intel core i5, main computer of the system. Finally, the power-over-Ethernet (POE) boost converter provides power for the main mast camera.

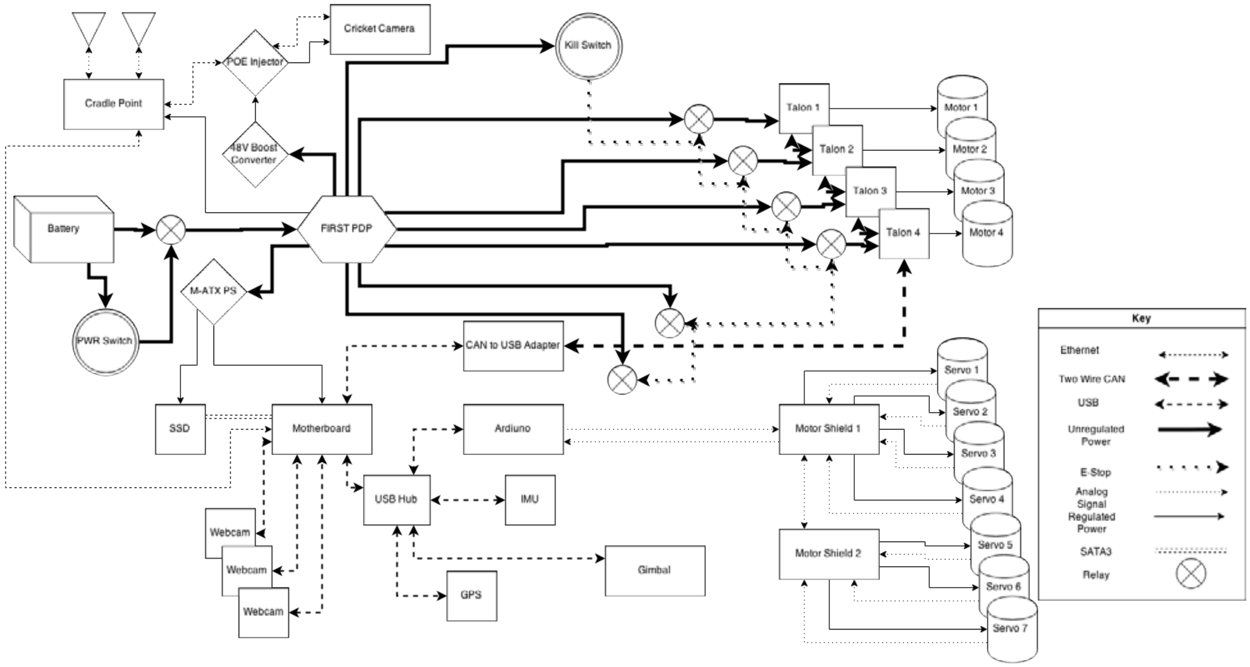


Figure 4: Control Architecture system block diagram

Power is important, but a robotic system will never be useful without data to act upon. As such, Frigg has a Cradlepoint IBR600 that it uses to communicate with the outside world over the T-Mobile LTE wireless wide-area network. Internal actions for the system originate and then radiate from the main computer in the system. The computer communicates to the motor controllers via a CAN-to-USB converter. Additionally, the main computer communicates via USB to most of the systems cameras, the Inertia Measurement Unit (IMU), Global Positioning System (GPS) receiver, mast camera gimbal, and the Arduino microcontroller that drives the robotic manipulator. Lastly, the mast camera communicates via Ethernet to the main computer as well as the radio modem.

Sensors

Over and above the cameras, several other sensors have been installed on Frigg to enhance situational awareness and simplify navigation tasks. A Fidget 1044 inertial measurement unit (IMU) provides three axes each of angular velocity, linear acceleration, and magnetometer readings. During the competition, this sensor will primarily be used to provide current heading of the vehicle, as well as base unit pitch and roll, which indicates local terrain slopes. A µBlox EVK-7P GPS unit provides sub-meter resolution on vehicle position. The current plan is, on the practice day Tuesday, drive the rover (or hand-transport the sensor) all around the Rockyard to get position measurements of critical terrain features, such as craters, rock formations, and the boundaries of "no-go" regions such as the sides of Mount Kosmo. During the competition run, the µBlox GPS will provide a moving map indicator to the driver (Figure 5) as to current and past positions, which will be coordinated with the item locations identified from previous teams to help optimize the collection strategy in real time.

Software

The software system block diagram is shown in Figure 5. It was decided early on that the robot internal communications would be done using ROS. For this to be possible, there needs to be some interface between the Mission Control computer and the Rover's ROS instance. Last year's code used RosBridge for that function; after experimenting with a system using NodeJS that incorporated RosBridge's provided Javascript utility, the decision was made to use RosBridge's WebSocket server to receive inputs and turn them into ROS commands.

To combat the latency that was experienced last year, it was decided that all non-critical networking operations would be performed using the UDP transfer protocol. This would include the drive system calls, because if the joystick is operating at a decent refresh rate (e.g., 20 Hz), then a few missed commands

Figure 5: GPS map indicator for drivers

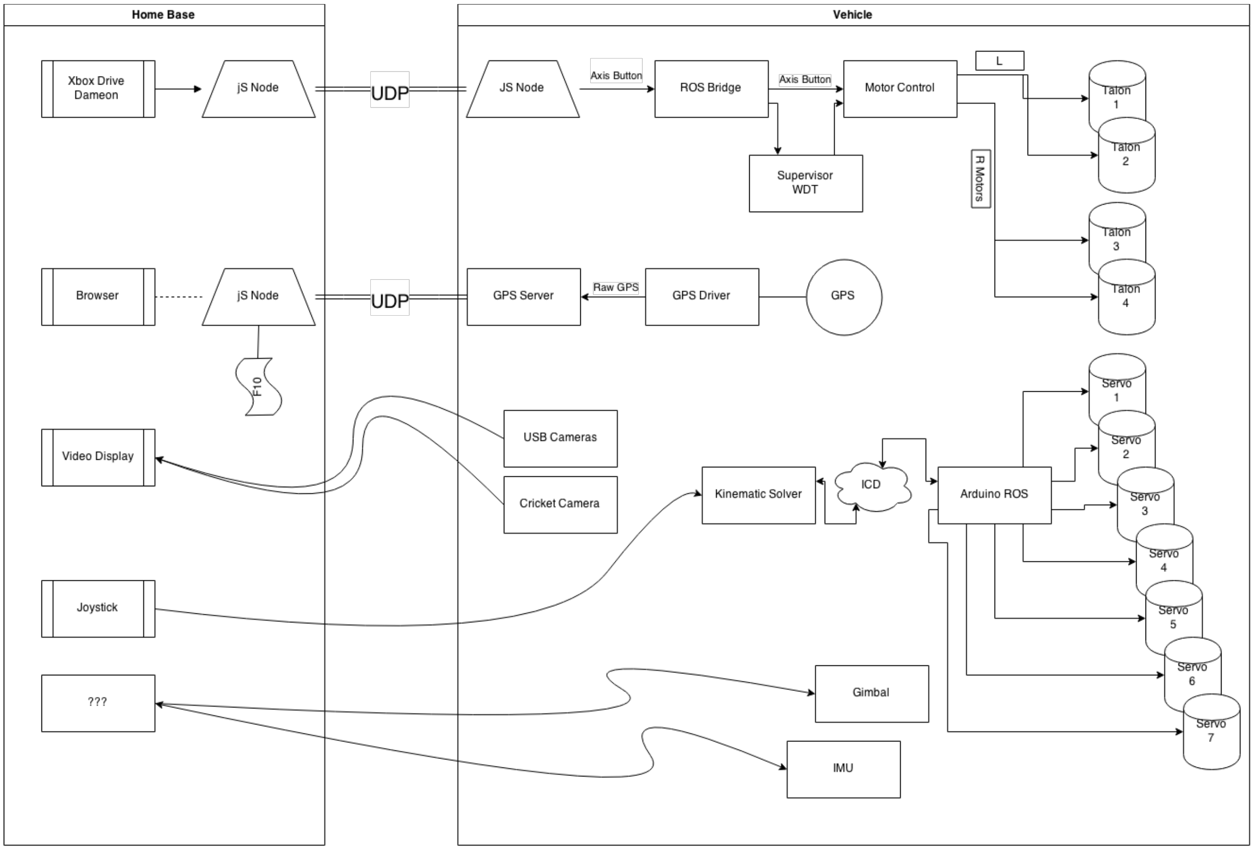


Figure 6: Software system block diagram

would not be disastrous. A NodeJS application was developed to run on the rover that would listen on a specific port for UDP messages and convert them into ROS sensor_messages/Joy messages to submit to the RosBridge Websocket Server for publication. This application is capable of receiving requests from multiple command types, so it can be repurposed for arm control as well as drive control; it just passes the data to RosBridge.

Mission Control's software was required to read values from the joystick and pass it to the rover. This is done using the `xboxdrv` driver and another NodeJS application. The initial attempt was to do it entirely with `xboxdrv`, but this proved to be infeasible. Another approach to perform this function in a GUI through Unity was too unwieldy and added unnecessary complexity.

The rover side of this software is entirely ROS nodes; the client is a browser-based user interface served by a NodeJS webserver. A c++ tool was written using a 3rd party library to communicate with the GPS unit over a serial port. This tool listens for any $GPGLL messages from the receiver which contain latitude and longitude information and echoes them to a ROS topic. A python ROS node subscribes to that topic, takes the messages, parses out the latitude and longitude, prepares them into a JSON object, and transfers them over UDP to a port on the mission control computer.

On the mission control computer, the node server serves a webpage to a local browser and opens a websocket to it. Then it listens on the UDP port, and passes the latitude/longitude data to the webpage. The webpage javascript then uses the GoogleMaps API to display a map and drop markers describing the location of the rover. A system was also implemented for tracking rocks with different colored markers. These are stored on a local file on the mission control computer, so it is available between browser sessions.

Rover audio will be streamed from one of the Logitech webcams on the rover. This uses the `ffmpeg` software; this has been installed on both the mission control computer and the robot. It also uses the UDP protocol.

Technical Specifications

Rated Payload

Payload is limited to samples collected; the collection basket is sized to accommodate all of the rocks and bonus items on the field during the competition.

Max Speed

Measured at 1.1 m/sec on level ground

Maximum Obstacle Size

Routinely climbs over 8-10 inch obstacles without problem

Operating Time

Demonstrated 90 minutes of pure driving (conservative, as compared to actual competition with driving/sampling alternation.)

Drive Power

Can deliver 40 A to each drive motor; typical drive operation is 6-8 A at 12VDC each motor

Battery

12V DC LiFePO4 battery

On-board Computer System

Quad-core Intel i5 hardened motherboard in mATX form factor

Control and Communication Systems

Cradlepoint IBR600 wireless router, communicating over the T-Mobile LTE network (unlimited data plan)

Latency

Measured at 40 msec round-trip (on campus)

Testing Strategy (incl. different terrains)

To the greatest extent possible, this year the team has focused on extensive piece- and system-level testing prior to vehicle integration. The Persephone rover from 2014 has been kept operational, and critical components (such as cameras, sensors, and wheels) were testing in the field on that vehicle while Frigg was under construction. A separate development system has been set up, running all of the vehicle "flight" code and incorporating all of the development tools, which can be used in parallel with the rover for software development and testing.

The RoboOps team modified the Space Systems Laboratory Planetary Surface Simulation Facility (or "Moonyard") to have sandy and gravel areas approximating the sand and lunar areas of the JSC Rockyard; there are short slopes and various-sized large rocks (Figure 7) to provide small-scale simulations of the Mars and Mount Kosmo areas. This allows joint integrated simulations with Mission Control active in the Neutral Buoyancy Research Facility, and the away team with the robot outside. The SSL terramechanics test fixture (Figure 8) is a 12ft x 8 ft box of dry fine sand, which represents a worst case for sandy terrains. Other areas on campus are also used for testing, which provide extended steep slopes and extensive gravel-covered areas to test driving in those terrains.

Figure 7: Wheel testing in Moonyard

Figure 8: Traction assessment in sand

Competition Strategy

This discussion will have to remain speculative until the details of the 2015 scoring algorithm is released (typically on the first day of RoboOps); for the purposes of this report, last year's rules will be assumed.

There are a number of "can't afford to miss" bonuses, particularly collecting a rock from each of the four zones and returning to the start point before the end of the competition time. The latter is a massive disincentive to explore the Mars area, as it has the largest chance of trapping the robot and causing them to lose that 20-point bonus. The strategy for 2015 will be to collect any easy-to-see targets at the top of Mount Kosmo, then quickly collect rocks in the sand pit and the lunar area. In past years there have been a number of "easy" targets in the sand pit; the time we spend there adding rocks will be based on our comfort at being able to extensively traverse the sand without significant chance of getting bogged down. There are, in general , insufficient points in the lunar area (with the possible exception of "alien" bonuses, usually at the bottom of craters); we will go after those early if our place in the rotation allows us to collect data on them from previous runs.

The bulk of our collection activities is planned to be in the Mars area, as in recent years there has been a disproportionate allotment of high-value targets trying to “entice” teams into the difficult area. Our approach to this is three-fold: a robot chassis capable of overcoming most obstacles without serious chance of “bottoming out”; a manipulator of extreme reach (comparatively speaking) and dexterity to allow collection of high-value targets from under or on top of boulders, or from regions bounded by “inpenetrable” rocks; and a “go-for-broke” attitude that we can collect far more points in the Mars area than we can playing it safe and heading back to the top of the hill early. Our aspirational goal for this year is to clean out the Mars area, and get everything else we can along the way.

Mission Control Center Operational Plan

Staffing

This is honestly going to be a challenge, as a change in UMd Aerospace Engineering electives requirements dramatically reduced the number of students taking ENAE 488R this year. Between people starting internships or jobs and the three team members heading to Houston, we hope to have four people in Mission Control. We are richly endowed with graduate student advisors to the team from the Space Systems Laboratory who will be available as off-line advisors, but our policy is that only ENAE 488R students will be on the controls or making tactical and strategic decisions.

Practicing

The switchover to the Cradlepoint router and T-Mobile wireless network means that (for the first time) we have full wireless coverage on the University of Maryland campus. The entire week of May 25-29 will be dedicated to joint integrated simulations of activities at Houston, with the “away team” at the UMd Moonyard with the rover, and the entire Mission Control staff directing Frigg over the T-Mobile network performing timed exploration and sample collection activities.

Decision Making Strategy

Operations are based on a triumvirate; a vehicle driver, a manipulator operator, and a camera operator controlling the mast cam pan-tilt unit and looking for the next target. Any additional personnel are tasked to help watch video feeds for targets, and provide “book-keeping” for real-time tracking of point count vs. time. The major change for this year is to incorporate two new roles of “Tactical Commander” and “Strategic Commander” (which may be combined into one person if we don’t have enough people available.) The “TACCOM” will direct the three vehicle operators in choosing the next target, deciding on how to best approach obstacles such as craters or rock formations, and so forth. The “STRATCOM” will keep track of time and point counts, monitor strategies (such as maximum allotted time in each area), and perform optimizations such as whether it is better to go for more collection targets or bail out and get bonuses. (This strategy is, of course, dependent on the vehicle operators being willing to take directions; past experience would indicate this is about like herding cats.)

Plan for Contingency/Redundancies

One of the major roles for STRATCOM is to decide when to call a “Mulligan” and ask for on-site intervention. Since this is a one-time opportunity, there is a real trade-off between using it early to optimize capability over as much of the run as possible, vs. saving it for a "disaster" case to save the entire run, such as (seen in past years) a wheel falling off. This is about the only contingency call Mission Control can make; we hope to create a structure to use it wisely.

Budget

The top-level budget for the University of Maryland 2015 RoboOps development and competition is shown in the following list. Recent budget problems at UMd required us to live within the $10K competition budget supplied by NIA. Since several major systems (e.g., electronics, drive motors) were scavenged from Persephone or previous RoboOps entries for Frigg, we were able to adhere to the basic budget.

Structure	$1233
Wheels/suspension	$528
Mast/cameras	$599
Manipulators	$1185
Electronics	$1293
Travel allocations	$5195
Totals	$10,035

Public/Stakeholder Engagement

The 2015 University of Maryland RoboOps team has been documenting their development and testing process, and publishing details such as photos, videos, and narratives on social media sites such as Facebook. In addition, the team performed extensive community outreach via face-to-face demonstrations and interactive control opportunities with the robot. In early April, the University of Maryland hosted the Chesapeake Regional FIRST Robotics Challenge competition, with 55 FIRST teams competing with their robots. During the three days of that competition, Dr. Akin (himself a FIRST mentor to team 2537) showed a number of FIRST high school students through the Space Systems Laboratory, and highlighted the RoboOps vehicles as an example of more complex robot challenges that could be available to them in college. In addition, Maryland Day (April 25 this year) is an annual campus-wide open-house in which 60-70,000 visitors come to the University of Maryland and enjoy interactive displays and demonstrations. This year, the RoboOps team used the six-hour Maryland Day for endurance and toughness testing of the rover systems, as the prototype rover was operated all day by visitors (most of whom seemed to be grade school) driving the robot over a series of large rocks (and more than occasionally the feet of team members.)