The Development of a Robotic Platform for Deploying Science Instruments in Unstructured Terrain

by

Nathanael Amos Miller
B.S. August 2007, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTERS OF SCIENCE

ELECTRICAL AND COMPUTER ENGINERING

OLD DOMIIOUN UNIVERSITY
DECEMBER 2009

Approved by:

_____
Linda L. Vahala (Director)

_____
K. Vijayan Asari (Member)

_____
Oscar R. Gonzalez (Member)

ABSTRACT

The Development of a Robotic Platform for Deploying Science Instruments in Unstructured Terrain

Nathanael Amos Miller
Old Dominion University, 2009
Director: Dr. Linda L. Vahala

Robotic systems have long been a standard in the exploration of remote and distant environments. The obvious implications of having a system capable of withstanding extraordinarily hostile environments, while still effectively serving as an emissary to project a virtual human presence to otherwise unthinkable scenes, has continued to fuel the development and deployment of disparate systems ranging from fancy puppets to fully autonomous exploration machines. As the range of application for robotic exploration systems expands, so does the number of challenges regarding the methods and mechanisms that might best serve a given exploration goal. The question of system implementation is closely paralleled, if not overshadowed, by the drive to develop new science instruments capable of making new measurements in hitherto unreachable environments.

This thesis presents a system for demonstrating new science instruments, while simultaneously supporting the study of methods and mechanisms pertaining to robotic exploration. In particular, the robotic system has been architected to

support studies in remote presence, i.e., giving an operator a sense of presence at a distant location in a remote environment. The discussion gives an overview of the development of the specification of a suitable sensor-suit/processing-architecture for the target application. This is followed by a description of the design of a plain-tire all-wheel-drive, all-wheel-steer omnidirectional rolling chassis developed to satisfy the requirements of the sensor-suit/architecture. The implied electronic system for the robot is then discussed as context for a detailed description of the control system for the rolling chassis, which includes the implementation of a novel, geometric state controller. The performance of the chassis and control system is then reviewed as results from the study and current/future work are outlined.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

TRL	Technology Readiness Level
PSI or Ψ	Platform for Science Instruments
NASA	National Aeronautics and Space Administration
IP	Internet Protocol
AWDAWS	All-Wheel-Drive All-Wheel-Steer
4WD4WS	Four-Wheel-Drive Four-Wheel-Steer
JTARS	Joint Technical Architecture for Robotic Systems
CLARAty	Coupled-Layer Architecture for Robotic Autonomy
MER	Mars Exploration Rovers
MSL	Mars Science Lab
CG	Center of Gravity
OCU	Operator Control Unit
PC	Personal Computer
cRIO	Compact Rio
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
IC	Instant Center
DoF	Degrees of Freedom
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
PD	Proportional Derivative
PID	Proportional Integral Derivative
PWM	Pulse Width Modulation
OS	Operating System
CCD	Charge-Coupled Device
G	Gravity
COTS	Commercial Off-The-Shelf
CNC	Computer Numerically Controlled
EDM	Electron Discharge Machine
LAN	Local Area Network
WLAN	Wireless Local Area Network
RT	Real Time
LOS	Loss-Of-Control
VISA	Virtual Instrument Software Architecture
RFLMS	Raman Florescence and Lidar Multi-sensor
XRF	X-Ray Florescence
IVHE	Immersive Virtual Human Environments

1. INTRODUCTION

The pathway to making an extraterrestrial observation with a new science instrument is extremely challenging. The high-level technical maturity required to execute even the most basic experiment poses a daunting obstacle to the development of would-be space-based instruments. The difficulty in achieving a high technology readiness level (TRL) is exacerbated by the typical ambient scarcity of funding, and the inherent uncertainty associated with a developmental scientific instrument. Thus, as a means of mitigating the risk posed to a perspective sponsor, a strong case exists for low-cost advancement in TRL.

The successful implementation of instruments for extra-terrestrial observations almost always requires some minimal level of robotic competency. Historically, the demands placed on the robotic platforms have increased at a rate that is at least comparable to the advancements of the planetary measurements in both complexity and precision. The various technologies required for the ever-growing demands of robotic exploration have been met with a wide variety of robotic test platforms. These test beds have been used for developments in mobility, sensor development, and control algorithm testing, and continue to play an ever-increasing role in our understanding and ability to build and deploy autonomous systems.

The interest in autonomy results, in part, from a desire to expand the facility of current exploration robots, with particular attention toward Martian- and lunar-

surface missions. There are approximately 56 million square miles on the surface of Mars, as compared to the 57.5 millions square miles of Earth's landmass. [1] It is reasonable to conclude that a meaningful, large-scale, surface-based exploration implies an increase in the number of mobile systems, as well as a decreased dependence on ground-based support for basic operations. Thus, a motivation to study low-cost, autonomous exploration robots clearly exists.

The ultimate vision for ground-based planetary exploration should be to have a number of autonomous mobile systems carrying out a coordinated exploration campaign. Each system would be tasked with reporting its findings based on its level of scientific interest. While these systems would still operate under the supervision of a ground-based controller, they would receive their primary guidance from their own science and navigational sensors.

1.1 Problem Statement

This study presents the design and implementation of a robust, outdoor, mobile platform that is named the Platform for Science Instruments (PSI or Ψ). The combined objective of this vehicle is to contribute to the vision of autonomous planetary exploration, by fielding science-instrument demonstrations, and conducting robotic-algorithm and sensor-integration experiments in natural environments, at a low monetary cost. The mobile platform is capable of carrying an array of robotic sensors and a science instrument into diverse and rugged

terrains with enough computing flexibility to be reconfigured for various sensor and algorithm experiments.

The design philosophy used, considers the mechanical aspects to be of lesser importance in comparison to the software, processor, and sensor architectures. The objective of the platform is to demonstrate instruments and experiment with robot sensor systems. This being the case, a ‘just-enough-design’ approach was taken in the development of mechanical systems. This led to a more robust, and less optimized, vehicle derived from good design sense, best practices, and concise use of analytical tools.

The greater emphasis in the design of the system is given to a rapid test philosophy in which care is taken to minimize the effort required to execute a design/build/test cycle. The operating principle of this scheme is that there is a proportional correlation between the number of experiments executed, and the quality of the resulting product, be it research or some other tangible. The objective is then to ensure that the effort associated with getting from the initial idea for the experiment to a test site, is kept to an absolute minimum.

1.2 Agency Setting

The system described in this thesis was developed in the Electronic Systems Branch of the Systems Engineering Directorate at NASA Langley Research Center in Hampton, Virginia. The activity was initially supported through a number of awards that were designed to foster creativity and innovation within

NASA. These awards were later complemented by various branch and directorate funds. These funds were used to cover the necessary materials, hardware, instrument, and fabrication costs. The bulk of the engineering time was paid for by various student programs.

Early discussions regarding the requirements of the perspective system were initiated late in the summer of 2006. During the fall and early spring of the following year, a number of concept designs were developed to evaluate different mobility schemes. These studies were complemented by a survey to determine and discover the state of the art, and how to best address issues and curiosities identified in previous work. Late in the spring of 2007, design of the mechanical system began. The detail design and fabrication of the components of the electrical system was carried out in part by ‘the first group’ of summer students. By the end of the summer, the power system and a skeleton of the sensor and computing architectures were demonstrated, along with the assembly of the mechanical chassis.

The late spring of 2008 saw an integration of the mechanical and electrical components, which included actuator control of the drive system. Initial field tests and demonstrations were conducted late that summer as ‘the second group’ of summer students participated in bringing the system to a crude, but operational level of functionality. Through the end of 2008, and the spring of 2009, an attempt was made to derive a completely general omnidirectional controller based on the method of instant centers described below.

During the following summer, the control system, herein presented, was implemented, and ‘the third group’ of summer students contributed to bringing the system to a higher level of functionality. The late summer and early fall of 2009, saw the close of the integration of the sensor system with the robot platform, and the final user interface. Also during this time, an effort to mount a laser-based instrument on the robot was initiated, with plans to demonstrate the device later this winter.

1.3 Statement of Uniqueness

This thesis presents a unique, low-cost approach to combining science instrument demonstrations with robotics research. This system is the first known robot that used National Instruments’ LabVIEW as the primary programming environment to implement the system in its entirety. It is also the only known system to use LabVIEW, together with Microsoft Windows XP, as the processing base and internet protocol (IP) connectivity to provide a superb reconfigurability for robotic experimentation. Finally, no prior art utilizes the principles of behavioral robotics, or a general geometry-based construct to control an all-wheel-drive, all-wheel-steer (AWDAWS) mobile platform.

1.4 Thesis Outline

The first section of this paper is the BACKGROUND, which introduces robotics, the experiences that led to the development of presented robot, an outline of the

general tenants of behavior-based robotics, a discussion of holonomic versus omnidirectional motion, and finally a bit of context regarding the control of AWDAWS mobile systems. A discussion explaining the top-down design from objective to specification is given in SYSTEM COMPONENTS, followed by a bottom-up description of the as-built system in IMPLEMENTATION. In RESULTS AND SYSTEM PERFORMANCE, measures of system effectiveness are presented to characterize the system's computing architecture, the performance of the mechanical system, the robustness of the control system, and the facility of the remote presence. As this system has been developed to support other research activities, FUTURE WORK briefly mentions follow-on efforts. CONCLUSIONS summarizes the project, and rounds out the thesis proper. Finally, the REFERENCES provide documentation on the foundation of this study.

2. BACKGROUND

The purpose of this section is to chart the collage of ideas from which this work has been drawn. The cited works constitute the core of the body of knowledge that directly and contextually formed the background for this study.

1.5 Previous Experience

To understand the context in which this study was carried out, it is useful to summarize the experiences of the Robotics and Intelligent Machines Lab, as they have led to an individualized view in three specific areas of robotics: computing and processing architectures, sensor-array design and use for remote presence, and control of omnidirectional mobility platforms.

The labs interest in omnidirectional mobility and robotic platforms started with *Mobius*. [2] Mobius, shown in figure 1, was an experimental system that relied on a fuzzy-logic controller to achieve holonomic and omnidirectional mobility of a four-wheeled mecanum-type drive-train [3] [4], and explored vision-based remote navigation through a web interface. The results from this experiment were sufficiently interesting and well received, that work was commenced to build *Trinity*; a second experimental platform. [5] Trinity served as a demonstration platform for a new, reconfigurable, scalable computing system [6] that was being developed by an affiliated research group. The platform was simultaneously used to explore aspects of behavior-based robotic control and artificial intelligence,

omnidirectional and holonomic control of a three-wheeled mecanum-type drive train, and experiments involving sensor fusion and navigation. [7]

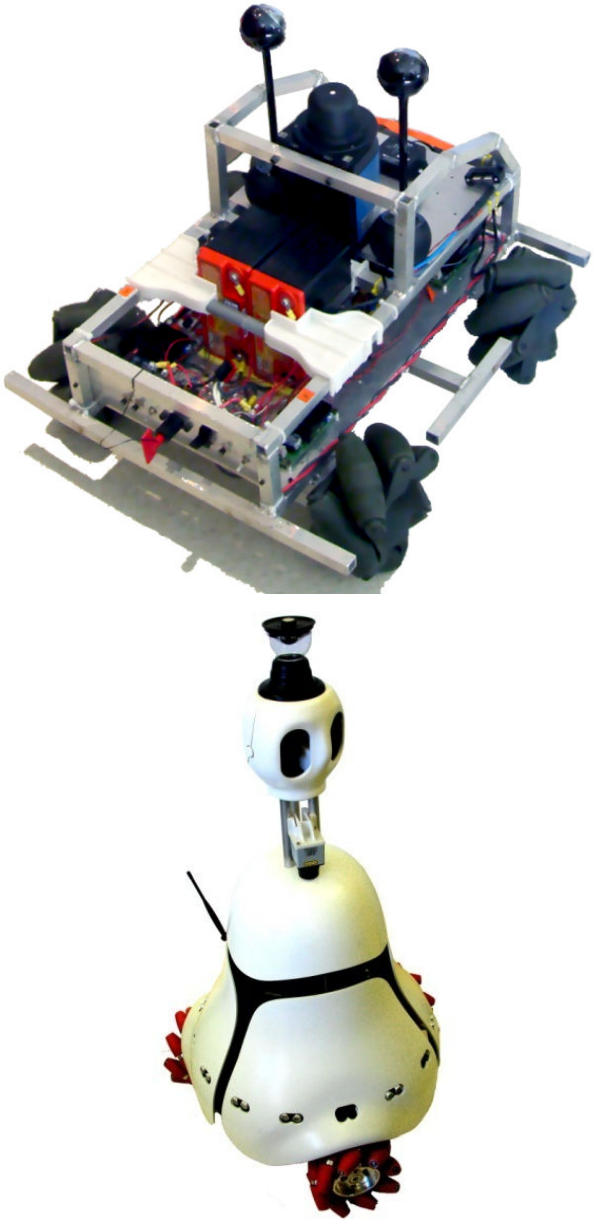

Fig. 1. Mobius (left) and Trinity (right).

In the process of this development, an acute awareness was developed of the criticality of the software and computing architectures. This awareness was

bolstered by participation in a study aimed at defining standards for large systems of exploration robots, which was being carried out under the Joint Technical Architecture for Robotic Systems (JTARS) initiative. [8] These studies were similar in objective, but broader in scope, to the concurrent development of the CLARAty (Coupled-Layer Architecture for Robotic Autonomy) standards now being used elsewhere in NASA. The primary objective of CLARAty is to provide a common framework across heterogeneous platforms, and consolidate disparate activities toward development of rover-centric technologies. [9] [10] While JTARS included a technology development component, the effort was geared toward interagency standardization for purposes of interoperability and intercommunication between various robotic exploration agents.

Trinity proved to be productive as a robotic research platform. The combination of the sensor suit and holonomic mobility base allowed a number of studies to be carried out. These studies ranged from investigating into human/robotic interactions [11], to the use of crude radial sensor arrays to execute behavior-based navigation[12], to the use of color spectrum analysis for map-less navigation [13] [14], to the combination of visual inputs for improved teleoperation. [15]

Trinity was found wanting in that it was primarily an indoor vehicle, which limited the range and class of possible demonstrations and experiments. It also lacked both the physical and computational modularity required for quick and effective experimentation. It was thus determine that such experimental systems require

software and computing architectures that give extreme emphasis toward flexibility, unlike the well-defined and optimized architectures used in actual missions. [16] [17] The result of these developments was a resolve to collect the lessons learned, and build a larger, more flexible, more capable outdoor robot that could take both robotic experiments and scientific demonstrations to a wider range of test sites.

1.6 General Robotics

For cultural, semantic, and philosophical reasons, the term *robot* defies concise definition, and exhibits disharmony among various dictionary definitions. This study claims a membership in the field of robotics on the basis that *robot* can be defined as follows: A robot can be any system that possesses a unique identity and three of the four characteristic elements, which include some physical embodiment, sensors through which it interacts with the world, a dependence on an algorithmic functionality, and a processing capability. The adoption of this definition is intended to emphasize the inherent specialization of robotic systems. This study is further confined to mobile ground-based robots whose design intent is to support scientific development, research, and discovery.

Most known cases of ground-based science supporting robots, with the prominent exception of agricultural systems, are oriented toward extraterrestrial planetary exploration. In July of 1997, the first successful planetary rover, Pathfinder, drove on the surface of Mars, ushering in a new era of planetary

exploration. [18] This mission demonstrated the value, but more importantly the viability, of a mobile rover capable of surveying a remote environment and performing scientific measurements. The success stimulated an effort resulting in a second rover mission that delivery the two astoundingly successful Mars Exploration Rovers (MER), Spirit and Opportunity [19] [20], to the Martian surface in January of 2004. The findings of the MER missions have proven fundamental to our understanding of Mars and other planetary bodies, including Earth. With promise of further discoveries a third mission, Mars Science Lab (MSL), began its initial definition phase in 2001. MSL is intended to expand on the discoveries of MER with a more sophisticated and capable set of science instruments, and is currently under development for the 2011 launch opportunity. [20]

Their demonstrated success has guaranteed this class of mobile science emissaries a prominent position in future planetary exploration endeavors. This occasion has given rise to a number of serious platform development projects aimed at advancing various instruments and robotic deployment technologies. These efforts have a similar role to the technology development effort that preceded the building of the Pathfinder with the Rocky rovers; in particular Rocky 7 [21].

An exceedingly wide variety of locomotion mechanisms have been proposed and implemented. However, the majority of the platforms being developed for planetary exploration use wheeled appendages that are fitted to the rover chassis with either actuated or kinematically-linked suspension mechanism**s**. A

summary of drive systems in use was published by Bartlett, Wettergreen, and Whittaker that gives a brief but thorough vehicle taxonomy for mobile robots. [22] The tendency has been for smaller science-dedicated robots to hold to the kinematic linkages, such as the rocker/bogie system proven on Pathfinder and MER, and to be used on MSL. [23] [24] [25] [26] In some cases, hybrid systems are being pursued for specialty tasks such as crater crawling, [22] but in many cases, active suspension is relegated to the larger robotic platforms, which are being developed for site-preparation activities for manned exploration. [27] [28] [29] Preparation for robotic missions, and validation of various technologies, both robotic and scientific, are carried out frequently at planetary analog sites around the world.

1.7 Behavioral Robotics

The term behavior-based or behavioral robotics is defined in contrast to cognitive robotics, and is driven by the notion that signal mapping, as witnessed by biological systems, can meet or exceed the capabilities of modeling and analysis in the area of autonomy and robotic control. [30] References to behavior-based robotics are often made in the context of control schemes for mobile systems that rely on high bandwidth channels from sensors to actuators. Functionality of such control systems is derived from a layered and asynchronous network of modular ‘behaviors’ that interact over low-bandwidth communication channels. Levels are organized by function, with more complex tasks executed at higher levels, and

base functions executed at lower levels. The ability for a higher level to ‘subsume’ the role of a lower level is characteristic of behavioral algorithms, such that a base function executes a certain task unless and until a higher-level behavior intervenes to augment or suppress the function of the lower behavior. [31] Such systems are frequently referred to as finite state machines. [32]

An individual behavior is simply a function that varies with, or is toggled by, the output of some other function in the system. A given behavior will execute a set of tasks in a particular manner, until modified by an inhibition or an assertion stimulus from a different part of the system. The stimuli originate from either a sensor or another behavior. The underlying premise is that, with intuition and ingenuity of the designer, individual behaviors can be easily assembled into a behavioral network (a.k.a. finite state-machine) that exhibit the appearance of intelligence. The ‘intelligence,’ or high-level of capability, is derived from interactions between the elemental behaviors that execute their several functions.

The driving notion behind the development of a finite state-machine is that it should be conceived and constructed in the ‘native language’ of the system it is to control. The control systems are usually established with semantic links that relate the stimulus to action with simple ‘if/then’ and triggering relationships. [33] [34] If required, these semantic models are then migrated to more analytical relationships based on simple heuristic functions that relate actuator response to the stimulating signals. [35] [5]

As the finite state-machines become more complicated, the use of a configuration space is frequently employed. These constructions are devised to aid the designer in understanding the relationships between the different sensory inputs and the actuation outputs. [36] As these configuration spaces become sufficiently complex, notions relating to neural-net training and tuning are invoked to resolve the sensor-to-actuator input/output mappings.

1.8 Holonomy and Omnidirectionality

The single greatest challenge posed to any mobile robot is one of the grace and practicality of its mobility. The absolute epitome of robotic mobility, especially for autonomous robots, is holonomic motion. As defined in standard text books [37]

“A holonomic system is a one that has the same number of independent virtual displacements as the number of generalized coordinates to locate a vehicle uniquely in space. Therefore a holonomic omnidirectional vehicle is a vehicle for which all three independent planar motions, two translational and one rotational, are admissible at a non-zero velocity from an arbitrary configuration.”

Holonomic drive systems hold intense attraction to roboticists because of the implications toward ultimate maneuverability, and ease of path-planning. However, the variety of navigable terrain is typically limited in proportion to the degree of holonomy delivered by a given system. The usual compromise is one of omnidirectionality as a surrogate for holonomic motion. Non-holonomic, but

omnidirectional motion, typically involves some reconfiguration of the wheeled motion base.

A wide variety of approaches have been taken toward achieving truly holonomic motion. A sampling of this diversity finds a number of ball-type drive systems [38] [39] [40] that rely on component actuators to drive a ball, or an array of balls, that are in contact with ground. These typically suffer a disadvantage due to obstacle height restrictions and poor traction on unprepared surfaces. Another class of systems depend on a variety of compound wheels, omni and mecanum [4], that have a series of small wheels peripherally mounted around a larger wheel, allowing free motion or at 45° or transverse to the rolling axis of the wheel. A variety of configurations have been proposed [2] [41] [42] using this class of wheel, but all suffer the inherent complexity of the compound wheels. While rough terrain compound wheels have been proposed, all known solutions of this type are extremely susceptible to debris contamination and fouling.

More traditional omnidirectional solutions tend toward a system of plain wheeled-powered casters that are actively or passively suspended. These systems are generally referred to as AWDAWS vehicles. The variation within this general class can be broken into two broad categories: offset and in-line casters. Schemes for providing holonomic motion using offset casters have been proposed. [43] [44] The difficulty with such systems on soft or uneven surfaces is that the offset, when combined with the angular torques, yield a screw-like effect that confines operation to prepared or stable surfaces. [45]. To avoid

reorientation issues and improve mobility, Udengaard and Iagnemma [46] have proposed pivoting pairs of individually powered wheels on free-turning off-set casters. This approach avoids the auguring affects of other offset casters, but does not deliver holonomic motion.

In-line casters schemes allow the wheels to pivot without shifting the tire contact area, with respect to the vehicle frame. This is done with a zero offset dimension, such that the rolling axis of the wheel intersects the steering axis of the caste. The majority of the omnidirectional rough terrain AWDAWS rovers use the inline caster. [23] [24] [28] [29] Such systems typically have tires with lower surface loadings, allowing routine in-place reorientation of the wheels for changing from one drive state to another. Reductions in auguring and power consumption are frequently achieved by positioning the wheel, such that the contact patch is not centered on the steering axis, to effect a combined steering/rolling motion. [25] [47]

1.9 Control of All Wheel Drive All Wheel Steers

A large body of work has been developed on the control of AWDAWS systems in attempt to capitalize on their potential utility and versatility. A number of studies have been carried out for specialized systems, including freight handling [48] and agricultural systems [49]. A number of these specialized control systems rely on graph transformations from the geometries generated by an instant center of

rotation. The geometry-based studies that were reviewed lacked generality, which is believed to have resulted from their specialization.

A larger and more developed body of work is based around a basic four-wheel AWDAWS system (4WD4WS) with passive differential suspension. Makatchev, et. al., proposed a modern control approach, based off of a linear state model, which captures the dynamics and cross-coupling of the of the drive/steer system. [50] This work was expanded to comprehend extreme out-of-plane wheel displacement with the use of a linear projection to transform an equivalent instant center of rotation. [51] The basic dynamics and cross-coupling models for the differentially suspended 4WD4WS platform have been used in conjunction with extensive studies in teramechanics to model wheel slippage. [24] [52] These models have been verified with simulation and experiments, and have proven reasonably accurate for well-characterized terrains.[53]

Known active areas of research for AWDAWS systems include the study of articulated and active suspension platforms. These systems add CG shifting and walking or crawling to the AWDAWS control problem, which may enable vehicles to navigate steep slopes and extreme terrain, [29] [54] [55] Very little work has been found on addressing issues of optimal or intelligent control of AWDAWS systems for the purpose of resolving the inherent redundancy in drivable states.

3. SYSTEM COMPONENTS

This section provides a system level description of Ψ's main subsystems. This overview frames the relevance of, and provides a basis, for the detailed discussion of the implementations that follow.

1.10 System Architecture

Ψ is designed to be more of an experimental robot than an operational test-bed, so a higher emphasis is given to flexibility and over-robust operation; though both are highly desirous. To this end, the sensing and computing architecture, outlined in figure 2, is a distributed computing network hosted local to the robot. The computing network is comprised of up to four embedded processing nodes, an embedded vehicle controller, and a data connection to the controller of an undefined science instrument. The network also connects to the network enabled sensors, as well as to a high-power, wireless link that serves as the main communication channel for remote operation. The operation of the robot is handled by an Operator Control Unit (OCU), a laptop or desktop PC that runs as client to the robot's embedded host computers.

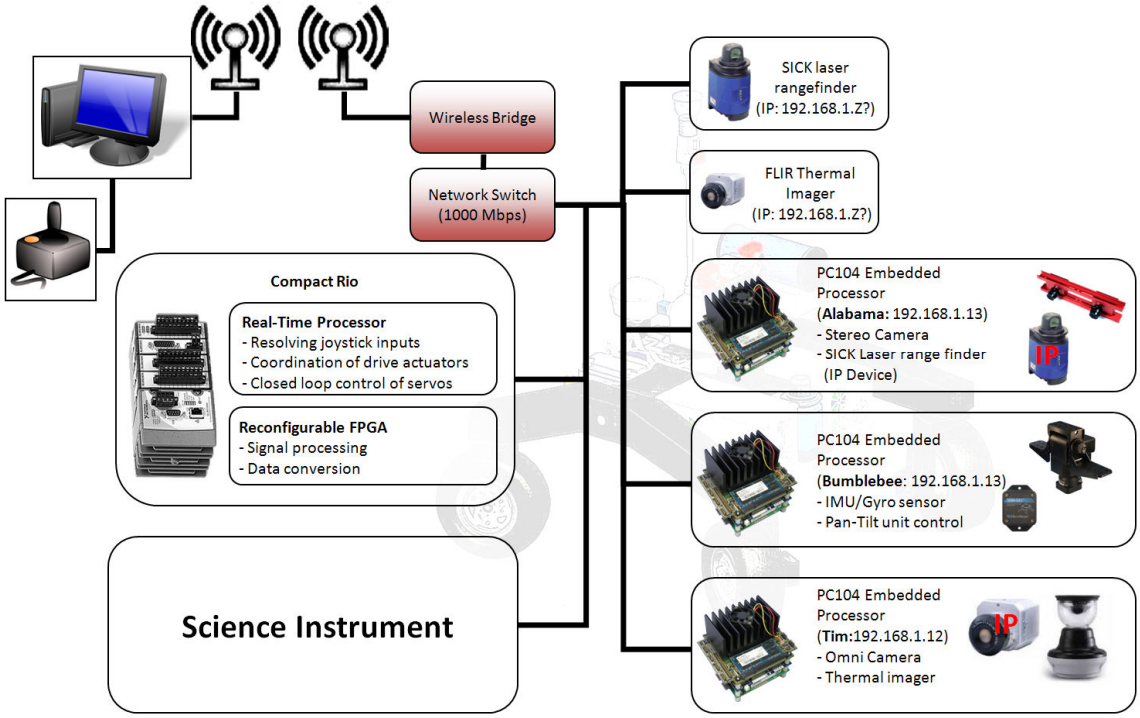

Fig. 2. The IP network and the devices that connect to it.

The onboard processing nodes are PC104 compact stackable units. Each node consists of a main processor board with added modular I/O boards, and is equipped with an installation of Windows XP Embedded. The embedded processors provide the robot-centric computing needed for executing algorithms, processing data from the network enabled sensors, and providing an interface for the non-networked sensors. With the intent of preserving the highest level of flexibility, National Instruments' LabVIEW was selected as the tool to implement the entire software architecture because of its range of functionality and ease of use. LabVIEW is installed on each of the embedded computers, including the robot controller and the client OCU. Device interface applications are developed and run on the embedded computers. These applications interface to the

sensors, process the sensor data, and publish prepared data products to the robot's network for use by other processors or presentation to an operator via the client operator control unit (OCU).

Finally, the control of the robot's mobility base and a directional pan-and-tilt unit – all of the actuated devices implemented on ψ – are handled through the Compact Rio (cRIO): an embedded device purchased from National Instruments. The cRIO has three parts: a modular data acquisition interface, an FPGA signals processing backplane, and a RealTime microcontroller. The cRIO is used to control the actuators, manage routing of power, and monitor the power consumption of the system.

1.11 Sensors

To transfer a true sense of presence from a mobile platform to a remote location, there are three primary needs the sensor suit must address with both quantitative and qualitative measures. Those needs include a general sense of awareness, a means to conduct detail scrutiny of a target of interests, and a way to determine the global location of the vehicle. Qualitative measures are critical for making high-level judgment decisions, and for providing a general sense of connection between the operator and the vehicle's environment. Quantitative measures allow accuracy and precision in a user's assessments by providing critical feedback for hazard avoidance and target acquisition. The objective sensor suit

of Ψ, shown in figure 3, is designed to fill each of these needs with a total of eight sensors.

Fig. 3. Full complement of sensors to ultimately be implemented on PSI.

The need for general or overall situational awareness involves taking wide field sensor scans, which give the user a complete, albeit low quality, indication of the vehicle's immediate surroundings. On Ψ, this is achieved with the use of an omnidirectional camera and a laser range-finder. The omnidirectional camera generates a complete panorama that is wonderful for visually identifying targets, but is almost completely useless for avoiding obstacles. Two different laser range-finders are used to provide quantitative obstacles detection. A long-range laser scans a plane 18 inches above the ground for large obstacle avoidance,

while a short-range laser scans a series of slices in front of the robot so as to detect and avoid navigational hazards.

For detailed target interrogation, a directional pan-and-tilt unit is used to slue a set of instruments to bear on a target of interest. The full instrument compliment consists of a stereo imaging pair and a thermal imager. Detailed and long-range imaging is accomplished by an accompanying ‘targeting camera’ mounted to a variable-zoom optical telescope. The resulting directional sensor-suit is capable of delivering a color image, temperature map, and distance measurement of objects within reasonable eye-shot of the rover. The image from a single camera of the stereo imaging pair, and the image through a telephoto lens, satisfies the qualitative needs of a remote user, while the stereo measurement and thermal mapping provide a good quantitative understanding of the target.

The last test to remote presence is localization. Qualitatively situating a rover in its environment from *in situ* sensor readings is very difficult. Thus, only quantitative measures are provided in the form of an inertial measurement unit (IMU)/gyroscope and a global positioning system (GPS) receiver. It is worth noting that qualitative localization is possible and is an active area of research [14].

1.12 Mobile Platform

1.12.1 Mechanical Requirements

To define the mechanical needs of the mobile platform, a set of requirements are derived from four specific needs: outdoor omnidirectional mobility over rough terrain; sensors required for remote operation; basic power, system control, data processing; and accommodation of a laser based science instrument. These particular needs were developed into a set of requirements that captured the objectives, both the necessary and desired, of mobility system, sensors, on-board processors, and perspective science instrument(s).

Mobility

The mobility of the system is specified to meet certain criteria, the first of which is navigation of rough terrain. The objective is to operate the vehicle without danger of tipping in a typical rural or urban setting, with the possibility of operation in a planetary-analog test site. The conditions taken as design requirements are traversal of uneven terrain with step obstacles of up to 12 inches in height, and inclines of up to 45°. The expected surface materials are loose, damp soil, sand, gravel, and/or Lunar or Martian regolith simulant.

While large obstacles are to be navigated with caution, the robot needs to move fast enough across reasonable terrain to be useful as an emissary for, or an assistant to, a human user/collaborator. Having a slow robot makes human/robot

interaction difficult and onerous. Thus, the design goal is for the robot to have a straight-line, level-ground transit speed of approximately 5 miles per hour.

The carriage of the robot is to have the capability of omnidirectional, or near omnidirectional, motion. Based on previous experience [7], such mobility can significantly increase the degree of effectiveness with which a remote operator can utilize the vehicle. Additionally, omnidirectional vehicles have significant advantages with regard to maneuvering in tight or congested settings [7] [47] as are likely to be experienced in a perspective test site. Finally, in the event that the robot is used for experiments in robotic autonomy, such a mobility base will mitigate the need for advanced algorithms by reducing the complexity of the path-planning problem, and enable the vehicle to more easily navigate difficult terrains.

The need for ease of transportability places a final constraint on the mobility system in the form of a very basic dimensional constraint. As the robot will frequently need to be moved from the lab to and from a test or demonstration site, the vehicle must be able to fit through a standard doorway. Even at this very vague level of definition, it is reasonable to expect that the need for partial disassembly of the platform is likely. The target disassemble time is then set for 30 seconds in which all electrical and mechanical connections must be separated. The target time to assemble – from a transport configuration to fully operable – is one minute. These short times are imposed in agreement with the

rapid test philosophy of the overall design to ensure the physical effort associated with getting to a test site is kept to an absolute minimum.

Sensors

The needs of the sensor suit required for remote operation are very basic, and are driven predominantly by the field-of-view of each device. The requirements are designed to ensure the appropriate instruments are given priority based on their contribution to the sense of presence of a remote operator. This priority corresponds closely to being centrally located on the vehicle, with a clear view in all directions.

The omni-cam provides visual situational awareness for the robot, thus it is important it have visibility of all the area in the direct vicinity of the robot. The laser range-finder is more of a navigational tool as compared to the omnidirectional camera, and is therefore required to be mounted with minimal optical interference, and relatively near the ground with as few occlusions as possible. Because of the 100-meter range near field occlusions result in reasonably large blind-spots that degrade the functionality of the sensor, and presenting a hardship to navigation.

The forward-cam is associated closely with the operation of the pan-and-tilt unit. This directional imaging function is intended to emulate the directional vision innately understood by an operator. Thus, the pan-and-tilt is required to point toward the front of the robot, and provide a broad field-of-view such that there are

no occlusions in the forward direction, and minimum obstacles in all other directions. Additionally, the pan-and-tilt is needed to support a range of remote sensing instruments and cameras and allow them free rotation, such that vision-based navigation is possible.

Computing

A perspective sensor suit of nine sensors, an actuator base of four powered casters and a pan-and-tilt unit, and the emphasis given to the reconfigurability of the processing architecture, leads to great emphasis being placed on accommodation of the computing and electronics. The requirement is then to provide power and an operating environment to the four PC104 configured single-board computers and the one NI Compact Rio controller. This implies a thermal, vibration, and preferably moisture-resistant enclosure where “thermally resistant” may be interpreted as the availability of air for convective cooling. The electronics enclosures must allow the robot to operate in expected on a test site conditions. These conditions include reasonable vibration, a wide temperature range and potentially moisture and/or humidity.

Science Instrument

Finally, the base instrument the robot is to support is to be a compact laser-based sensor whose laser head is directed by the pan-and-tilt unit, and whose electronics/controller is housed in a custom enclosure. In support of the

experiment, the pan-and-tilt should have an unobstructed view from the front of the robot such that the instrument can be easily pointed at objects located in the front hemisphere of the landscape around the robot. The goal is to accommodate the sensors in a way that makes installation and removal from the robot as easy as possible.

1.12.2 Rolling Frame

The rolling frame of ψ has three main features that were selected based on the target use-case of carrying a scientific payload into a test site for demonstration purposes. The anticipation of severe terrains accompanied by a need to remotely operate the vehicle in a transparent manner, drove a number of mechanical requirements pertaining to traction, navigability, instrument and processor packing, and overall dimension of the vehicle.

The most dominating characteristic of the rolling frame is the kinematic suspension system. The suspension scheme was selected for its demonstrated ability to cope with rugged and steep terrains. Though frequently veiled behind different body or wheel configurations, this suspension system has been employed by a number of very successful robotic platforms. [20] [18] [23] The system uses a differential system to distribute the load evenly across all wheels; center-of-gravity consideration aside. Such kinematic systems have an advantage over a comparable spring-suspended chassis in that they can handle

extreme deflections without significant modification to the wheel contact load distribution.

Omnidirectionality is extremely desirable to reduce the significance of path planning in remote or autonomous operation, and to facilitate statically mounted components on the rover. To achieve this goal, a four-wheel, all-drive-all-wheel-steer (AWDAWS) system is used. The drive train consists of four powered casters mounted at the ends of each rocker leg. The wheels are arranged in a square pattern to increase the omnidirectional nature, while reducing the overall directional bias of the vehicle.

The final characteristic of the rolling frame is the chassis, which consists of a multi-box system. Two removable electronics boxes are mounted forward and on either side of the sheet-metal body. Storage compartments for the robot's batteries, sensors, onboard network components and other equipment are positioned aft in the body. A structural sub-frame is fitted inside the body, which transmits the various loads to the suspension differential, and ultimately to the legs. Additionally, the frame provides rigid hard-points for mounting the pan-and-tilt unit to the chassis.

To allow for future upgrades or ruggedization of the vehicle's control and processing systems, the electronics boxes can be easily removed and replaced. Additionally, the system is designed for institute servicing of the electronics and wiring system as is required for a demonstration vehicle of this sort. The boxes

have three access panels, top, side, and bottom. A ‘chassis-vertical’ servicing configuration of the suspension system allows for quick-and-easy access to all the system components and wiring. In the servicing configuration with panels removed, any component on the entire system can be accessed or removed in a matter of minutes.

1.13 Control System

1.13.1 Novel Control Approach

For a remotely operated vehicle ergonomics of the control interface is of the utmost importance. If there were a ‘theoretical upper-bound” for ease-of-use of a ground-based mobile vehicle, it might be the case in which proportional displacements of a standard three axis joystick result in matching and immediate motion of the vehicle. This means the control system would grant an operator intuitive access to all the possibilities of planar-motion.

To implement this mapping on an AWDAWS vehicle, the dimensional constraints of planar-motion must be mapped onto the actuated degrees-of-freedom (DoF) of the drive system. The principle challenge in this mapping stems from the physical systems multiple-redundancy in its ability to satisfy planar-motion constraints.

Clearly any surface vehicle, including the AWDAWS systems, is confined to planar motion, which means all the wheel directions and velocities must conform to a configuration that renders the vehicle drivable. A vehicle state is said to be

drivable when all the wheels are positioned with their rolling axes passing through a single imaginary point, and the velocities of each wheel are scaled in proportion to their respective distances from that point. That point, sometimes called the point of zero-velocity, describes the center of rotation for the vehicle's frame. [56] In the global frame, this point can also be expressed as the vehicle path's radius-of-curvature for any given position along the path. To emphasize the fact that the location of this point is fundamentally transient in nature with respect to both the global and vehicle frames, it is referred to as the instantaneous center of rotation, or simply the instant-center (IC).

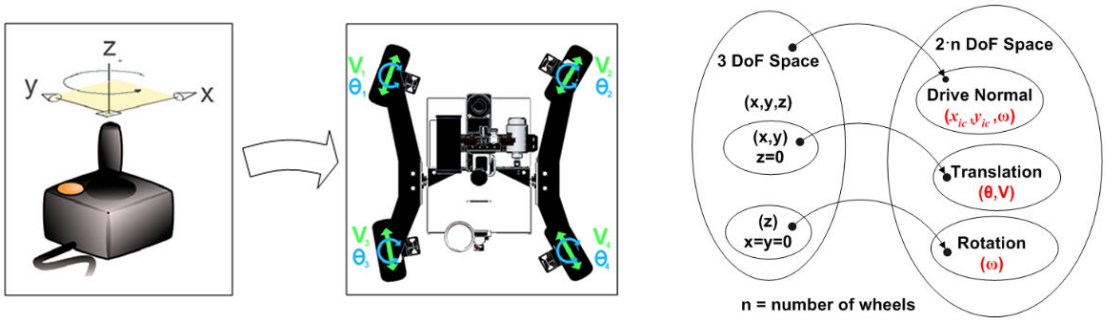

Fig. 4. Illustration of the challenge facing the development of controller (left), and a more quantitative 'view' of the problem space (right).

In general, the location IC and a measure of the frame rotation-rate completely describe every valid configuration in which the vehicle can move, assuming no wheel slippage. Since this application is one of vehicle control, it is most practical use a vehicle velocity to specify the rate of frame rotation. The velocity magnitudes of the individual drive actuators can be calculated based on the IC location. Thus the drive-state of the vehicle can be completely defined by the location of a single IC and a velocity magnitude.

An instant center view of planar-motion, [57] with thoughts toward an omnidirectional controller, leads to an examination of the relationship between the global vehicle motion and the IC location. As seen in figure 5, such an examination results in recognition of three distinct classes of operation. Rotation without vehicle translation is simply rotation about a point central to the origin of the vehicle coordinate frame. Driving along any curved path corresponds to combined vehicle translation and rotation, which is achieved by rotating the vehicle frame about any point other than the frame origin. If the IC is placed at a sufficiently large radius, the rotation component becomes insignificant, resulting in straight line translation.

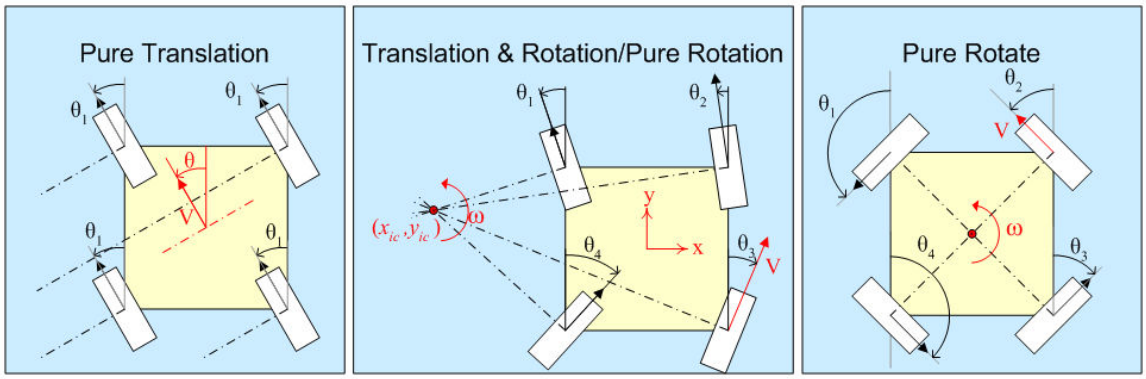

Fig. 5. Three classes of global vehicle motion are shown with respect to IC location, translation (left), combined translation rotation (center), and rotation about a point (right).

The most basic challenge in giving an AWDAWS vehicle true omnidirectional capability is that of making every possible drive-state achievable, while restricting the range of inputs to the three planar motion degrees-of-freedom. The difficulty of the problem is compounded when restrictions to the range of motion of the steering actuators exist. With the controllable range for each wheel limited to

±180°, most, but not all, of drivable-states are redundantly satisfied, which makes optimal and unambiguous definition of the drive-state impossible.

In the context of a control system, however, the more important question centers on the transition between states. If the vehicle is to drive and alter its path while in motion, the system must move from one drive-state to another while maintaining drivability of the system. As illustrated in figure 6, the use of a discrete controller implies a series of incremental transition steps from one drivable state to the next. This transitional process is described by a path consisting of a series of transitional IC steps from the present IC position, to a target.

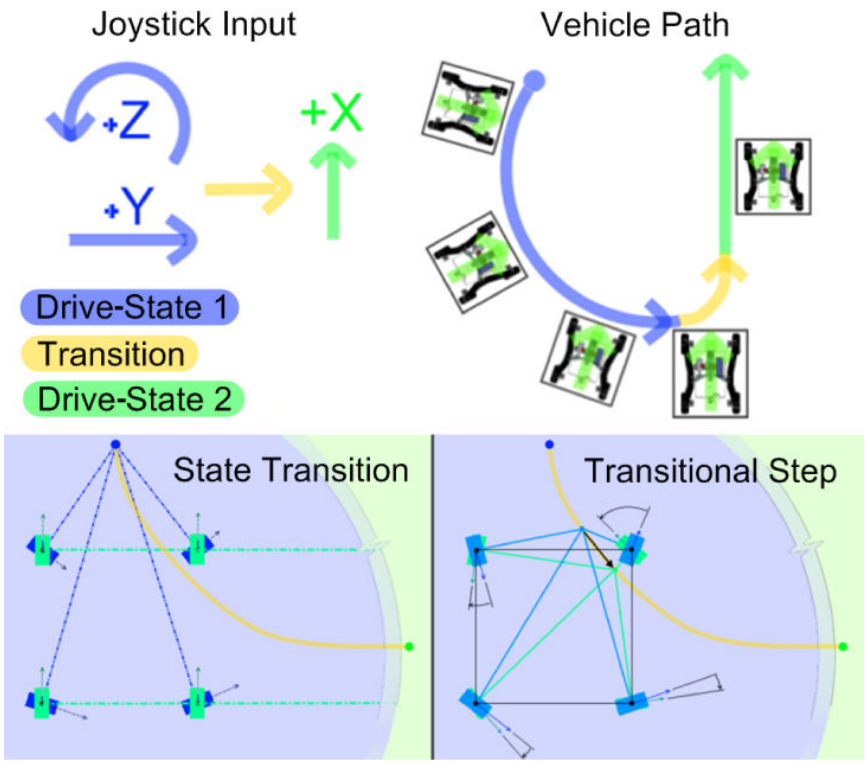

Fig. 6 Transition between two states is shown with the initial state in blue, the transition in yellow, and the final state in green. Different aspects of the transition are user input (left),

vehicle trajectory (left center), instant center motion (right center), and single step in transition (right).

Development of a control-system capable of both setting the state of the system, based on a 3 DoF user input, as well as managing the state-to-state transition dynamics, is challenging. Resolving the constraints governing the general transition between two arbitrary IC specified drive states, in a manner meaningful to a global vehicle, involves a reasonable level of artificial intelligence and decision-making capability.

This approach may hold significant advantages over contemporary control schemes [51] in that it carries the fundamental elements of the control problem and the wheel geometries in pristine form to the highest level of the controller without assumptions of linearity or loss of generality. This holds particular significance to AWDAWS vehicles with variable wheel geometries. The majority of contemporary AWDAWS control systems assume the vehicle geometries to be fundamental, albeit in some cases able to be generalized, within the base assumptions of the construct. The above approach, though rudimentary, may offer a solution that is intuitive to implement and thus readily adaptable to a range of problems associated with control of AWSAWS systems.

1.13.2 Control System

The control algorithm is implemented over six of distinct levels. Each level delivers a different class of functionality, and is schematically shown in figure 7.

The level nearest the user input is the most abstract and the functional abstraction decreases with proximity to the actuators. The first of these levels is the user input itself, which can be from a human or an autonomous user. Level two is a communication layer, and is the highest level housed on-board the cRIO. Level three is a tri-mode mapping function that interprets inputs from the user, and generates mode-specific command sets for the actuator controllers; each command set, or set-point contains a direction and a velocity command for each wheel. Level four executes base functions involving control interaction with physical devices. Level five operates on an FPGA and interfaces between the control-law and the data-acquisition and signal-generating cards. The final level is the physical level: the motor drivers, actuators, and power system. This level is distributed over the body of the robot.

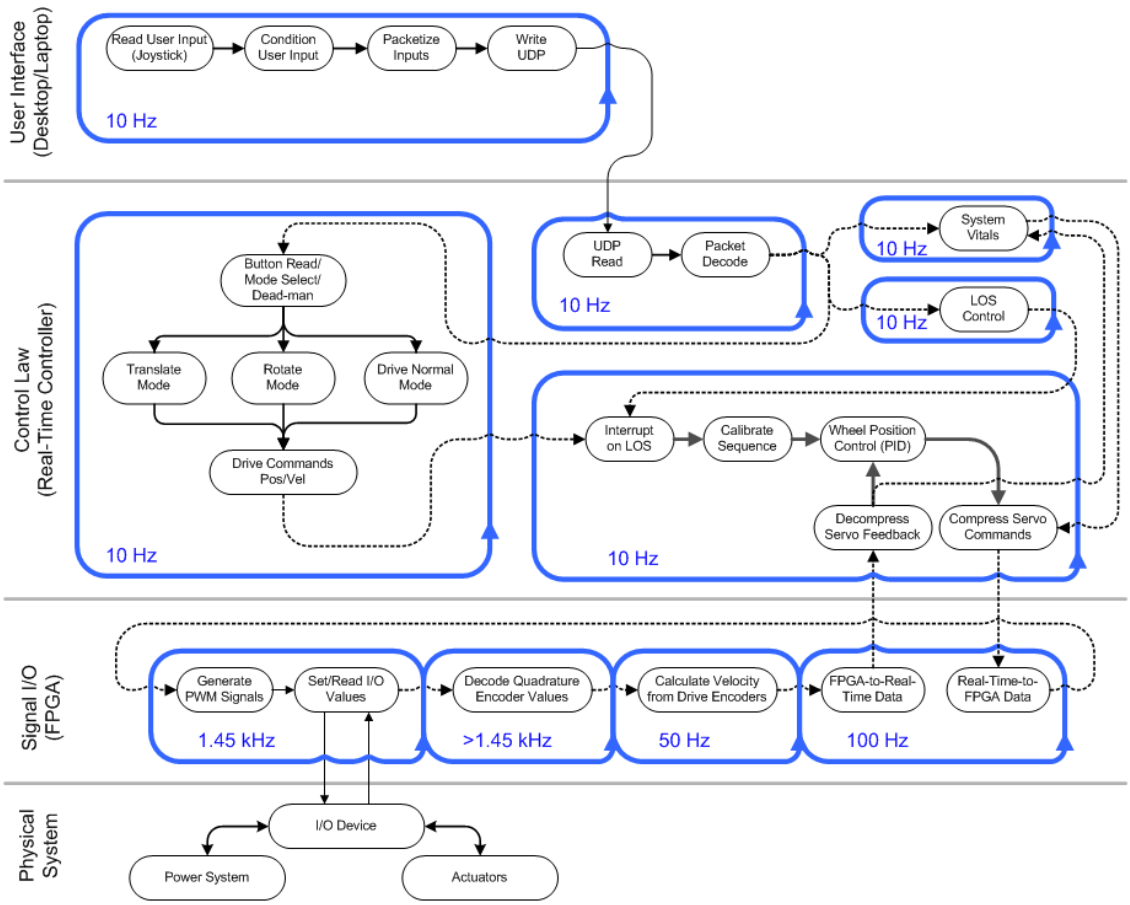

Fig. 7. Control architecture as it is implemented by function across the different levels.

The first two layers, the joystick mapping and the communication, are very simple. First, the joystick is read and the command inputs are mapped to redistribute input sensitivities. The mapped values are transmitted from the OCU via UDP (user datagram protocol) to the onboard controller. This involves building and decoding a packet structure, as well as detecting loss of signal.

The received command signals are used in one of three operational modes based on operator toggle command. The three modes, as outlined in figure 7, are drive-normal, translate in any direction, or rotate about a point. The rotate mode

generates static wheel position commands, and allows the user to control the angular velocity by twisting the joystick. The translate mode takes the joystick translational direction magnitude, and sets the wheel direction and speed commands. The ‘drive-normal’ mode employs a novel geometric state estimator that tracks the geometric motion of the platform. This mode supports double Ackerman steering, single Ackerman steering, or a user-defined fractional combination of the two. In the drive-normal mode, the fore and aft displacement control of the velocity, and the rotation of the joystick controls steering of the wheels.

The servo set-point generated by the tri-mode mapping function is passed to a physical interface level. Here, closed-loop control of the actuators is executed with the use of a proportional-derivative (PD) controller for each of the four wheel position actuators. For reasons explained later, the velocity actuators are run open loop. Also executed at this level is a position calibration routine, the loss-of-signal interrupt, and the interface to the FPGA. This interface involves compression and decompression of commands and actuator feedback.

On the physical I/O and signal processing level, the FPGA carries out four parallel operations to interface with the servos, in addition to controlling the power-board and current consumption monitoring. The first operation is the interface to the real-time processor, which routs signals to a set of PWM generators. The logic values of the PWM signals are mapped to the digital I/O card. Different channels of the same card simultaneously write the control

signals, and read the encoder signals. These encoder signals are then decoded into numeric position values. Finally, the signals from drive encoders are sent to a velocity calculation process before being returned to the original real-time communication loop where they are routed back to the Real-Time processor and the PD controller.

1.14 Science Instrument

The final implementation of the robotic system is geared toward demonstration of a laser based remote sensing instrument capable of performing Lidar-based atmospheric measurements, as well as mineralogical examination of ground-based targets. The components involved in demonstrating this instrument are a directional laser-head that is to be mounted to the pan-and-tilt unit, the electronics needed for operation of the laser and acquisition of data from the various sensor devices, and the software/communication interface required to operate the laser remotely.

4. IMPLEMENTATION

The purpose of this chapter is to give a detailed description of each of the component parts of the robot system. Attention is given to the different aspects of the system based on a personal contribution level. While the mechanical aspects parallel the control-system in complexity and originality, the primary focus is given to the electrical system out of respect for the intended audience.

1.15 Computing, Operating System and Software

1.15.1 Network

The computing network hosted on the robot is an IP-based, local-area network implemented on a NetGear ProSafe 8-port Gigabit Ethernet Switch. A static IP address scheme is used to simplify the configuration of the various sensors and computing nodes, and avoid the need for a routing or port forwarding. Wireless connectivity is enabled with a EnGenius Long Range Wireless Client-Bridge/Access-Point. The wireless-bridge allows remote clients to connect and access the robot's various IP-enabled devices. The wireless link is accomplished with a pair of rugged RadioLabs Omni-8 high-gain antennae; one antenna is mounted on the robot, and the other is connected to the OCU. The antennae are rated for a frequency range of 2.4 to 2.5 GHz. A gain of 8 dBi allows for communication over IEEE standard 802.11b/g (WiFi) at ranges of up to one mile.

1.15.2 Embedded Computers

The actual processing nodes are PCI-104 (PC104) Single Board Computer, manufactured by LiPPERT Components Inc. The model used is a Cool RoadRunner 4 with an 1.4 GHz Intel® Pentium® M 738 processor. The boards are equipped with 1.0 Gb of RAM and 16Gb USB storage drives in place of hard-drives.

Two different I/O boards are used, the first of which is a FireWire break-out board. The COM-1461, made by Parvus/EuroTech Group, is a PC-104 compliant board that supports three IEEE1394 FireWire connections. The FireWire controller board is used for acquiring the image data from the FireWire cameras described later. The complete vision-processing node is a PC104 ‘stake’ consisting of a Cool RoadRunner 4 processor and a COM-1461 card.

To interface with the various serial devices used on the robot, a processing node is assembled with an Xtreme/104-Plus board, that when assembled with the Cool RoadRunner 4 processor, enables a standard serial interface bus. The serial bus is then configurable with the use of pin jumper settings to support four RS-232, RS-422, or RS-485 serial connections. The board is configured to interface over an RS-232 connection, which is compatible with the various devices used on the robot.

1.15.3 Operating System

To provide a user-friendly, if not stable, computing base for the robot, Microsoft© Windows XP Embedded is run on PC104 processors. The XP Embedded package allows for custom configuration of the standard Windows XP operating system, reducing the installation footprint by limiting the functionality of the OS to the specific needs of the given application. The OS compilation used is based on the absolute minimum OS installation with modifications to support a 'headless' boot, which precludes the possibility of directly connecting a monitor. All server-side development, configuration, and testing of the software and operating systems on Ψ, is therefore handled across the Windows Remote Desktop Connection client. This, combined with the use of a Windows® workgroup for sharing optical-drives and hard-disk space for software installation and file transfer, provides a robust and user-friendly, albeit limited, development environment.

1.15.4 Project Structure and Software

Installed on the Windows© OS, National Instrument's LabVIEW 2009 is used as the Ψ's primary software system. Supplemental C/C++ code and driver libraries, usually provided by the device manufacturer, are used to interface with various sensors. The function of this supplementary code, however, does not extend beyond basic data-I/O and device interface layer. All of the signal processing,

data management, and communication functionalities are carried out within the LabVIEW environment.

A core element of the LabVIEW development environment is the project architecture. Projects are used to hold all the indices to the relevant data (libraries, code-base, IO definitions, etc.) required to execute a given LabVIEW program or Virtual Instrument (VI). In Ψ's software implementation, one project is developed that includes a code-base that supports connection and operation of all the devices on the robot. Portions of this standardized project are then manually ported to the individual PC104 processing nodes for execution, as needed. The development scheme for the cRIO is slightly different in that it is treated as an external device whose software is compiled, and then deployed, across a network. In practice, the cRIO appears as a distributed processing 'target' on which code can be run.

To move from a development phase to an operational phase, compilation of executable files supported by LabVIEW is used. These executables can be run independently of the project framework, and require fewer system resources to operate. In the operation of Ψ, such executables are constructed and run on each of the embedded processors, and on the cRIO as start-up processes. The result is a 'tamper-resistant' set of software that executes automatically on start-up with minimal processing load beyond that of the operating system.

General data handling is conducted through the use of LabVIEW's network-shared variables. As the name implies, this is a LabVIEW construct that allows porting of data between IP networked processing nodes. Configuration considerations aside, the use of the network-shared variable is well-integrated into the coding, environment and reduces the implementation complexity of the distributed software system. The computing architecture on Ψ takes advantage of the network-shared variable for disseminating data to the client OCU, and for distributing operator commands to the different on-board processes.

1.15.5 Instrument Support

In the implementation of an instrument demonstration on Ψ, particular care is taken to separate the function of the different systems at the highest level possible. Ψ is intended to support both robotic and science instrument experiments and demonstrations in close quarters. Because of the unstable nature of developmental systems, it is critical the interconnections between the various systems be kept to a minimum. In practice, this means the system should be constructed in such that any major sub-system can fail without compromising the operational integrity of any other sub-system.

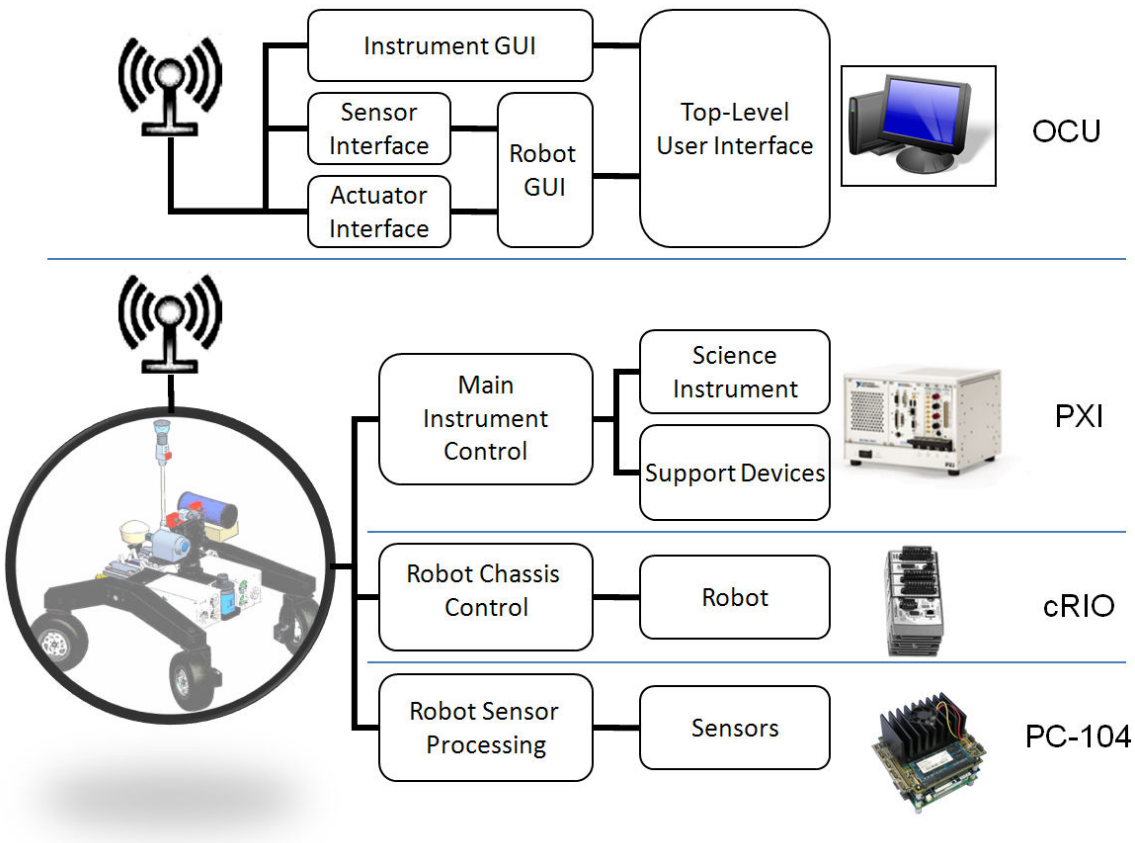

Fig. 8. Software architecture and topology as it exists on the various computing resources.

To meet this separation requirement, both the parallel functionality of the IP network, and the object oriented nature of LabVIEW, are used. As shown in figure 8, the three primary subsystems on the robot: the actuators, the navigation sensors, and the science instrument, are controlled with two separate graphical interfaces – one for the robot, and one for the instrument. Formulating the software architecture in this way allows for independent development of an instrument controller separate from the already-developed robot platform. Appropriate interlocks between the instrument and the robot system, which are

required for operating a mobile, high-power laser, can then be easily added after the instrument's base functionality has been established.

1.16 Sensors

The implemented sensor suite is a subset of the full complement of instruments originally anticipated. While all the sensors were successfully interfaced and tested at one time or another, the incremental complexity, and the design of a user interface for the entire suite, was deemed to be cumbersome and noncritical for demonstration of the target science instrument. A subset of the original suite, highlighted in yellow in figure 9, was thus implemented with logistical considerations given to the remainder of the sensors, as shown in the same figure. The reduced sensor suite was defined by a combination of testing heads-down remote operation, and of the reliability/difficulty of the sensors interfaces. The final sensor-suit consists of a forward-looking camera mounted on the pan-and-tilt unit, an omnidirectional camera mounted atop a mast fashioned from the high-gain antenna, a long-range laser range-finder, and the IMU/Gyro.

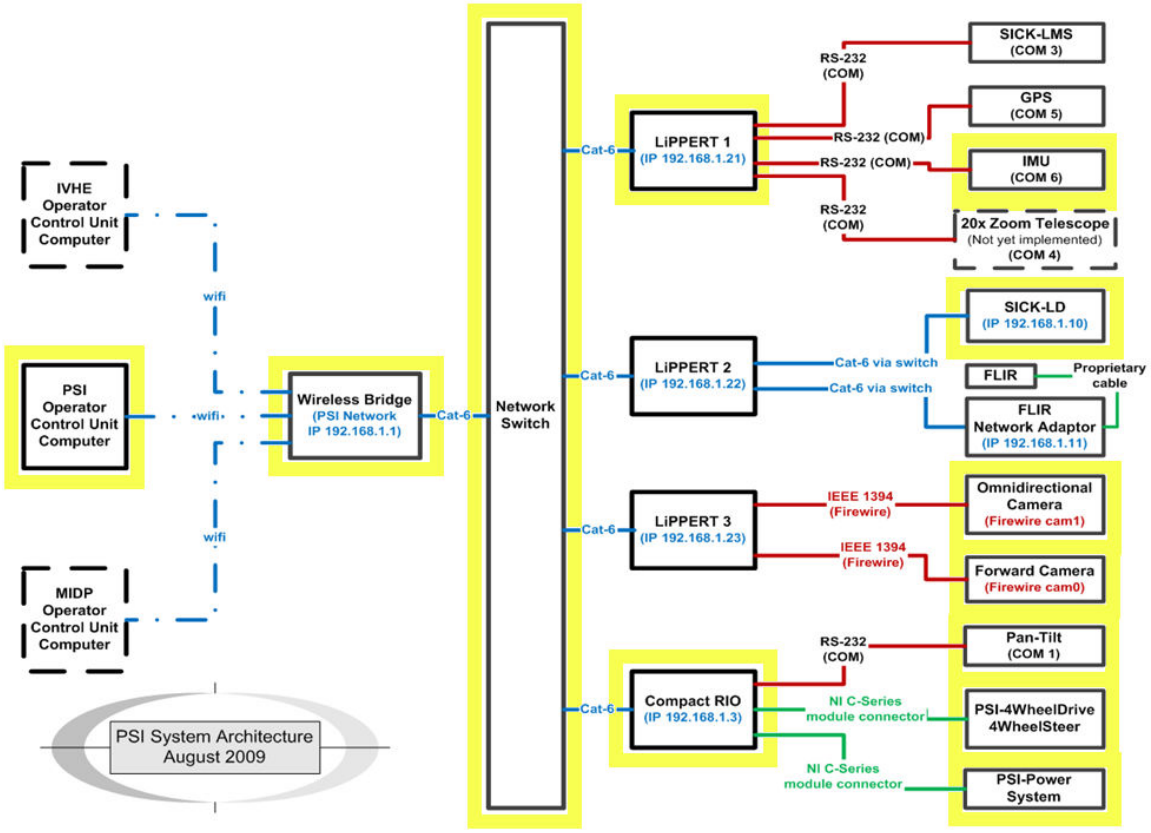

Fig. 9. Map of the entire network architecture, including the connectivity of the full complement of sensors and the components of the system displayed in yellow, that are implemented in this report.

1.16.1 Forward Looking Camera

Target interrogation, for the demonstration of the science instrument, consists of either an upward atmospheric measurement, or a near-range examination of a surface test specimen. The only requirement of such a demonstration is a visual verification of the target being scanned. Thus, as seen in figure 10, the pan-and-tilt mounted directional sensors array was reduced to the forward-looking camera (forward-cam) alone.

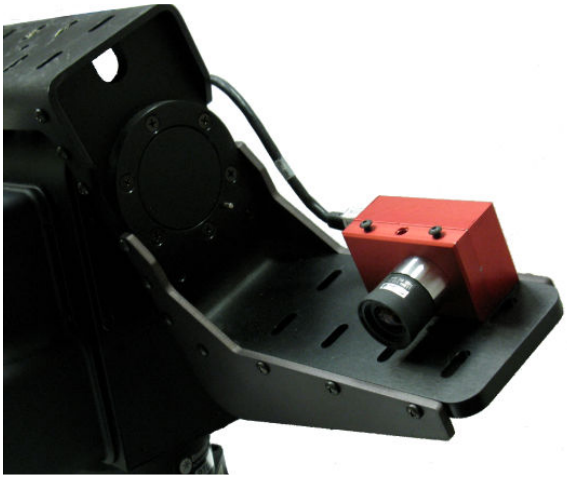

Fig. 10. Forward-cam as it is mounted on the Pan-and-Tilt.

This is implemented with the use of a single CCD imager that was purchased from Videre Design LLC. The model STH-DCSG camera is operated with automatic gain, brightness, and exposure settings to deliver 1024x768 video image over an IEEE 1394 connection. The embedded PC104 processor, with the IEEE-1394 expansion board, hosts a VI that reads the image data from the camera at a rate of 15Hz. The VI compresses the image frame-by-frame with a standard LabVIEW function (IMAQ Remote Compression), converts each frame into a single string, and serves the sensor data to a network-shared variable. The client then reads, decodes, and presents the data to the user.

1.16.2 Omnidirectional Camera

The specification and connectivity of the CCD imager for the omnidirectional camera (omni-cam) is identical to that of the forward-cam, saving two variations: a specialized omnidirectional lens, and ‘un-wrapping’ software, run the on OCU, to a flattened the raw-image into a panoramic view shown in Fig. 12. The lens is

a Hyper110 hyperbolic mirror, manufactured by Eizoh Company in Japan. It takes a 360° panoramic field-of-view, with a vertical viewing angle of +30° to -50° from the horizontal, and reflects it into a 23° camera viewing-angle with a focal-length of 220 mm. The resulting 'donut-shaped' image, shown in Fig. 11, is captured by a Videre STH-DCSG imager, at a frame rate of 10 Hz. The frame-rate difference between the omni-cam and the forward-cam is a result of a bandwidth constraint of the FireWire board. As discussed later, the omni-cam takes a lower priority compared to the forward-cam, and therefore, is allocated a smaller fraction of the bandwidth by frame-rate.

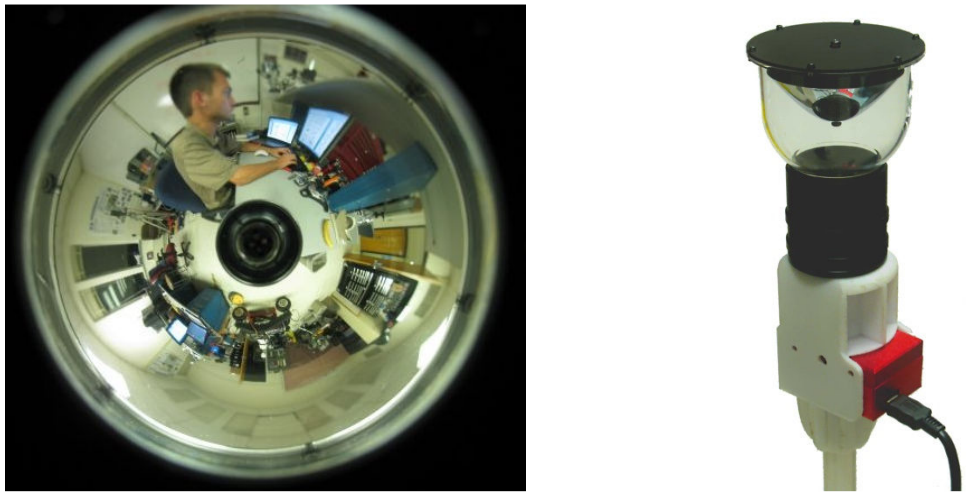

Fig. 11. Raw image from the omni-cam (left), and its mount atop the high-gain WiFi antenna (right).

The raw image is sent via a network-shared variable, to the client OCU, where the un-wrapping transform is performed. The transform is accomplished with the use of a standard C++ library provided by Eizoh. While precise calibration of conic-based vision systems is possible [58], manual adjustment of the un-wrapping transform is sufficiently accurate for the intended application.

Fig. 12. Sample image of the unwrapped panorama generated by the omni-cam.

1.16.3 Laser Range-Finder

Manufactured by SICK, Inc., the LD-OEM1000 scanning-laser range-finder shown in figure 13, is an excellent complement to the visual panorama of the omni-cam. This range-finder is based on an eye-safe, class-1 inferred laser, with a wavelength of 905 nanometers. The invisible laser beam is emitted through a spinning mirror assembly mounted on the top of the drum-like unit. The device is capable of making radial distance measurements in the plane of the laser head of up to 100 meters, with angular spacing between measurement points of 1/8 of a degree. The scan rate and scan resolution are both user defined values, and are set to 10Hz at the maximum resolution. The interface to the laser is handled via UDP over the IP network; such a data packet containing the measurements for each 360° rotational scan is published to the IP port of the device. The laser packets are read by a VI on one of the embedded computers, and the data is published to a network-shared variable for access by the OCU. The client OCU reads the shared-variable, and displays the data in a way meaningful to the user as seen in figure 13.

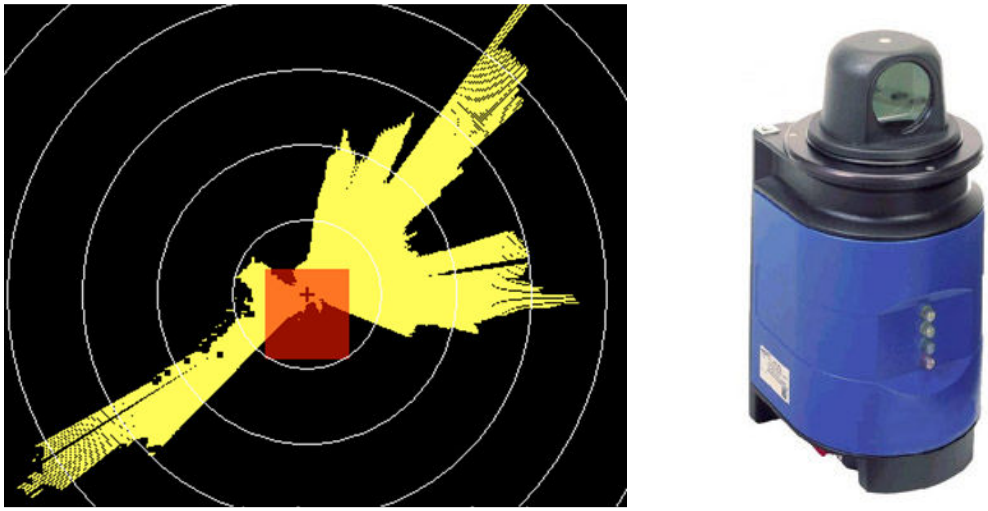

Fig. 13. Data taken by the laser-range finder as it is displayed to the user (left), and the SICK LD-OEM1000 scanning-laser range-finder (right).

1.16.4 IMU/Gyroscope

When demonstrating the science instrument, safety of the experiment and the robot, are the primary localization concern. As a result, the need for the GPS was relaxed, and the unit was removed from the sensor suit. Emphasis was given to vehicle orientation, to give a remote user a feel for the stability of the platform as it makes its way over rough terrain. The orientation measurement is provided by a 3DM-GX1 combined rate-gyro and accelerometer manufactured by MicroStrain, Inc. The accelerometers are rated at up to 5 G's of acceleration, with errors of less than 0.2%. The gyro measures 360° of full rotation on all three axes, to a repeatability precision of 0.2°, and angular rates of up to 300 degrees per second. The device is equipped with an RS-232 serial interface, and is connected to the serial I/O-board of the LiPPERT, where a VI reads the measured values, and publishes them to a network-shared variable at a rate of

10Hz. In the present implementation, only the rotation axes are used to indicate to an operator the orientation of the vehicle, via an icon animation of the robot on the OCU.

1.17 Mechanical Structure

To design the mechanical system, a large number of existing robotic systems were examined. The methodologies, ideas, and lessons learned from these robots, combined with the experience gained from building Mobius and Trinity, led to a fairly clear picture of what is important in the physical embodiment (or the mechanical aspects) of a robotic platform. From these notions, a specification was developed with special care taken to separate actual needs from desires; the needs were listed as requirements, and the desires were categorized as design goals. The requirements and goals were then used as a basis for completing the design of a rolling chassis.

1.17.1 Meeting the Mechanical Requirements

The specification and design of the mechanical system follows from the established goals and requirements of the robot. In the design of the rolling chassis, cost reduction is held as a high priority, such that the requirements and goals are developed into a design based on availability of the various commercial off-the-shelf (COTS) components. The design philosophy being that the engineering time of the COTS provider can be leveraged by first developing a rough design sketch, selecting purchasable items to fill in the major elements of

the sketch, and then completing the detail design using the detail specification of the selected COTS items as a basis.

In following with this design philosophy, the dimensions, sizing, and general definition of the vehicle is first established. The suspension and drive system are the most involved aspect of the vehicle, and thus hold the highest design priority. The drive system leads to the detail specification of the robot chassis, and finally to the layout and packing scheme of the electronics enclosures.

General Vehicle Definition

In formulating the system-level mechanical design, for obvious reasons, precedence was given to the constraints of mobility. The grade and obstacle requirements suggest the use of a kinematic suspension system, rather than a standard spring-based suspension. The ability of such a suspension system to distribute carriage loads evenly to all the wheels, CG offset considerations aside, results in superior traction over diverse terrains. The suspension scheme selected as a baseline is a direct derivative from the wildly successful MER suspension system, [59] and is defined based on the availability of a low-cost, go-cart differential. The differential unit served as the core of a double-rocker differentially-suspended system shown schematically in figure 14. The CG placement is selected to be centrally located with a 50:50 front-to-back wheel loading, and a 60:40 differential loading, with 60% of the vehicle mass being forward of the differential. A tenth of the vehicle mass is then used to keep

positive torque on the suspension elements to avoid excessive bouncing and instability.

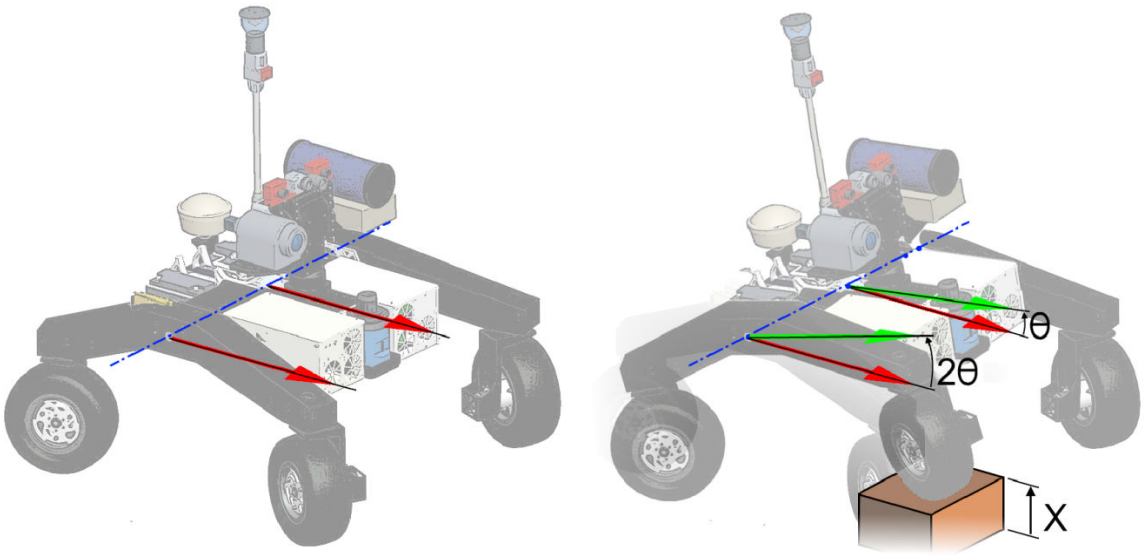

Fig. 14. Chassis as it rests on flat ground (left), and an illustration of the differential suspension in action as one wheel is displaced out of the plane by a distance X (right).

To reduce any propensity of forward bias, a unity aspect ratio was selected as the dimension for a power-caster style, AWDAWS type drive-system. Large diameter tires, to cope with rough terrain, and a need to keep the CG below one-half the wheelbase dimension, driven by the 45° grade requirement, resulted in rocker arms positioned to the side of, rather than beneath, the chassis. The exact dimension of the wheel base, and consequently the rocker arms, was specified by the need to navigate a 12 inch obstacle. This dimension constrained the minimum ground clearance of the chassis, which set the ultimate CG height, as illustrated in figure 15. Once the CG height was fixed, the dimension of the wheelbase was specified.

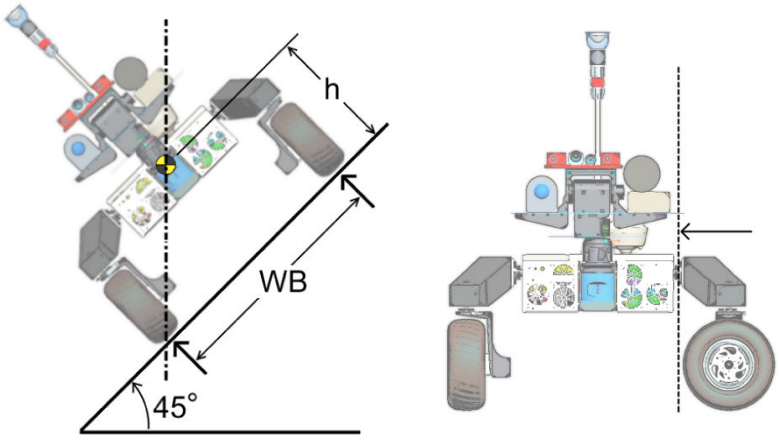

Fig. 15. Illustration of the constraint on the height of the CG above the ground (right), and a constraint on chassis size based on wheel swing (left).

The dimension of the chassis was ultimately defined from the wheelbase and wheel clearance constraints, based on the range of motion of the rocker arms, and the width of a standard doorway. The ‘fit-through-a-door’ requirement, along with a specified wheel-base dimension, led to a removable rocker arm design. The attachment point for the legs is the axle of the differential, whose length represents the minimum dimension of the disassembled rolling chassis. The thickness of the legs, as dictated by the structural loads, and the overall length of the axle drives the final width of the chassis. The other chassis dimensions are developed from geometries of the batteries, sensors, and the two electronics enclosures.

After the mobility requirements have defined the basic envelope of the vehicle, the sensor placement is negotiated based on the priority and requirements of each device. The challenge faced here is that there are too many devices all vying for central location with an unobstructed field-of-view. The laser range-

finder and the omni-camera are both very sensitive to near-field occlusions due to their radial sensing pattern. For the omni-camera, this simply implies a high vantage point. The compromise for the laser range-finder is to mount it squarely on the front and top of the chassis, so it has a clear field-of-view in the forward direction. In this location, it suffers occlusion only from the base of the pan-and-tilt, which is mounted directly to the rear of the laser head.

Finally, to accommodate a perspective science instrument, a mounting interface on top of the frame, in the rear of the vehicle, will be used for mounting an instrumentation/electronics package. The sensing head of the instrument will be mounted on the pan-and-tilt mast. A series of mounting hard-points are located on the top forward portion of the chassis to allow adjustment in the exact position of the pan-and-tilt unit. This adjustment serves a means to adjust the CG, as well as to provide some flexibility to future instruments, which may not need a pan-and-tilt functionality

Drive-Train and Suspension

The most basic component of the drive train, is the wheel itself. Attempts were made to specify the wheels for the robot by diameter, width, and wheel type, based on previous studies. [60] The complexity of the matching an exact wheel specification to a range-of-target use cases ultimately resulted in selection by rule-of-thumb. For convenience and range of availability, a standard pneumatic tire was selected. The specification of the diameter and tread width was

developed by considering the size of the MER vehicles with respect to its wheels. [59] The relative scale was then compared to the size of Ψ, to specify the dimensions of the wheels.

From these specifications, tires, inner-tubes, and wheels with a suitable bolting pattern were obtained. To reduce the overall profile of the caster assembly, the wheel hubs were modified to place the drive motors inside the wheels, as is common in these types of applications. [20] [25] The modification is done by removing the center of the wheel, so the wheel can slide over the motor, which is fitted with a COTS flange collar that clamps to the shaft of the motor. A custom hub, cut from plate aluminum using a computer numerically controlled (CNC) water-jet cutter, supplies a clean interface between the wheel bolting pattern and the bolting pattern on the flange collar. The hub is fashioned so as to fit the inner diameter of the wheel, and bolt directly to the wheel face as shown in figure 16, thus effectively eliminating the need for wheel balancing and alignment.

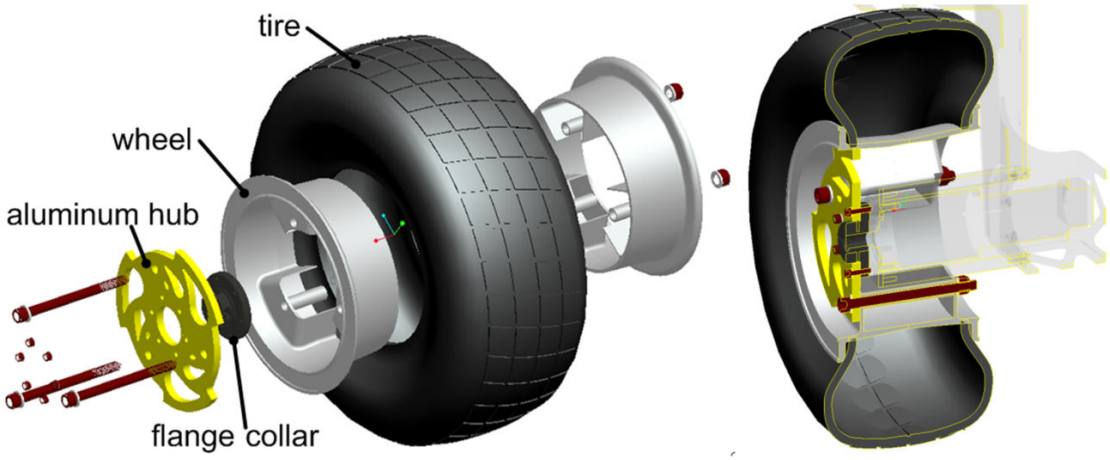

Fig. 16. Exploded view (left), and a cut-away view of the modified COTS wheel as it is assembled (right).

Throughout the construction of the vehicle CNC water-jet is a preferred tool because of its low cost and high precision. The machine accepts two-dimensional computer generated drawings and executes cut patterns directly from the supplied geometry. The water jet is extremely inexpensive to use because the tool-head is a stream of super high-pressure water, so tool ware and pattern intricacy is immaterial.

To support the motor, a five-part ankle assembly was developed as shown in figure 17. The assembly consists of a vertical keyed shaft, or steering shaft, on which the assembly is mounted. A flange collar used to clamp to the steering shaft and provides a bolting pattern that attaches to custom ankle bracket. The ankle bracket was made by Protocase Inc., a company that specializes in low-cost, sheet-metal fabrication. The bracket provides a bolting interface to a mounting tube, which extends from the bottom of the bracket into the cutout of the wheel. Finally, a motor-mount face-plate bolts to the end of the mounting tube. The face-plate provides a bolting pattern, to which the mounting hard-points on the face of the gearbox, are attached.

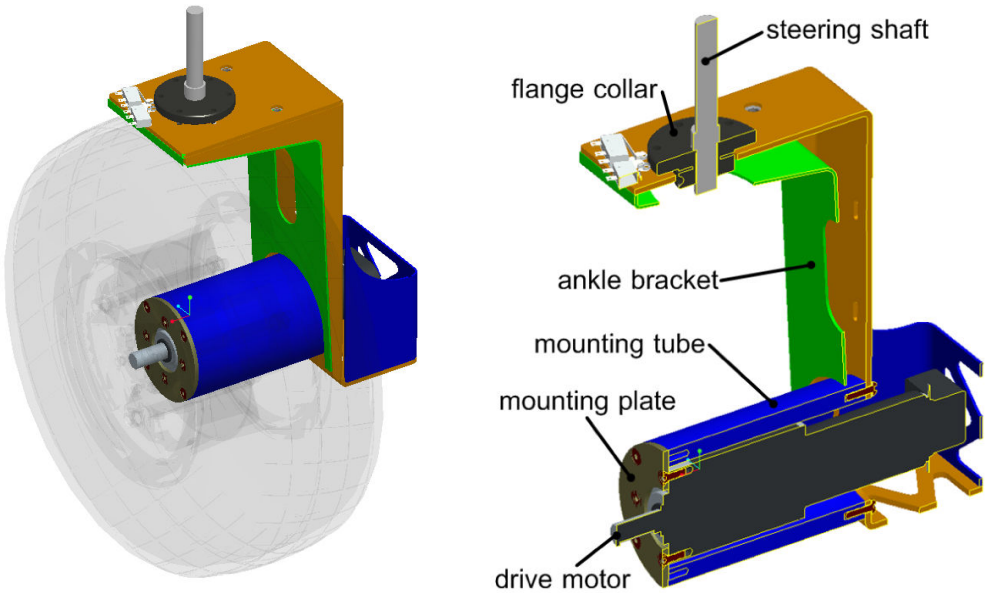

Fig. 17. Ankle bracket with the wheel mounted (left), and a cut-away view of the complete ankle assembly (right).

The drive motor used is a brushless DC motor, purchased from Maxon Motors Company. The unit is the motor/gearbox/encoder assembly RE40/GP52C 126:1/HEDL 5540 that ships pre-assembled with a 126:1 planetary gear-train, and a 500-count-per-revolution optical encoder permanently attached to the motor. The drive motor was specified such that the gearbox of the drive motor could handle the entire bearing load of the wheel, thereby avoiding the need to design for a wheel bearing assembly.

The ankle assembly provides for a clear load path from the output-shaft of the drive gearbox, to the steering shaft. The gearbox mounting plate is made from water-jet cut 1/8-inch aluminum plate. The mounting tube is made of a cut section of thick-walled extruded aluminum tubing. The tubing is cut to length and

prepared on each end with a pattern of threaded screw holes which secure the motor mounting plate at one end, and attach to the ankle bracket at the other.

A conscious effort was made to keep the fabrication complexity confined to the bracket design, thus leveraging the engineering and technical expertise of the vendor. For weight considerations, the bracket is made of two pieces of steel that are welded together to form a closed section of an L-shaped box beam, as seen in figure 18. A third piece is welded on the outside of the ‘L’ to carry the loads from the attached mounting tube and provide protection for the delicate encoder on the back of the drive motor.

The ankle bracket designed is based on a load case developed from a full-body dynamics simulation that was performed in Adams Metaphysics simulation tool. The bracket was then designed using Pro/Mechanica, a finite element analysis (FEA) program incorporated into Pro/Engineer. The principle design load case considered the structure with a load on the output shaft of the gearbox. Resulting from the findings of the dynamics simulation, the load components used are 100 pounds axial toward the bracket for a steering load, and 100 pounds transverse the axel laterally for the drive load. The weight of the vehicle on the wheels was represented with an upward load of 200 pounds transverse to the drive axel, which corresponds with the maximum vehicle loading condition.

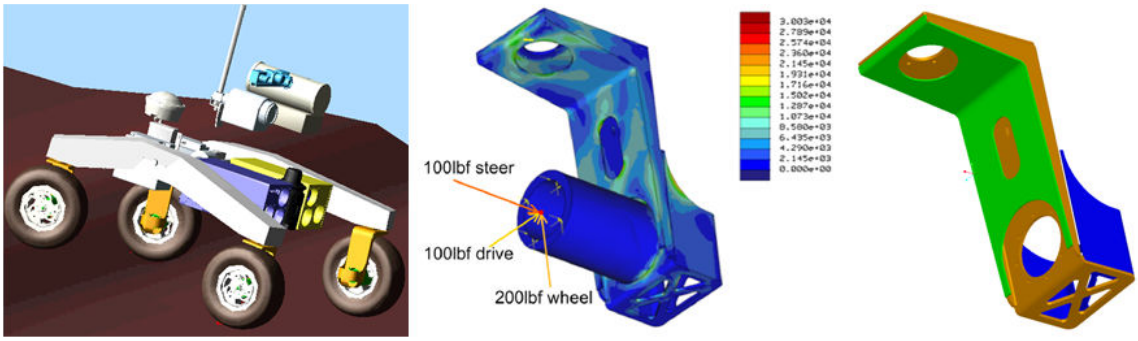

Fig. 18. Screen shot of the dynamics simulation (left), which provided the loads for the finite element analysis of the ankle bracket (center), which yielded the final sheet metal design (right).

The steering shaft of the ankle assembly is connected directly to the steering drive assembly, which consists of a worm drive gearbox, a steering motor, and a drive interface. Balancing backlash and point-accuracy considerations with the unit cost and load-carrying consideration, led to the selection of the 512 BMQ 50:1 worm drive, made by Bravo. The robust construction of this gearbox, like the gearbox used for the wheel mount, allows the ankle assembly to be mounted directly to the drive assembly without further bearings or load-sharing devices. The use of a second planetary gearbox resulted from a mass/space saving decision in which a low-torque, high-speed motor with a second gear train was selected in favor of a more bulky, high-torque motor, connected directly to the worm drive. The steering motor is the RE30/GP32C/HEDL 5540 brushless motor/planetary gearbox/optical encoder that was purchased from Maxon Motors as a preassembled unit, where the gearbox has a 14:1 planetary reduction.

The drive coupling between the steering motor's planetary gear-train and the input of the worm drive gearbox is accomplished with two custom spacers and a

custom shaft coupler. The spacers made from ¼-inch aluminum plate cut on the water jet and position the steering motor to the bolting pattern of the worm drive casing as shown in figure 19. The shaft coupler is made of steel precisely cut to shape with a wire electron discharge machine (wire EDM). The coupler captures the flat on the steering motor assembly output and interfaces to the keyway of worm drive input shaft. Precision machining is avoided by using bolts with locking threads to secure the spacers to the worm drive casing. The spacer and guide bolt configuration allow compliance in the coupling. The whole steering drive assembly is mounted to the inside of the rocker-arm box beam via the bolting pattern on the output face of the gearbox, as shown in figure 19. This output bolting pattern picks up the load of the steering shaft, and thus supports the wheels. To service the ankle assembly, the shaft of the gearbox can be removed by releasing a sir clip on the top of the axle.

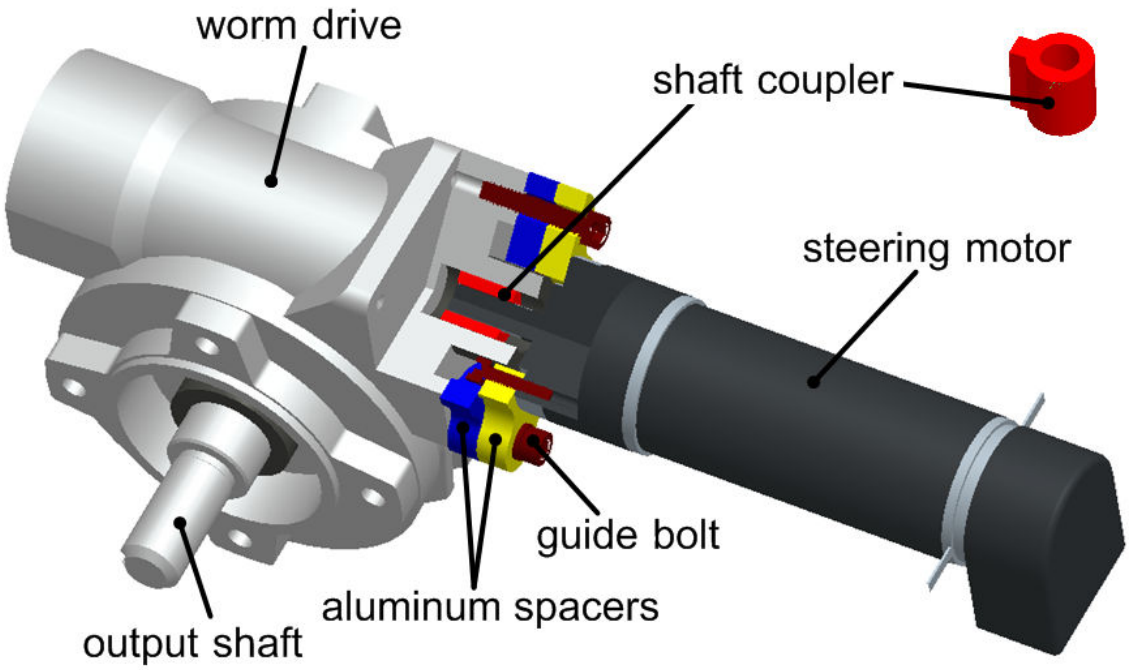

Fig. 19. Cut-away picture of the steering drive assembly with a detail of spacer shaft coupler (upper-right).

Each ankle assembly is fitted with a pair of roller limit switches that bound the range of the ankle's rotation. The switches provide a safety net to keep the steering motor within the programmatic limited of ±180°, and serve as reference point from which to calibrate the position of the steering motor. The switches are housed in a plastic mounting bracket that is mounted around the steering shaft, as shown in figure 20. The rollers of the limit switches follow the perimeter of the flange collar, and are tripped by a divot that switches them to their low state when the positional limits are reached.

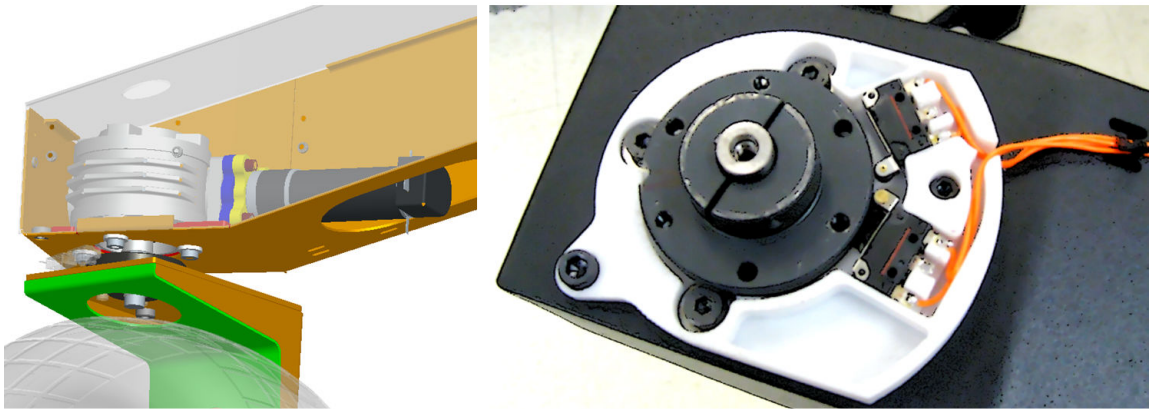

Fig. 20. Steering drive assembly as it supports the ankle bracket with the configuration of the limit switches as seen from below (right).

The structural design of the rocker arms of the suspension system satisfy the same load assumptions used for the design of the ankle bracket, such that the loads are appropriately transferred to the mounting point of the steering drive assembly. The rocker arms are constructed of three welded sheet-metal plates that form a closed-beam structure. To transfer the load of the wheels to the edge of the box-beam structure, the steering drive assemblies are installed on mounting plates that are built into each end of the rocker arm. Additional beam stiffeners, detailed in figure 21, are also needed to transfer the torsional loads from the forward ankle attachment points to the shoulder. At the shoulder, a heavy-gage stiffening web is built into the beam to prevent the beam from collapsing under the increased torsional loading that results from the wheel load being offset from the rocker-arm attachment point.

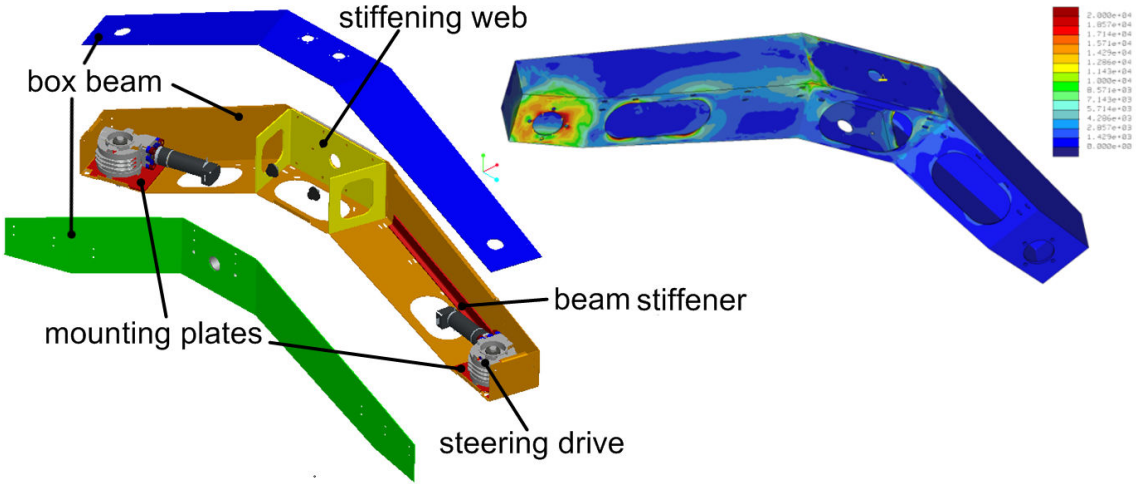

Fig. 21. Exploded view of the welded rocker-arm assembly (left) is shown to expose the stiffeners and drive mechanisms, along with the results of the final finite element analysis (right).

To meet the quick disassemble and reassemble requirement, a quick-release shoulder interface was devised. The interface is based around a quick-release clamping collar placed on the axle inside the rocker arm. The clamping collar holds the inner wall of the leg to an interface pattern, which transfers toque to the rocker-arm structure. The wiring harness for the drive motors is connected through a pair of standard quarter-turn bayonet-style quick-disconnect D38999 socket-and-plug connectors. The sockets are mounted on the top face of the box beam for easy access. The quick-release shoulder interface allows complete removal and replacement of the rocker arm in well under a minute.

The interface is achieved by capturing the differential axel at two points in a single bending as shown on the left of figure 22. The connection to the differential shaft is made with a patterned positioning flange bolted to a flange collar. The mate to this pattern, shown on the right of figure 22, is found on an interface

flange bolted to the inboard wall of the rocker arm. The quick-release clamp collar holds the wall of the rocker arm tight against the face of the positioning flange. The clamp collar uses a ball-bearing mechanism to grip the shaft allowing the unit to slide freely in one direction, while supporting up to 400 pounds of axial force in the opposite direction. The clamp collar is released by simply pulling on a flange that allows the ball-bearing locks to retract. The interface flange and a face-mount pillow block, mounted to the outer wall of the rocker arm, support the load transfer from the axel to the rocker arm. The positioning and interface flanges carry torsional and shear loads, while the bending loads are carried by the combined position flange and outer pillow block.

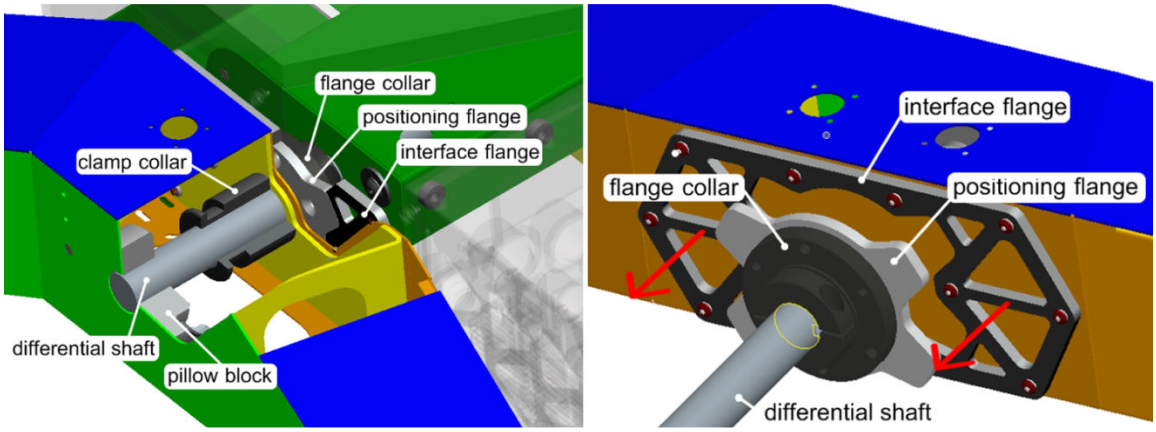

Fig. 22. Positioning flange, flange collar, differential shaft. Interface flange, with quick-release clamp-collar face-mount pillow block

To mount the rocker arm to the robot, the arm is slid over the end of the shaft. The clamp collar is placed on the end of the shaft, before slipping the shaft into the pillow-block bearing. Once the clamp collar pushes the rocker arm completely

up against the positioning flange, the patterns on the flange and rocker arm are seated, and the installation is complete.

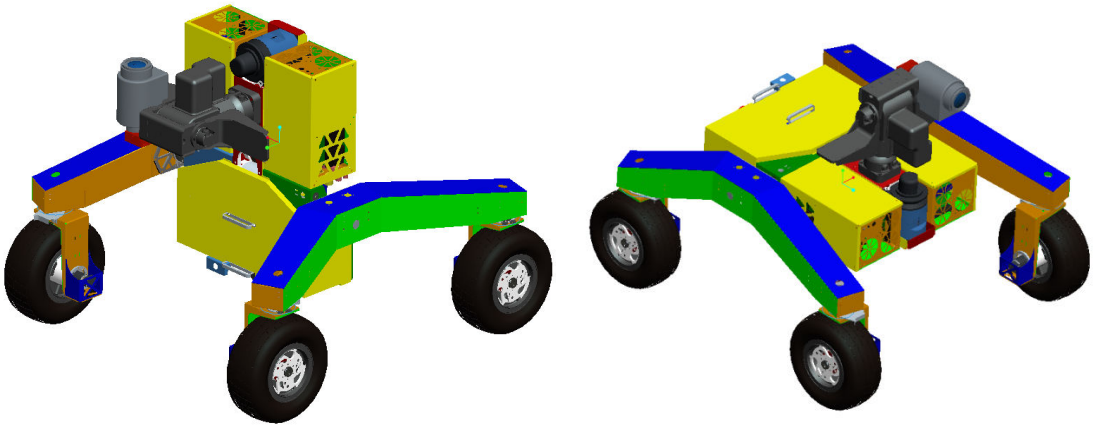

Fig. 23. Vehicle servicing configuration (left), and normal operating configuration (right).

To accommodate the ‘maintenance configuration’ shown in figure 23, the pattern used for the interface flange is selected to be half symmetric, allowing for two different shaft positions 180° apart. By causing one side of the differential to rotate 180°, the chassis rotates 90°, which presents the bottom and top of the rover for easy maintenance.

Chassis

The envelope for the chassis is defined from the general vehicle definition, resulting from the suspension, CG location, and obstacle-clearance requirements. The elements driving the detail layout and dimension of the chassis are the loads and linkage interface to the suspension, and the packing of the batteries, wiring harness, electronics enclosures, and network components.

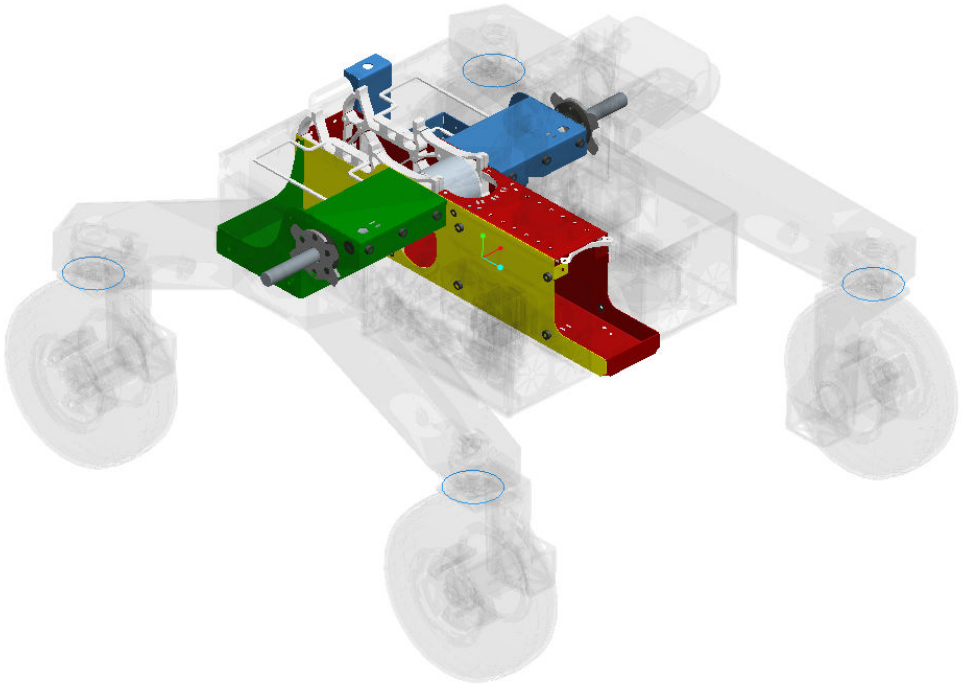

Fig. 24. Rover complete chassis as it is integrated into the vehicle.

The chassis design, seen in figure 24, is based around the differential link of the suspension being located 10% behind the CG; the CG is positioned for a 50:50 weight distribution to the wheels. The batteries are mounted directly behind the differential and act as a counter balance for the electronics enclosures, mounted directly in front of the differential. An open area is reserved down the center of the vehicle to house the wiring harness and components of the onboard LAN.

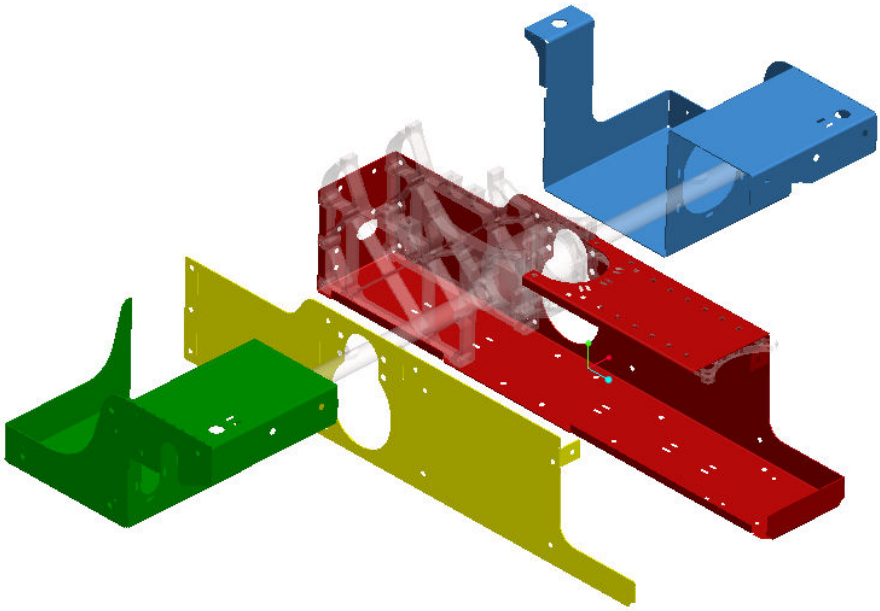

Fig. 25. Components of the welded sheet-metal body assembly.

The robot chassis consists of two parts: a sheet-metal body and an aluminum sub-frame. The body, shown in figure 25, is constructed from four panels of 0.062-inch sheet metal that are bent and welded together to provide mounting surfaces and compartments. The assembly is built around a long box beam that runs the length of the robot and houses the main network components, wiring harness, and other devices. At the back, and on either side of the central box beam, are two compartments for the batteries and other devices. These are formed by open-top boxes adjacent to and behind open-bottom boxes. Both compartments are formed from a single piece of bent sheet metal. The open-bottom boxes house the shaft of the differential and provide a protected, but accessible space for the wiring harness where it connects to the back of the

electronics enclosures. Mounting points for a processor box and a controller box are positioned forward and to the left and right of the frame.

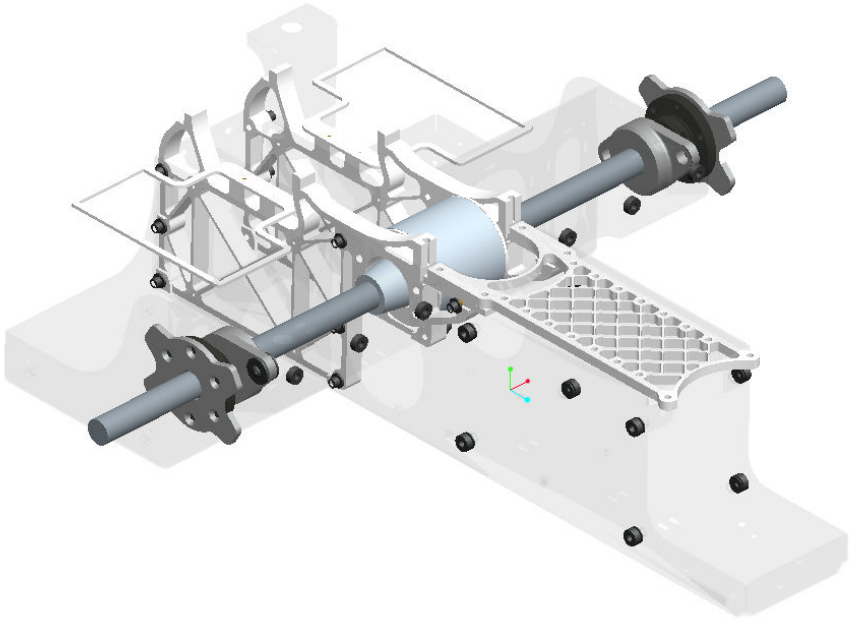

Fig. 26. Sub-frame as it is integrated to the body.

The sub-frame is a bolt-together aluminum structure that attaches directly to the suspension differential, and provides a solid load path to the distributed loads of the sheet-metal body. Three water-jet cut-aluminum plates are bolted together, as shown in figure 26, and inserted into the central box beam of the body. The top plate serves as a foundation for the Pan-and-Tilt and a reinforcement point when it is attached to the housing of the SICK laser. The two vertical frame members are secured to the sides of the differential gearbox and to the walls of the main box beam. Both the vertical members and the Pan-and-Tilt foundation provide secure mounting points for payloads in that they transfer the concentrated loads directly to the suspension system.

Electronics enclosures

The two electronics enclosures, shown in figure 27, provide a protective environment for the power distribution system, the vehicle controller, and the embedded PC104 processors. The definition of the boxes is heavily driven by higher-level design decisions, such that the size and content of the boxes are almost completely pre-defined. To cope with the high component density, a palletized approach was taken in which most of the devices are mounted to removable plates. This allows much of the wiring harness to be built prior to final box assembly. These pallets are mounted between the ends of U-shaped box frames, which are mounted to the robot chassis. The U-style box provides three-sided access to the box contents when the covers are removed. The two end faces allow for convenient accommodation of the cooling hardware and cabling connections.

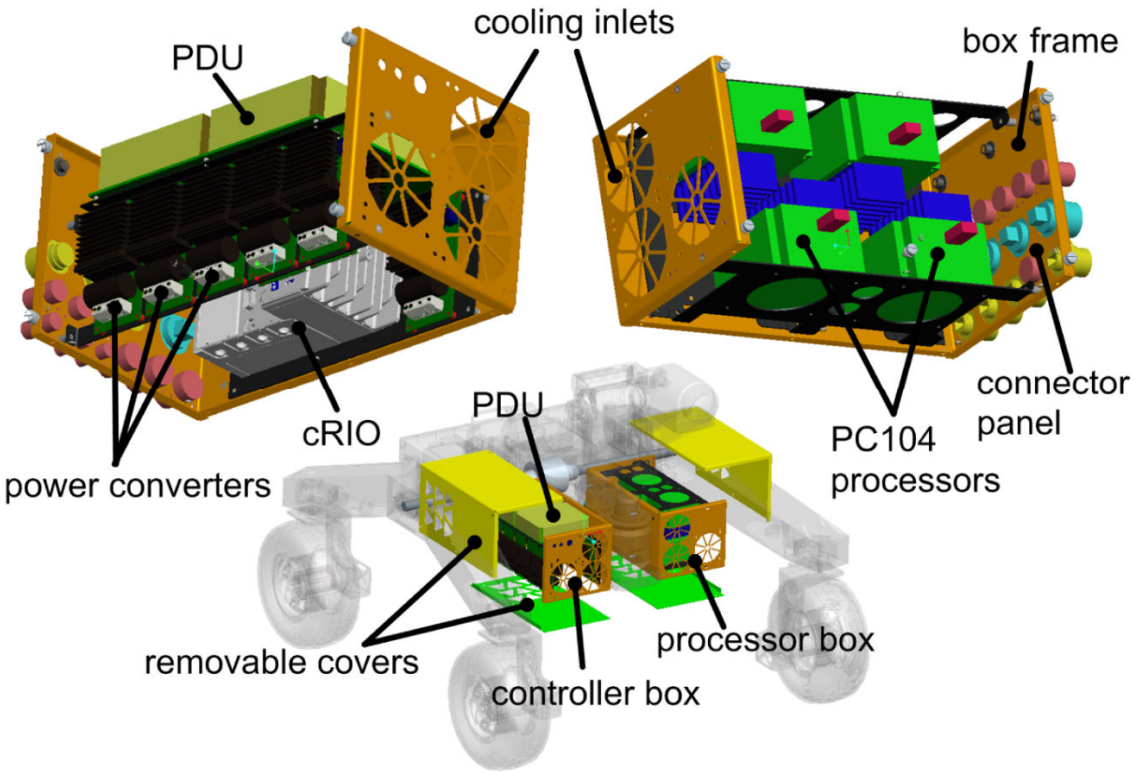

Fig. 27. Detailed layout of the box components and mounting configuration shows the controller box (top left), the processor box (top right), and the access panel configuration (bottom).

The back of the U-shaped box frames are mounted to the center box beam of the robot body, and bolt-on covers are used to protect the bottom of the enclosures. Removable covers close in the top and side of the enclosures, and are fitted with thumb screws for tool-free access to the compartments. When the enclosure covers are removed, and the robot is in the before-mentioned maintenance configuration, a user has easy access to the all the contents of both boxes. Additionally, the cabling/wire-harness connections to the boxes, and the center-body cavity on the top of the rover, are both exposed for easy access.

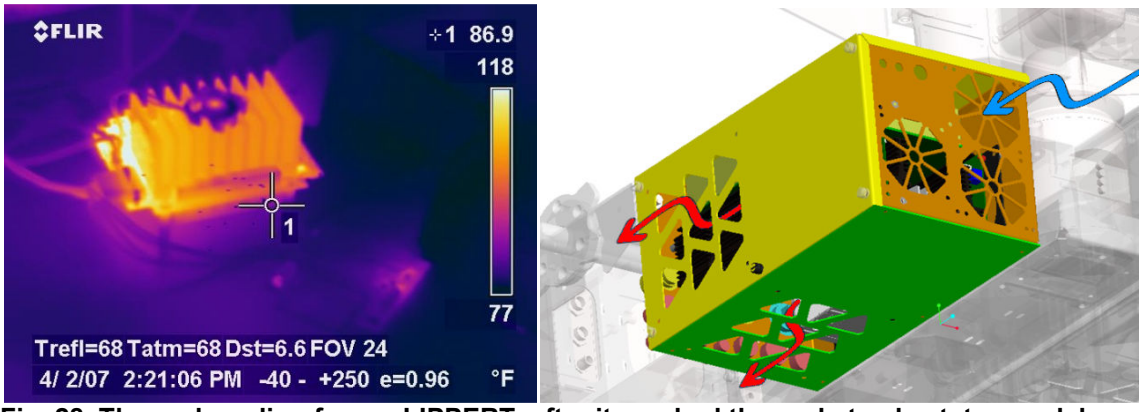

Fig. 28. Thermal reading from a LIPPERT, after it reached thermal steady state on a lab bench (left), and a venting scheme used for the electronics enclosures (right).

The electronics enclosures use a forced convection system, shown in figure 28, with optional inlet filtration to cool the electronics. A first-order thermal analysis shows the option of operating the computers in closed compartment to be non-viable based on current budget levels. Fans draw air through the front of boxes, which passes over the devices and exhausts out the bottom and sides of the enclosure. The PC104 units in the processor box are mounted so the inlet fans blow inlet air directly through the cooling fins on the processor heat sink. The configuration of the controller box is similar with the heat sinks of the power converters aligned to fans, and the cooling fins of the cRIO positioned near the cooling inlet. If future applications have higher heat generating capacities, mounting points for fans to be placed over the exit vents are provided for push/pull convection cooling.

1.18 Electrical and Control System

The electrical and control system of Ψ is considered to be everything between the operator's hand and where the ‘rubber meets the road.’ The description of the system closely follows the order in which the different systems came into existence. The description begins with the physical system and builds through the FPGA signal processing, the control algorithm executed Real-Time processor, the communications link to/from the robot, and the human computer interface.

1.18.1 Power System

Everything in the electrical system starts and stops with the power. The requirement set for the power system stated that the robot should accomplish a reasonable mission to demonstrate the functionality of its onboard science instruments, navigation sensors, and algorithm control methodologies. To meet this requirement, two 40-amp-hour 12-volt U1-12RT batteries, purchased from Valence Technologies Inc., are mounted in the rear of the vehicle. These units are connected in series to terminal blocks on the power distribution unit (PDU) through a prominently mounted ‘main-kill’ switch. The PDU is a custom-built, printed circuit board that fuse protects the batteries from shorts, converts the available 24 volts to required DC levels, and provides relay control for routing power to all systems on the rover. The only component that is routed directly to the battery with no switching, other than the main-kill switch, is the Compact-Rio.

Once activated, the cRIO controls the PDU by sending logical control signals to the relays and monitoring the currents on all routed power channels. Then on command, the PDU routs power to the network devices, the various sensors, the motor drivers, the Pan-and-Tilt unit, and the embedded processors.

1.18.2 Network setup/Connectivity to C-Rio/Software Setup

Under normal operating conditions, as soon as cRIO completes its boot sequence, the cooling to the electronics and the onboard Local Area Network (LAN) switch and the wireless bridge are powered up. The wireless bridge hosts a Wireless Local Area Network (WLAN), and bridges the connection between the WLAN and the LAN. Once active, an operator can connect to the WLAN and remotely access any of the networked devices on the LAN; this access includes the ability to assume control of the cRIO. During development, reliable connectivity is ensured by providing external power to the network devices.

1.18.3 C-Rio Setup (cards/components/etc)

Control of the PDU is managed through two cards: an ‘NI-9205 Analog Input’ card and an ‘NI-9476 Digital Output’ card. The N I-9476 card connects to the PDU and the TTL level logical signals (0-5 Volts) to provide the triggering for the mounted relays. The current sensors on the PDU generate analog signals, which are routed to the cRIO through the NI-9205 card.

The I/O functionality of the Compact Rio is defined by the selection of C-Series modules, which are interchangeable units that plug into the FPGA back plane. In this implementation, all four slots of the Compact Rio are used.

The motor-command generation and acquisition of the encoder and limit-switch signal are handled through two ‘NI-9403 Digital I/O’ modules. The NI-9403 supports user-defined digital input and output on each of the 37 channels. Each channel operates at TTL logic levels and supports switching speeds of up to the 7µ-second per cycle.

1.18.4 Motor control connection.

The two motors and a pair of limit switches are needed for control of each of the four wheels. Motor power is channeled through motor drivers, which require two digital-input control lines, in addition to the 5-volt logic power. The two signals are a logical high/low signal and a pulse width modulated (PWM) signal that drive the voltage polarity and the magnitude of the output voltage. The driver accepts a high-power 24-volt connection from the PDU and delivers a directional and regulated voltage to the DC motors.

1.18.5 Position control limits

A 256-count-per-revolution quadrature encoder returns a motor position to the NI-9403 via two TTL logic lines. The quadrature encoder generates two 50% duty-cycle square waves that are time shifted by 90 degrees. The sign of the shift

corresponds to direction of rotation, and the number of pulses corresponds to an angular displacement. Absolute references for the steering actuator are provided by the limit switches mounted to the steering assemblies. These active low/high switches take a 5-volt power connection from the PDU and, when triggered, route the signal to the NI-9403 data card.

In all, each steering motor requires four digital I/O lines: two for motor command, and two for encoder position measurement. The drive-and-steer functionality of each of the four wheels requires two motor/encoder assemblies and two limit switches corresponding to a total of ten digital control lines. All 40 digital control lines for the drive system I/O are routed to the FPGA via the two NI-9403 modules.

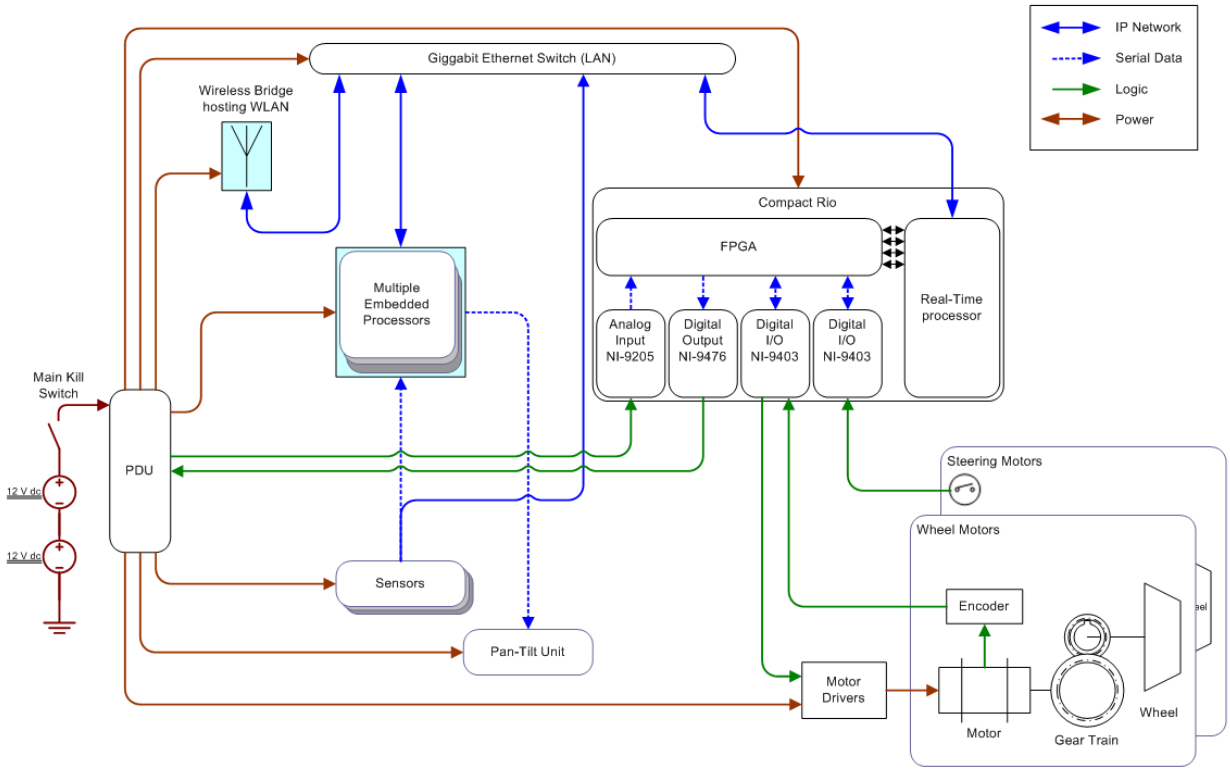

Fig. 29. System-level connectivity diagram for power, logic, and data transmissions.

For reasons of future serviceability, it is important to note that reliable functionality of the system depends heavily on robust implementation. The connectivity, shown in figure 29, is implemented with each device properly connectorized and documented. Additionally, bulkhead connectors are installed at every system boundary to allow for system-level trouble shooting and maintenance.

1.18.6 Signal Processing, I/O and FPGA

The software-hardware interface and associated signal processing is all handled onboard a Xilinx three-million-gate field-programmable gate array (FPGA) that is

integrated into the cRIO control unit. This layer interfaces to the NI C-Series modules, where it reads the digital and analog input lines and sets the state of the digital outputs.

Implementation of this code base was made possible through the use of LabVIEW FPGA Development Module for Compact Rio. The module supports the development of FPGA code through a translation interface that converts the native LabVIEW code into an instruction set for the integrated Xilinx compiler. In practice, the compilation process is completely automatic, assuming convergent FPGA code, and a ready-to-use ‘bitfile’ is immediately downloaded to the FPGA. The code is then executed with a standard VI interface. It is worth mentioning that in the context of the cRIO, efficiency of FPGA circuit design is sacrificed for ease of user interface, and subsequent versions of LabVIEW use the FPGA resources more efficiently.

The digital design was fashioned in four asynchronous processes, shown in figure 30, that handle the PWM generation and the physical I/O to the cards, the quadrature decoding, the calculation of velocities, and communication with the Real-Time controller. The final code was driven to this form based on two primary constraints. The constraints were the size of the FPGA itself, and the bandwidth limit of the combined multiplexer/chassis on the digital-I/O cards.

In an effort to maximize the actuators rate and maintain positive control, the limitations of the cRIO chassis must be considered. The high-count encoders

(500 counts per revolution) used on the servo motors can easily outpace the sampling rate of the digital-I/O cards in a no-load condition at low-command voltages. Thus, it is critical that high-cost calculations or long-duration logic constructs are executed in such a way that the propagation delay of the hardware is kept as the limiting factor. This is achieved by separating the read/write operation into its own process loop with no programmatic time delay. The cRIO/C-Series module system supports sample periods and dwell times of up to 7μS and 70nS, respectively. These timing values include the performance gains achieve by predefining the input/output state of each buffer element. When using static I/O definitions, only one read/write function call to the hardware is permitted in the code being executed. Thus, the PWM output and the quadrature signal input must be synchronously executed. The final result is that the PWM generation logic and the hardware I/O library call, which run on the standard 40MHz clock of the FPGA, execute faster than the cRIO bus update speed.

On every iteration of the PWM/IO process, the values of the limit switch and the quadrature signal are read and stored to local FPGA memory. From this location, a quadrature decoding process accesses the stored quadrature value. The decoding process, whose state architecture is sketched in figure 30, executes extremely quickly with an execution period of just five clock cycles. Aliasing of the quadrature signal count is then avoided, because the decoding process executes at approximately 8MHz: 1/5 of the chip clock. The bus speed limit, which controls the update rate of the quadrature signal, is limited to ~140kHz.

The high-count encoders, combined gear train ratio of 700:1 (two reductions of 14:1 and 50:1), make integrity of the encoder-count critical. Poor quality on the steering-position count results in loss of position reference, which, as discussed later, ultimately results in mechanical strain to the structure due to wheel misalignment.

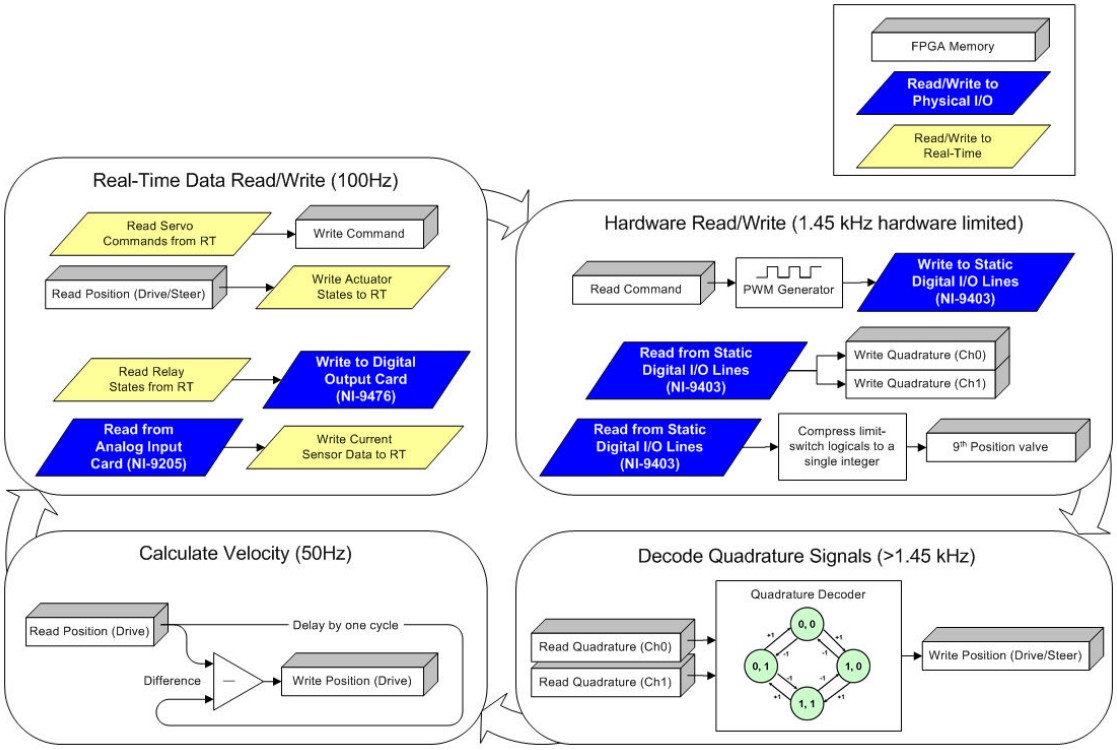

Fig. 30. Processes and data flow of the digital design implemented on the FPGA.

The output of the decoder is a signed count that records the displacement of the motor armature. To avoid overflow, the steering quadrature count is recorded with 32-bits of signed integer precision, and scaled with a factor of 2^{-8} to avoid logistical ramifications of carrying unneeded precession. This scaling brings the maximum value of the position displacement to within 16-bit precision. The

resulting position indices are posted to memory with 16-bits of signed integer precision. From here, the drive encoders are read again into differencing a loop where velocities are calculated. To manage the resolution of the velocity calculations, a programmatic time delay is used to slow this process to a time scale meaningful to a global sense, e.g., an execution frequency of 50Hz. The calculated velocity values are then written to memory, where they are read into a compression process that formats the data prior to porting it the Real-Time controller.

The communication channel between the Real-Time unit and the FPGA operates at 100Hz and is based on a 32-bit word. Because getting signals on and off the chip is an expensive proposition, significant efficiency can be gained by optimizing for the IO to comply with the systems architecture. Up to this point in the digital design, all the position/velocity values, passed from loop to loop, are stored in 16-bit precision memory. To accommodate the IO constraints of the chip, the values are paired and sequenced as integer elements in an array of 32-bit precision. The final array is length five, where the state of the limit switches is encoded into the last array element. These switch values are later used for a calibration sequence.

In practice, this data transfer is made by simply making a LabVIEW front-panel 32-bit array indicator. This value can be read with a library call on the Real-Time controller. Concurrent with the reading of the system state, a command array specifying the actuator rates is passed to the FPGA; also in the form of a 32-bit

signed-integers array of length five. This command is decoded into the eight 16-bit servo commands before it is stored in FPGA memory.

The actuator commands from the Real-Time controller are interpreted as a duty-cycle command to the PWM signal generator. The duty-cycle command is a number that corresponds to the number of high counts per period; each period contains a total of 1024 counts. The PWM generator sets the state of the PWM pin to high until the duty-cycle commanded count is exhausted, after which it sets the state to low for the remainder of that PWM cycle. The 1024-count period of the PWM cycle, combined with the bus speed of the cRIO, results in a PWM frequency of about 140Hz.

Additional logic incorporated into the PWM generator is used to interlock the servo command signal with the signals from the limit switches. This interlock puts a definite limit on the range of motion achievable by the steering actuators. Limiting the motion in this way ensures that the range of motion of the actuator is limited at the very lowest level of control. The system is thus robust with respect to failure of higher lever code.

1.18.7 Closed Loop Control

The actuator-control level, fully detailed in figure 31, is the highest level of the control system that is directly involved in servo control. The remaining levels are the system controller that handles servo coordination, the input mapping that interprets the commands from the operator, and the communication level. The

actuator controller reads from, and writes to, the system controller and the FPGA signal processor. The process reads the state of the actuators via the FPGA, which provides feedback for the proportional-derivative (PD) controller.

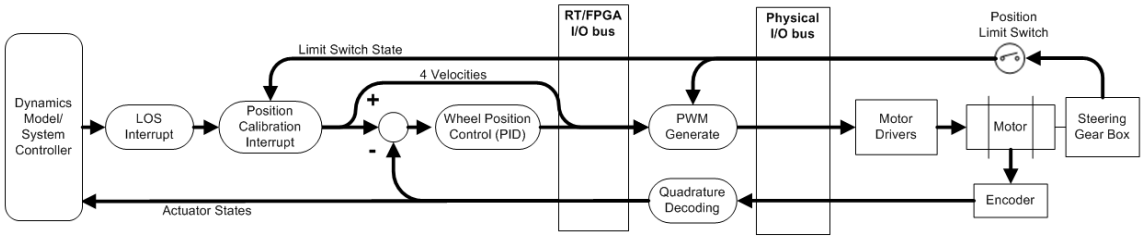

Fig. 31. Flow chart of the entire closed-loop controller with all its programmatic electronic and physical elements.

The actuator state is also passed to the system controller, which in turn provides set-point values for each of the servos. The PD controller is accompanied by a loss-of-control (LOS) interrupt and a calibration interrupt that sets the home position for the steering actuators by cycling between the limit switches. The LOS and calibration interrupts operate on the set-point from the system controller by either blocking the signal or, in the case of the calibration, supplying their own set-points.

Interrupts

The interrupts are activated by triggers. In the case of an LOS interrupt, the set-point to the PD controller is forced to zero when the LOS trigger indicates the absence of direct control. (The detection mechanism is discussed later.) The trigger for the calibration interrupt is provided by the user, and sets in motion a routine that asserts its own commands to the PD controller, in place of the

position commands from the system controller. The calibration routine functions by first commanding the wheels to a position greater than one revolution. Once the limit switches indicate that all the wheels have reached their maximum position, the position count on the FPGA is initialized to zero, and each wheel is commanded to a predefined displacement that puts the system in a good, forward-driving configuration. Once the routine detects that all the wheels are within a certain tolerance of the target command, the position count is again set to zero. The interrupt then relinquishes control of the set-point value to the system controller.

Scaling the IO

Before any work can be done on the control system, the data to and from the FPGA needs to be scaled and compress/decompressed to and from the 32-bit signed integer arrays mentioned earlier. For ease of coding and clarity in higher-level code, standard ranges of velocity and wheel angle are selected: ±180° for the steering, and [-100, 100] for the velocity. These standards are not only important for coding convenience, but are essential for executing closed-loop control such that the set point, command value, and feedback signals are all of the same scale at the control point. The command values to the PWM generators are all scaled with a common factor of 10.2 to span the 1024 PWM count. The measured velocity value is scaled with a factor of 1/20, which is derived from the 50Hz differencing period. The position value is a very large number, on the order

of 10^6 quadrature counts for one rotation. The return-position value is adjusted with a fine scaling factor of 1/8.5, in addition to the course scaling of 2^{-8} performed on the FPGA.

PD Control

The design of the controller is carried out with ‘acceptable performance’ as the objective, leaving optimization to later endeavors, if increased performance is required. Resulting from the artificial rate limit derived from the encoder bandwidth, it was found that the drive motors posses sufficient control authority to allow for open-loop operation. Thus, the only actuator needing a controller is the wheel-position servo.

The parameters considered in the selection of the position-controller design are transient response, overshoot precision, and steady-state pointing accuracy. Surprisingly, the pointing accuracy is the least demanding of the three due to gear-train backlash and mechanical slop in the wheel pivot mount. Because the vehicle is designed to be teleoperated, the actual slew time of the actuators was the most critical parameter; the objective being to cause the system to respond to the user's input absolutely as fast as possible. Additionally, a small overshoot is needed to avoid excess power consumption and undesirable servo noise associated with a steady-state error, caused by the resilience of the tires and the backlash of the gear train. A small amount of overshoot allows the tires and gearbox to come to rest and the motor currents to go to zero.

The needs of the controller point toward the use of lead controller, whose steady-state accuracy may not be the best, but whose transient response is needed. All the desired effects of a lead-control scheme can be attained with a proportional derivative (PD) controller. The system is implemented with a standard PID controller found in the LabVIEW control library.

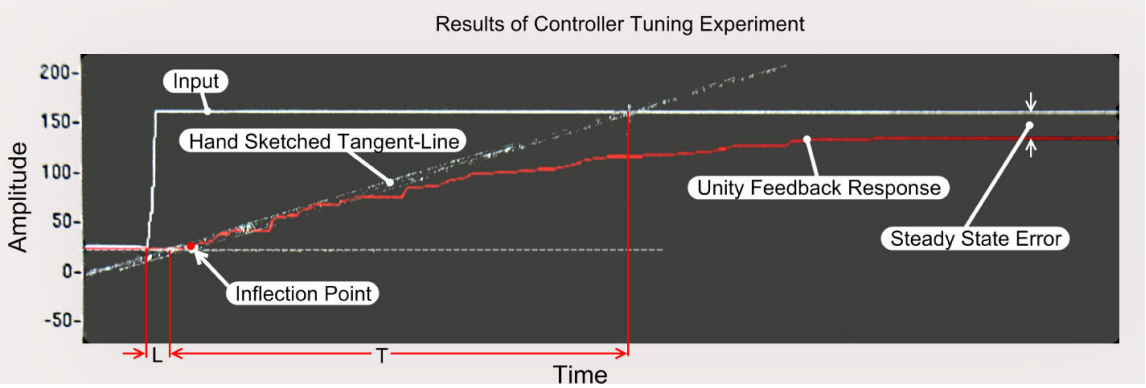

Fig. 32. Digitally master photograph of the hand-sketched graphical differentiation.

Tuning of the PID was performed by loosely following the Ziegler-Nichols rule for PID controllers. [61] This approach calls for a graphical analysis of the transient response of the system, which is outlined in figure 32. The rule depends on a tangent line, located at the inflection point of the S-shaped response curve, to determine two characteristic constants. The constants are used to evaluate first-guess values of the *P*, *I*, and *D* parameters. Once the system exhibited stable performance, the *integral* term *I* was incrementally diminished to zero to yield the final PD controller.

1.18.8 Operational Modes

Based on the functional requirements of remote deployment of a science instrument, a controller is used that restricts the range of motion to a subset of omnidirectionality. A three-mode scheme is implemented that demonstrates a novel control approach for the AWDAWS system. The modal scheme allows control by instant-center without the encumbrance of a generality. The primary mode accommodates comfortable, forward car-style driving of the vehicle, and is useful for open-area navigation. The two remaining modes, translate and rotate, allow the vehicle to execute the primitive motions for basic maneuverability in tight or constrained settings.

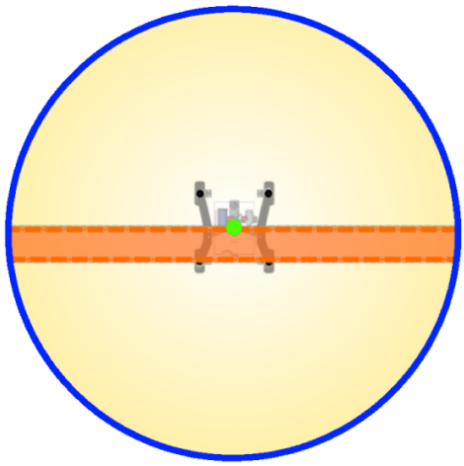

Fig. 33 IC-plane where the green dot is rotation, the blue circle is translation, and the orange box is the allowable range of combined translation and rotation.

These modes, shown in figure 33, correspond to particular regions of the instant-center plan. The departure point of this approach is a tri-mode mapping function, which maps the users input onto the IC-plane. The geometry of the IC-plane is used to generate actuator commands, based on the specified mode. The modes

are drive-normal, translate, and rotate, of which only the drive-normal mode possesses all aspects of the proposed control by instant-center. Though the drive-normal mode is not capable of general plane motion, many aspects of the objective omnidirectional control are preserved, such that the novelty of the control approach is demonstrated. This demonstration also bears testament to the expectation of future development. The rotation-mode by definition has no component of path planning. The translate-mode, for expedience, is implemented with a simple direction-and-velocity command scheme

Drive Normal Mode

The drive-normal mode places a constraint on the motion of the vehicle, such that the instantaneous center of rotation resides on a specified line-of-zero motion. The operational mode supports customization in the performance of the controller by allowing adjustments to the location of the line-of-zero motion, and the range of the wheel steering angle. The location of the line is controlled by a steerage ratio, which positions the line within a range of steering configurations that vary between double- and single-Ackerman steering. In the case of the double-Ackerman steering, the line-of-zero motion passes through the center of the vehicle. In the single-Ackerman steering case, the line is positioned over the rear wheels, causing the vehicle to drive like a conventional car.

The drive-normal mode is constructed in three basic sections that are illustrated in figure 34: the user interface, the IC-plane state-transition controller, and the

output modifying behaviors. The user interface is simply a parameterized mapping function that transforms the user's command to an input suitable for the transition controller. In particular, the mapping converts a joystick input into an instant-center position and a vehicle speed command.

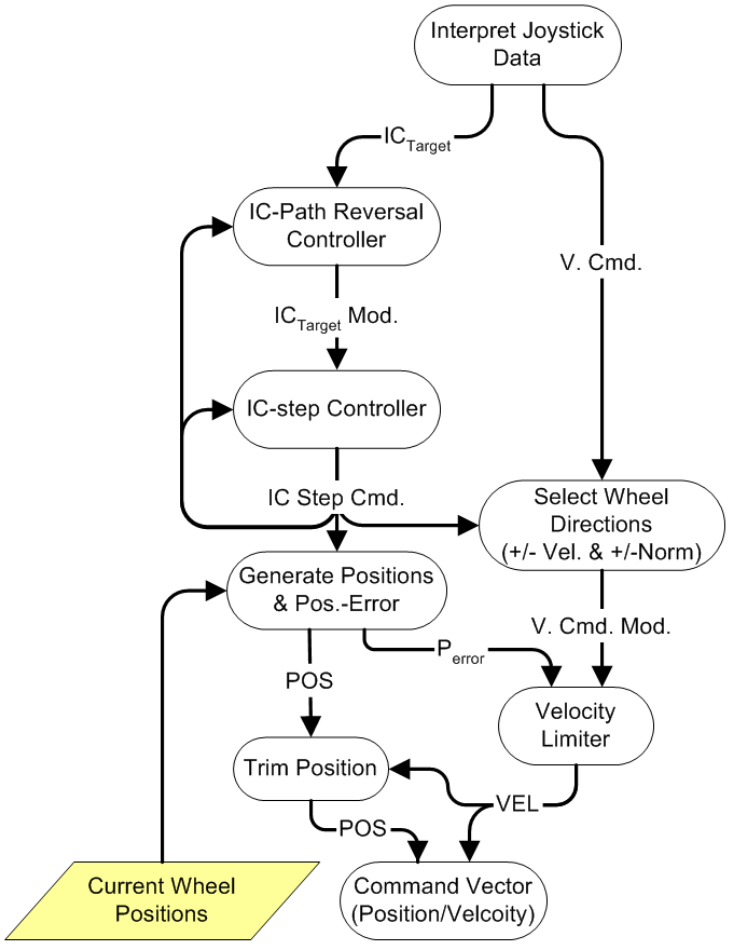

Fig. 34. Componentized flow chart for the drive-normal mode.

Once the user's input has been mapped to a point in the instant-center space, it is the task of the transition controller to cause the system to move to the target configuration. This is accomplished with two behaviors: The first, the IC-transit behavior, models and regulates the dynamics of the AWDAWS system. The

second, the path-planning behavior, augments the path tendencies of the IC-transit function to generate a meaningful IC-path. These two functions operate in feed-forward fashion with memory of the previous state.

The first two behaviors define the set-point for the closed-loop actuator controllers. The set-point command vector is generated in two parts: the position command and the velocity command. The position command is calculated with respect to the IC location. The velocity command is derived from the joystick input and is modified in scale and direction for each wheel, based on the location of the IC. A safety behavior inhibits the velocity command if the position error is outside of tolerance to ensure that the vehicle is in a drivable state prior to powering of the drive servos. The final component is a trimming behavior that compensates for mechanical slop based on the direction of vehicle motion.

User Interface

The drive-normal user interface maps the user's commanded joystick inputs to a location on the instant-center plane. The location, referred to as the target instant center (target-IC), is a point relative to the robot frame that is representative of the vehicle configuration corresponding to the user's input. To generate the coordinates for a target-IC, the z-axis of the joystick is first converted to a vehicle path radius-of-gyration, which corresponds to a radius R from the origin of the vehicle frame to the target IC that resides on the line-of-zero-motion. To allow

customization of the control scheme, parameterized limits on the turning angle and steerage ratio are supplied to the operator.

The basis for calculating the radius of gyration is derived with the use of an imaginary fifth wheel, shown in figure 35, which is located on the centerline of the frame and even with the front wheels. This scheme results from the notion that intuitive operation can be achieved if a user could control the vehicle by directing the motion of such a fifth wheel. To realize this mapping, a simple trigonometric function is used to evaluate the radius to the instant-center from the vehicle origin.

$$R_{IC} = \frac{x}{\tan(b \cdot \theta)} \tag{1}$$

Where R_{IC} is the radius to the IC, θ is the steering angle, and x, as shown in figure 35, is the distance from the imaginary wheel to the zero-motion line.

From the plot in figure 35, it is clear that the form the function meets the general needs of the mapping. When the joystick is near neutral, a discontinuity between extremes is needed such that the IC ‘jumps’ to the appropriate side, corresponding to slight correction in course. As the steering angle approaches 90°, the IC needs to draw to the origin of the frame.

The ‘fifth-wheel’ approach makes the values at the edges very important from a user control standpoint. The region surrounding the neutral position of a typical

joystick is generally sensitive, but not accurate. It is thus important to have some measure of intent from the user before relaying a command to the system. This is accomplished with a dead-band zone, which translates to the feature denoted with dimension D in figure 35.

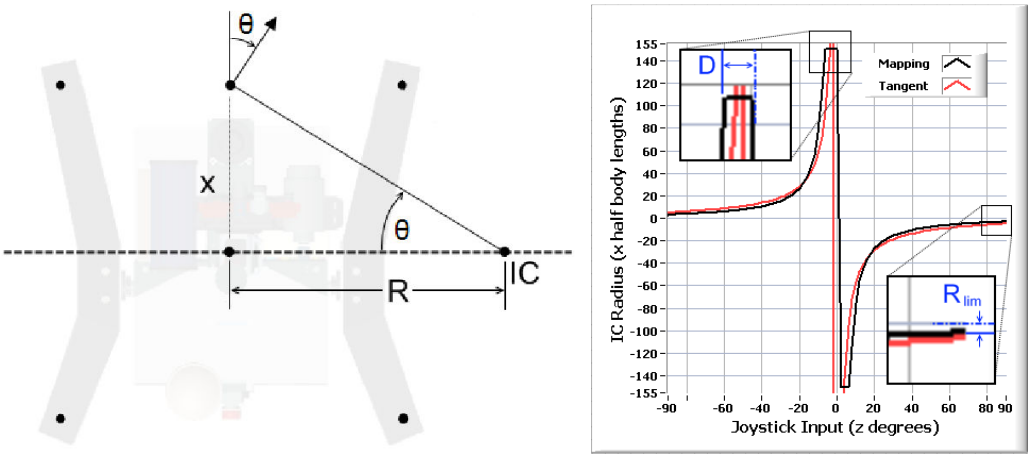

Fig. 35. Geometric construction of the trigonometric function (right), and comparison between the trigonometric function in red (coefficient of x = 5 and b = 1) and the mapping function in black (coefficients of angle limit θ= 45 deg and steerage ratio a = 1) (left).

In the other extreme of the mapping, the case of a maximum turn also requires fine adjustment such that the vehicle should rotate about a point. This means that the mapping must drive the IC location to zero for ±90° of joystick input.

In the above trigonometric function, the scalar x is a magnitude adjustment, which corresponds to change in both the dead-band and radius-of-curvature. The variable *b* affects the period of the tangent function, which effectively adjusts the R_{lim} value. In fine adjustment of the mapping, independent control over the *D* term, the R_{lim} term, and the shape of the curve are needed.

To satisfy this need, a variation of the above trigonometric function is developed whose overall shape, shown in figure 35, is very similar to the tangent function. The difference in form allows for ordinate and co-ordinate shifts in the asymptotes, while maintaining control over the shape of the function. The mapping function

$$R_{IC} = \left(\left| \frac{c}{|z_{joy}| - d} \right| - r \right) \frac{z_{joy}}{|z_{joy}|} \tag{2}$$

is tunable, as shown in figure 36. A scaling parameter *c* provides adjustment to the shape of the function. An *ordinate asymptote* parameter *d* specifies the input dead-band D. And a *co-ordinate asymptote* parameter *r* specifies the minimum IC radius R_{lim}.

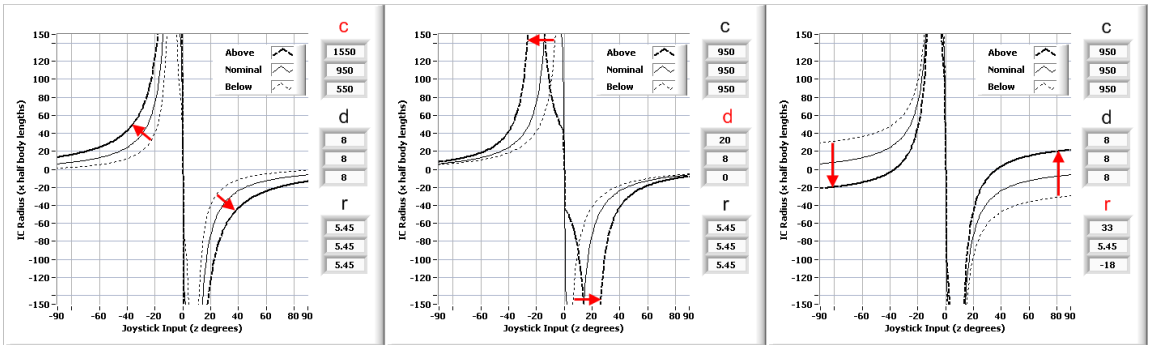

Fig. 36. Mapping function with a range of values for the scaling parameter c (left), the ordinate shift/dead-band parameter d (center), and the co-ordinate shift/IC-limit parameter (right).

The only remaining concern is the domain of the IC, which corresponds to the radius of infinity. The radius of infinity, or R_{∞}, is taken to be 75 times the wheel-base dimension or 150 half wheel bases. The final radius command to the system is evaluated as

$$R_{cmd} = \min(150, R) \tag{3}$$

The complete mapping function used to generate the radius command is then

$$R_{cmd} = \min\left[150, \left(\left|\frac{950}{|z_{joy}| - 8}\right| - 5.45\right)\frac{z_{joy}}{|z_{joy}|}\right] \tag{4}$$

Selection of the *c*, *r*, and *d* parameters was made experimentally with the vehicle 'in the loop.'

To implement a limitation on the range of motion of the vehicle, an angular limit of a given wheel is considered.

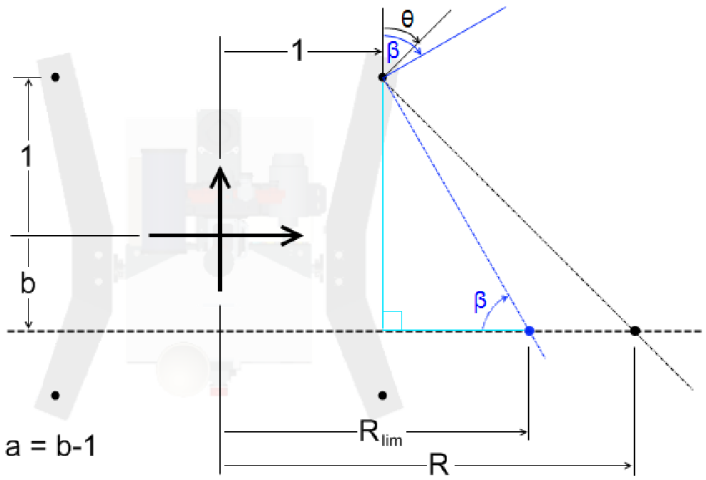

Fig. 37. Geometric construction for the steering angle limit as a function of the steerage ratio.

As shown in figure 37, *a* limit *β* is established based on a maximum desired angular wheel displacement. The limitation is translated to the zero-motion line

as a minimum IC radius R_{lim}. Rather than simply setting a value to $R_{lim,}$ a general relationship between the angular-limit and the magnitude of R_{lim} is established.

$$\tan(\beta) = \frac{1+a}{R_{\lim} - 1} \tag{5}$$

In this expression, a, R_{lim}, and β are the before-mentioned steerage-ratio, IC-radius limit, and angular wheel limit, respectively. A general expression is needed for R_{lim} because the line-of-zero motion restriction varies with the steerage ratio.

$$R_{\lim} = \frac{1+a}{\tan(\beta)} \tag{6}$$

Because R_{lim} is dynamically assigned based on the placement of the line-of-zero motion, the overall form of the input mapping must be modified if loss of control range is to be avoided. This is accomplished by linearly scaling and shifting the command to within the range of the non-restricted region of the line-of-zero motion.

$$R_{IC} = R_{cmd} + R_{cmd}\left(R_{\lim} \cdot \frac{150 - |R_{cmd}|}{150 \cdot |R_{cmd}|} \right) \tag{7}$$

The final result, shown in figure 38, is a function that maps ±90° of z-axis joystick input to an instant-center location that is confined to ±150 unit lengths, or half wheel bases, of the zero-motion line.

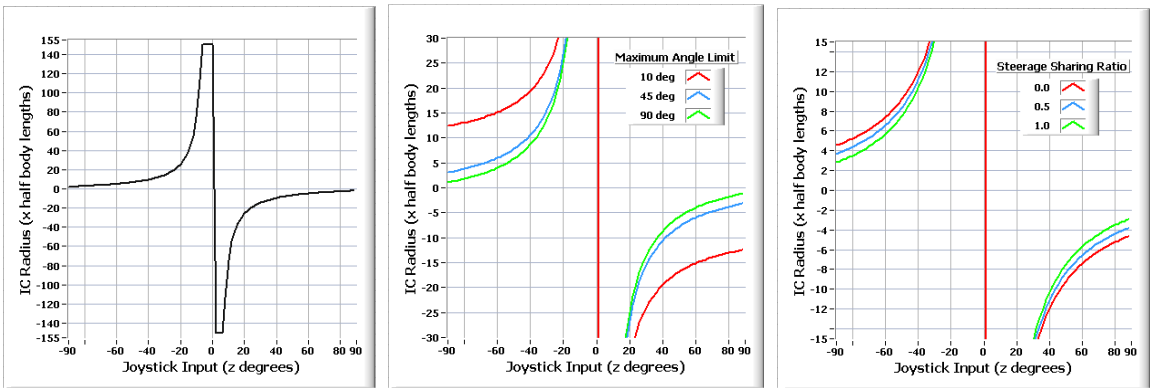

Fig. 38. Standard transformed curve with 45-degree steering limit and unity steerage ration (left), steering angle limit comparison with fixed unity steerage ration (center), steerage ration comparison with fixed 45-degree steering limit (right).

The mapping provides precise output control at the extreme inputs, and reasonable control of the intermediate regions. The result is a well-defined dead-band region, parameterized limitation to the vehicle's range of motion, and an operator-specified steerage ratio allowing continual adjustment between single- or double-Ackerman steering schemes.

IC-Space for AWDAWS systems

The IC-transit behavior is developed with the construction of the IC-plane as the basis for control of an AWDAWS system. The objective of the controller is to transition the system from one drivable state to another, while maintaining vehicle drivability at all times, and causing the system to comply with the constraints of planar motion. This is done by holding a current drive-state in the form of a present-IC, and incrementally transitioning the current state to a target drive-state, or a user-specified target-IC. The transition controller itself falls away from the behavioral control model in that it is predictive rather than reactive in its

modeling of the system dynamics. However, the overall system remains behavioral in nature as the transition controller is an individual element, whose output is modified by other functions.

The limitations of the physical system are captured in the IC-transit behavior with an angular rate limit, which results from the above-mentioned bandwidth constraint. The transit-behavior is a member of the tri-mode mapping functions, and thus executes on the fixed clock of 10Hz. The exaction period results in a maximum step angle, which is derived from the before-mentioned physical rate-limits set by the position controller. The controller rate limit is 29 degrees per second, which gives a step-limit angle (referred to as θ) of approximately three degrees. The value of the controller rate limit was developed based on signal quality measurements, and was selected to maximize actuator speed.

Once the step-angle limit is set, it must be converted into a step-magnitude for the transition-vector increment. To do this, the geometric relation between all four wheels and the transition-vector must be fully comprehended. In general, the angular slew rate of one wheel will drive the step-magnitude *L*. The fact that the closest wheel is not necessarily a limiting wheel introduces a subtle complication. This is illustrated by the case in which the present-IC, the wheel closest to the present-IC, and the target-IC are all co-linear. In other words, the transition vector points directly away from the closest wheel. The implication is that the vector does not reach beyond the ‘one-step region’ shown in figure 39.

The calculation of the step-magnitude relies on the transition vector and a wheel vector, which extends from a designated wheel-of-interest to the present-IC. From these two vectors, a transition heading *α* can be calculated, which relates the angle of the transition-vector to the wheel-of-interest. The wheel-vector and the angular step-limit are used to define a one-step region. The angle formed between the edge of the one-step region and the transition-vector is defined as the boundary angle *β*. From the boundary angle and the transition heading the largest possible step magnitude *L'* is calculated. This maximum step-magnitude will vary in length in the direction of the transition vector and will always terminate at the edge of the one-step region.

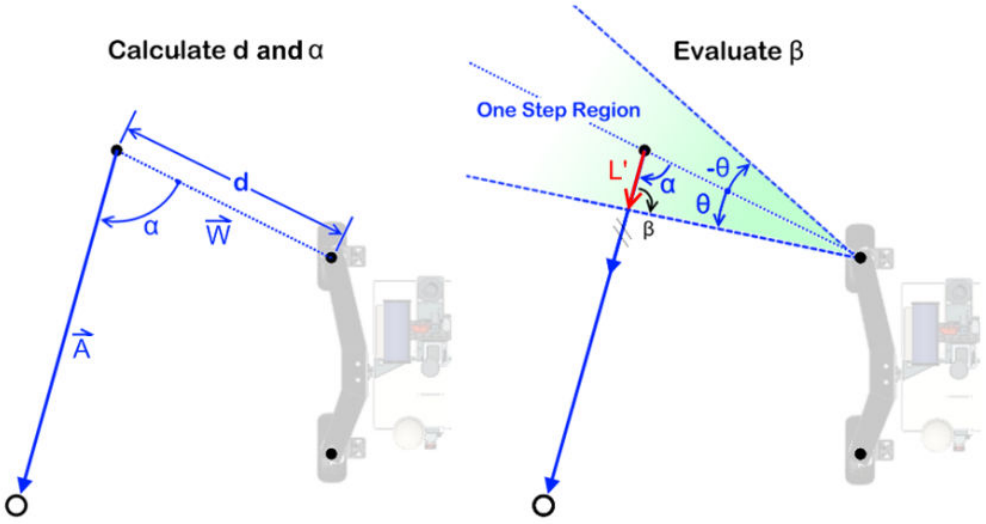

Fig. 39. Transition-vector, A; and wheel-vector, W, with magnitude d; and constrained transition heading α (left). One-step region and step-limit angle, θ; the boundary angle, β; and the step magnitude (right).

To implement the above behavior, the angle between the wheel vector and the transition vector must first be calculated. This angle calculation is shown in general in figure 40 as angle-ABC corresponding point definitions and resulting vectors.

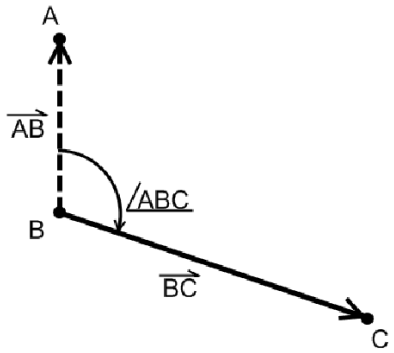

Fig. 40. Angle calculation from three points, where *B* is typically the IC, *C* is the wheel of interest, and *A* is the reference point.

The angle is calculated by solving the inner-product of the vector pair; defined as

$$\overrightarrow{BA} \bullet \overrightarrow{BC} = \left|\overrightarrow{BA}\right| \cdot \left|\overrightarrow{BC}\right| \cdot \cos\left(\angle ABC\right) \tag{8}$$

To compensate for the absence of a sign in the resulting angle, the unit vector of the cross product is included as a coefficient.

$$\angle ABC = u \cdot \cos^{-1}\left(\frac{\overrightarrow{BA} \bullet \overrightarrow{BC}}{\left|\overrightarrow{BA}\right| \cdot \left|\overrightarrow{BC}\right|}\right), \text{ Where } \quad u = \frac{\overrightarrow{BA} \times \overrightarrow{BC}}{\left|\overrightarrow{BA} \times \overrightarrow{BC}\right|} \tag{9}$$

The resulting angle has a range of ±180°, and from the sense of the cross-product, is defined with respect to the AB vector. In the definition of the transition-

heading angle α the correct sense of the is achieved when the cross-product is evaluated as the wheel-vector crossed with the transition-vector.

Because α is bounded on ±180°, the boundary angle β can be evaluated by the total internal angle of the oblique triangle formed by the wheel-vector, the transition-vector, and the boundary of the one-step region,

$$\beta = 180 - \alpha + \theta \tag{10}$$

where the sign of θ is the angular step-limit.

With β defined, the angles of an oblique triangle are known, and the step-magnitude can be calculated as seen in figure 41.

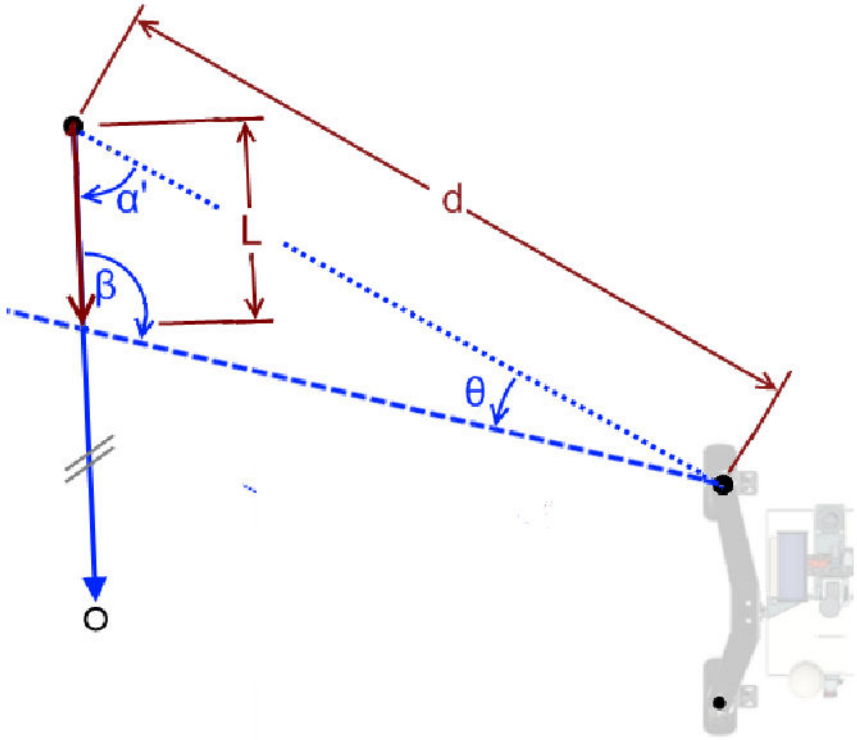

Fig. 41. Geometric construction of the step-magnitude calculation.

The step-magnitude of the transition vector is calculated using the Law of Sines as,

$$\frac{\sin \beta}{d} = \frac{\sin \theta}{L'} \tag{11}$$

where d is the distance from the wheel to the present-IC, β and θ are of course the boundary and step-limit angles, and L' is the prescribed step-magnitude. With this form, we can easily solve for L'.

$$L' = \left| \frac{d \cdot \sin \theta}{\sin \beta} \right| \tag{12}$$

This solution is a general one for any transition-vector, such that it specifies a step magnitude whose length is defined in terms of the locus of origin of the transition-vector with respect to an arbitrarily wheel location.

The final step is to calculate an overall system step magnitude limit. To avoid the need of detecting which wheel is the limiting factor, *a priori* a step-magnitude is calculated from each wheel.

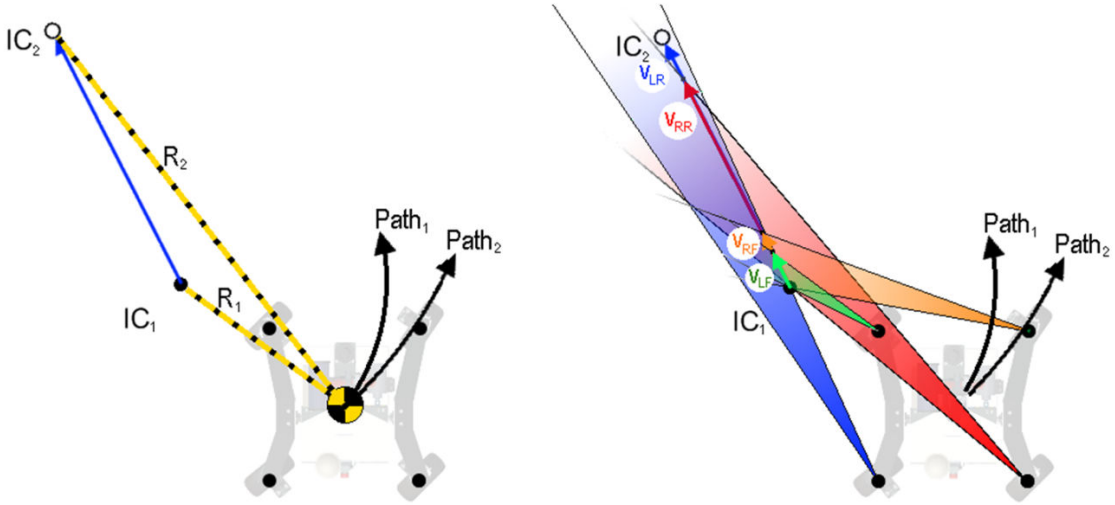

Fig. 42. Example transition is shown where the vehicle moves from a one drive-state, Path1 characterized by R1, to a second drive-state, Path2 characterized by R2, with transition-vector IC1-IC2 (left). The step limitation for each wheel is mapped showing the Left-Front wheel to be the limiting component (right). Notice the maximum dimension of the transition vector does not leave the 'one-step region' of the left-rear wheel.

To ensure that every incremental IC-path terminates present-IC coincident on the target-IC, the step-magnitude calculation is conditioned such that if the transition vector does not extend beyond the 'one-step region,' the original dimension of the transition vector is reported. The magnitude calculation is repeated for each wheel, as shown in figure 42, and the shortest step magnitude is taken to be the step magnitude of the transition vector.

Before moving on, it is worth noting that the 'set up' is not a 'rigorously defined' behavioral system in that it holds a type of a state-estimating dynamics model. The construction still qualifies as behavior because due to its strict feed-forward nature, and its reliance on enable-and-inhibit triggering from external functions as discussed later.

Because the system requires the processing load associated with holding and evaluating a type of a world mode, some of the advantage of a behavioral-control system is lost. This is, however, compensated for by the system's extreme insensitivity to geometric variation and on-the-fly system reconfiguration. While Ψ is not equipped with movable limbs, the implications to more general systems can be seen in this demonstration.

Behaviors of the Drive-Normal Mode

The final step of the drive-normal mode is translating the position from the IC-transit behavior into a set of servo commands, which is done with a series of behavior functions

- **Normal-Axis Selection Behavior**

Planar motion is redundantly satisfied with the AWDAWS system, so a wheel heading alone leaves ambiguity in the command specification. In general, there are two positions for each wheel that will satisfy a given drive-state specification, thus the wheels require angular control through ±180. To avoid complicated decision making, a ‘normal-axis select’ behavior is implemented. The normal-axis is defined as a line that is perpendicular to the wheel heading and passes through the IC and wheel’s pivot point. This behavior ‘watches’ the sign of the IC’s y-axis, and sets the sign of the normal-axis for each wheel, which translates to the sense of the wheels heading

In a more general control system, such normal-axis specification is not viable if long-wheel reorientations periods are to be avoided. In such a general system, the expedience of a maneuver would rely on comprehension of IC-path evolution and actuator range-of-motion constraints in order to appropriately specify the normal axes.

- **Velocity Generation**

Once the *normal-axis selection behavior* has used the location of the current-IC to define the desired drive-state, the servo command requires only a velocity for each wheel to be calculated. The location of the IC parameterizes a two-dimensional path, so only a rate, velocity or rotational, remains to be specified. To hold to a vehicle-centric, rather than a global, frame of reference, a tangential velocity is used as a rate specification. Since the tangential velocity is linearly scaled with path radius, the largest, farthest wheel from the IC receives the direct velocity command from the user. The commands to the four wheels are scaled with coefficients $c_{v\text{-}i}$, where *i* denotes the individual wheels, which are evaluated as the ratio of wheel path radius to the radius of the extreme wheel path. These velocity coefficients are evaluated as

$$\vec{c_v} = \frac{\overrightarrow{R_{IC}}}{R_{IC-\max}} \tag{13}$$

where the vector $\vec{c_v}$ is the velocity coefficients expressed in vector form, $R_{IC-\max}$ is the radius to the extreme wheel, and $\overrightarrow{R_{IC}}$ is the vector of radii for the individual

wheels. When the vehicle drives in a straight line, the difference between the radius of the extreme wheels and the other wheels is very small, and the velocity coefficients approach unity.

- **Velocity Limiting Behavior**

To handle velocity commands in off-nominal driving configuration, a velocity-limiting function is used. When switching between modes, or transitioning from one drive-state to another, it is common for the position actuators to pass the drive system through non-drivable states. During such 'uncontrolled' periods, passing velocity commands to the drive servos will result in poor performance, and possible damage to the mechanical structure. Thus, a velocity-limiting behavior is used to transition the system into and out of the non-drivable state. The errors for this behavior are easily obtained as they are byproducts of the above-mentioned closed-loop control.

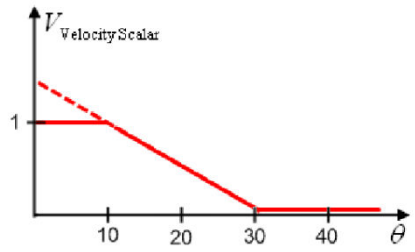

Fig. 43. Mapping function for the velocity-limiting behavior.

The variation in position error maps to a velocity-scaling coefficient with range zero to one, similar to the construction used by Emery, Balch, et. al. [35] As seen in figure 43, the mapping is a piecewise continuous function, such that when the

position error is greater than 30º, the velocity command is completely suppressed. As the error decreases between 10º and 30º, the suppression is linearly reduced, until at 10º, the full magnitude of the velocity command is passed. Mathematically, the function is expressed as

$$V_{\text{Velocity Scalar}} = \begin{cases} 1 & , e < 10° \\ \dfrac{4-\theta}{3} & , 10° \le e \le 30° \\ 0 & , e \ge 30° \end{cases} \quad (14)$$

where $V_{Velocity\ Scalar}$ is a coefficient to the velocity command. This behavior is primarily a safety feature that prevents the system from operating in off-nominal configurations. It also controls the ramp up and ramp down of the velocity as transitions between drivable and non-drivable states. Thus jitter and unnecessary impulse loading to the frame is avoided.

- **Position Trimming Behavior**

The steering motor and gear-train, which were selected to serve mechanical design requirements, were found to have insufficient pointing accuracy due to mechanical slop in the linkage and backlash. The lack of mechanical pointing precision resulted in poor tracking and excessive loading to the frame shown in figure 44. These loadings were exacerbated by changes in steering geometries, resulting from the axial deformation along the length of the rocker arm's box-beam structure and outward splaying of the wheels.

The key to the backlash compensation is found in understanding the structure of the rocker arms themselves. A basic mechanical analysis shows the structure more capable of carrying an inward side loading on the wheel. Thus backlash compensation is accomplished by trimming the position of each wheel with an inward-heading bias with respect to the direction of motion.

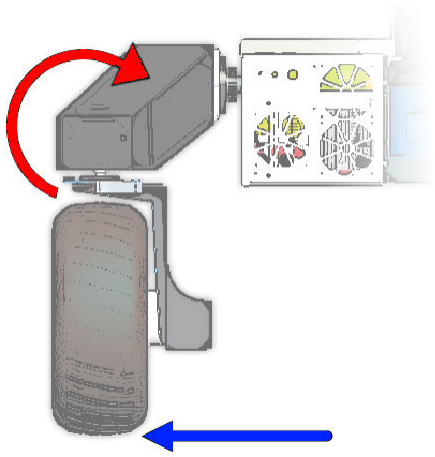

Fig. 44. Outward side load (blue arrow), and detrimental moment acting on the rocker arm (red arrow).

The trimming behavior simply reads the direction of motion and adds preset, calibrated trim values to the wheel position commands, as illustrated in figure 45.

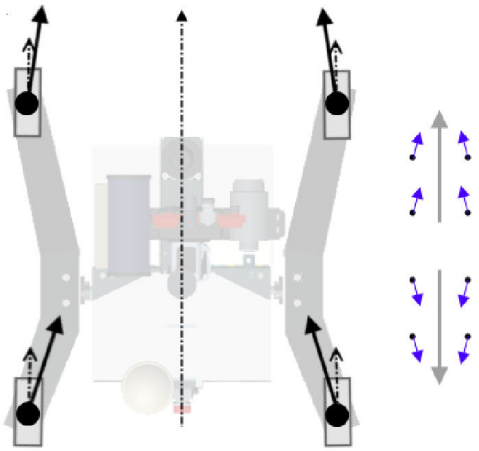

Fig. 45. Illustration of the trim corrections for the drive-normal mode (left), and the drive-normal cases with their corresponding trim corrections (right).

The calibration of the trim values is based off the home reference definition of the steering actuators, which is defined in the calibration sequence. As mentioned above, the calibration sequence operates at the level of the closed-loop control, and is calibrated for stable forward driving. The calibration involves first setting the initial reference of the position actuators with a ‘home-reference’ finding routine that is executed at the start of the robot operation. A measure of subtlety is introduced in the initial tuning of the routine by the need for agreement between the base calibration and the backlash trim calibrations of the different modes.

In practice, this calibration is done by first performing the home-reference find routine whose settings are iteratively determined with the normal-drive forward trim enabled. The selection of the forward-trim values has bearing on all other trim modes, so the entire process is executed several times in an iterative

fashion. Once the system can reliably track in a forward direction, the backward trim values can be estimated. From this point, the translate-and-rotate trim values can be experimentally determined.

- **IC-Path Planning Behavior**

The IC-transit behavior gives the current-IC the ‘desire’ to converge directly on the target-IC point. At present, the only implemented IC-trajectory-modifying behavior handles the case of reversals in vehicle path curvature. Future behaviors might, in a more general controller, read implications from the user’s input, avoid characteristic weaknesses of the system, and interact with the world by modifying the vehicle path based on sensor inputs.

The reverse path behavior responds to the rotation command input by remapping the target-IC with a trigger-activated, way-point mechanism illustrated in figure 46. The geometry-triggered state machine nominally rests in the move-to-target state in which the IC-transit behavior operates without interruption. When a sign change is detected between the y-axes values of the present-IC and the target-IC, a trigger sets the ‘move to p-side waypoint’ state active. In this state, the target-IC is suppressed until the present-IC reaches the p-side waypoint, which is asserted at the edge of the IC-plan on the side of the present-IC. When the present-IC is coincident with the p-side waypoint, its motion is interrupted as it is moved directly to the t-side waypoint, located on the extreme opposite side of the

IC plane. From the t-side waypoint, the system is returned to the normal move-to-target state.

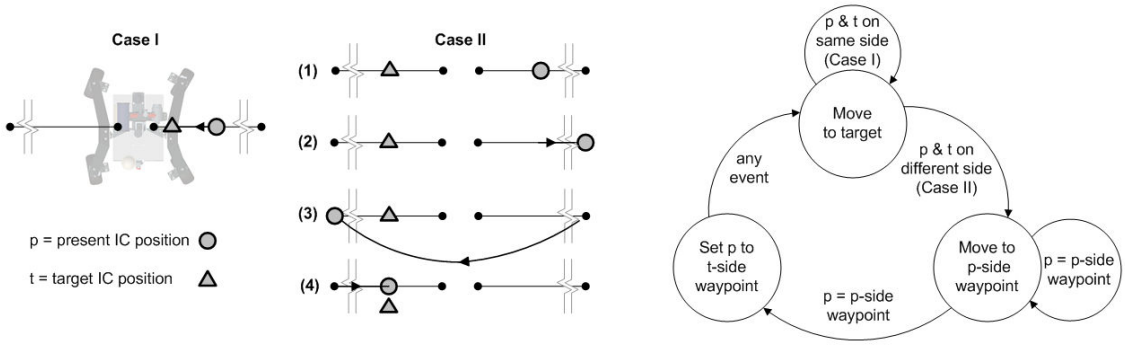

Fig. 46. Normal operation illustrating present-IC's tendency toward the target (left), IC transit sequence for the path reversal case (center), and path-reversal behavior state transition diagram (right).

Rotate Mode

The rotate mode is a constrained version of the drive-normal mode, in which the IC is fixed at the origin of the frame. With a fixed IC position, there is no need for a wheel selection or an IC-transit behavior; the wheel headings and directions are statically defined.

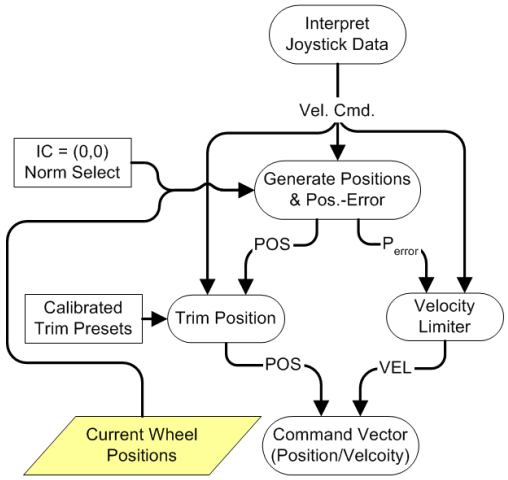

Fig. 47. Componentized flow chart for the rotate mode.

As a result of the symmetric drive system, the user-input mapping is trivial. Aside from the standard sensitivity mapping discussed later, the velocity of the wheels is directly proportional to the input of joystick's z-axis. The velocity limiting behavior from drive-normal mode is reused to ensure drivability of the system before a command is given to the drive servos.

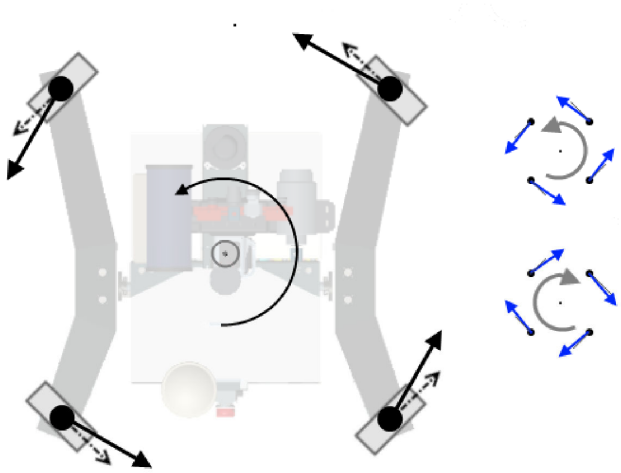

Fig. 48. Illustration of the trim corrections for the pure rotation mode (left), and rotation trim cases with their trim corrections (right).

The rotate mode also makes use of the drive-normal mode's trimming behavior scheme, with the slight variation that the trim values are identical in sign and magnitude as a consequence of the vehicle's symmetry. As shown in figure 48, the sign and convention of the trim value is established in that it follows the sign of the joystick input. The calibration of this value is developed empirically based on performance observations.

Translation Mode

The approach taken in the translation mode is different from the two preceding modes in that a more simple interpretation of the problem space was considered. An elegant solution to the case of pure translation on the IC-plane was not found, so a more straight-forward view of the problem was adopted in which translation heading maps directly to a wheel heading as is outline in figure 49.

This change in approach restructures the problem such that, aside from the input mapping, the first element to be calculated is the wheel trim values. These trims are added to the vehicle heading prior to an error calculation, which differs slightly from the previous cases. The error calculation is then used to evaluate the drivability of the system as in the previous modes.

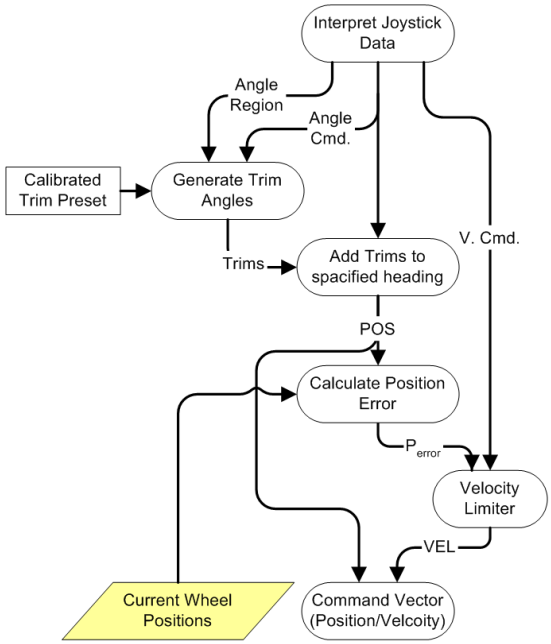

Fig. 49. Componentized flow chart for the translate mode.

- **Input mapping**

The inputs for this mode consist of heading and rate, which are read from the joystick's x and y displacements. The heading and velocity are separated by a Cartesian to polar transform, such that the velocity is proportional to the joysticks total displacement from the neutral position, and the heading is simply the direction of the joystick. From this point, the velocity is mapped directly to the wheels and the direction is used to set up the trimming function.

- **Trimming**

The trimming function is a bit complicated for this mode because, as illustrated in figure 50, the trim must always keep the wheels 'toed-in' to the direction of motion of the vehicle if the slop in the steering is to remain in check. In the case of a 45° diagonal translation, two wheels are in perfect alignment and the trim assignment for the aligned wheels must be nil.

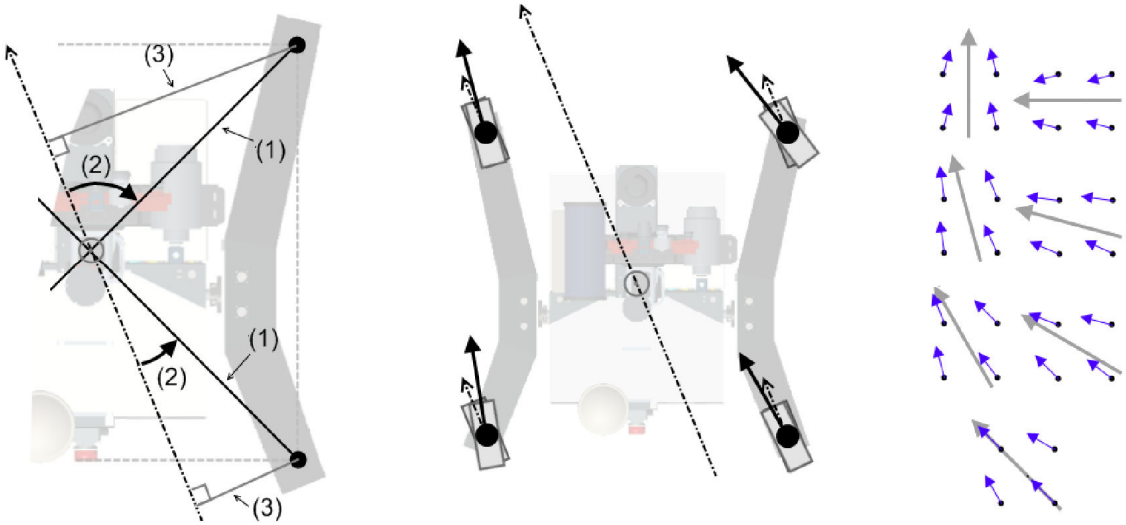

Fig. 50. Illustration of the trim algorithm for the translate mode with items (1) trimming reference datum, (2) trimming reference angle as referenced to the direction of travel, and (3) trim magnitudes (left). Translation heading (the long dashed line) with corresponding exaggerated trim corrections for each wheel (center). Selection of translation headings and their trim corrections (right).

The key to answer both of these cases relies on a constructed direction-of-motion vector that passes through the coordinate origin of the vehicle. From this vector, a measure of deviation from the heading is calculated for the individual wheels as

$$Trim_{wheel-i} = \sin\left(\angle DatumRef - \angle DirectionOfMotion\right) \tag{15}$$

where the datum reference angle is the angle described by the line between the origin of the coordinate frame and a given wheel. The resulting value, $Trim_{wheel-i,}$ is a per-wheel measure ‘trim need’ that is bounded between zero and one. An added benefit to this approach is that the magnitude of trim value is periodic on

180°, so the direction and heading axis is irrelevant, which eliminates the need for a messy case-sensitive trim definition.

The $Trim_{wheel\text{-}i}$ values are treated as coefficients to an angular trim specification. As with the rotate mode, the calibration of the exact trim value is done empirically and cataloged as a simple lookup reference.

- **Managing the velocity input**

In the previous two modes, there is a strong notion that the wheels arrive at the specified state before velocity is applied to the wheels. In the drive-normal mode, the velocity limiting behavior is activated if the system fails to meet a prescribed sequence of drivable states. In the rotate mode, there is only one possible configuration, so the user's commands are executed immediately with only an initial delay experienced by the user as the system moves to the rotate configuration. In the translate mode, however, the user has the ability to change the state command much faster than the system can respond. Additionally, there is no controller managing the transition between states: the slew rate of the steering actuators is fixed at the level of the closed loop controller.

To accommodate this difference in structure, the velocity limiting behavior is performed in two parts based on two different error sources. The first error is with respect to the mean wheel direction rather than the input command. Because the vehicle is being commanded to perform pure translation, the system never leaves a drivable state once drivability has been reached. Thus, it is more practical to

execute the velocity-limiting behavior based on deviation from an arbitrary drivable state, than from the commanded heading.

The second error source, like the translate and rotate modes, measures the deviation between the command and actuator positions. The basic velocity limiting behavior is used here, but the value of lower limit is raised from 30° to 50°. This means the velocity command will monotonically increase from 0% to 100% of the velocity command as the steering error decreases from 50° to 10° off the commanded heading.

Both of these behaviors output a scalar with range (0,1) and are multiplicatively combined to give a user smooth transition between stationary reorientation and directional motion, while avoiding awkward translations during drastic reorientations.

Control of the Pan-and-Tilt Unit

To consolidate the control for all actuators to a single location, the Pan-and-Tilt unit is controlled from the cRIO. The consolidation required the development of a custom driver, which using the proprietary communication protocol defined by Directed Perception [62] for control of their line of Pan-and-Tilt devices. Unit relies on a serial RS232 interface, which is run on the cRIO Real-Time controller through the Virtual Instrument Software Architecture (VISA), a generic LabVIEW serial IO interface library. VISA operates in a session in which communication channels are defined, used, and terminated.

The design intensity of the device driver was kept to a minimum by restricting the desired Pan-and-Tilt functionality to three basic operations: calibrate, move to a position, and move at a given rate. The *move to a position* mode simply defines a vector instruction for the Pan-and-Tilt based on direct coordinate inputs. The vector instruction assembles a move to an absolute reference command, and assigns a maximum velocity. The *move-at-a-given-rate* mode commands the Pan-and-Tilt to an extreme coordinate, positive or negative, and parses the displacement of the joystick as a rate specification. When the value of the joystick input falls below a certain threshold, the instruction to the Pan-and-Tilt is replaced with a command to move to a relative displacement of zero, causing all motion to stop. Finally, instructions that govern the configure of the return data are included in calibration command that sent to the Pan-and-Tilt, The actuator position data from the Pan-and-Tilt is read by the device driver, which translates and publishes the measured servo positions in degrees.

In operation, the unit must be calibrated prior to use to activate the position limits, and avoid damaging the wire harness. Once calibrated, the *move-at-a-given-rate* mode is set as the default mode, which can be interrupted by a position command. The rate control is tied directly to the joystick, and, like the drive system, is interlocked to the trigger button. The x-axis is inversely tied to the tilt actuator, while the pan is commanded directly by the y-axis. This input mapping provides intuitive operation to a user viewing an image from a mast position imager: forward is down, left moves the picture left, etc. The *move-to-a-position*

mode receives its command vector via a combination joystick hat-switch and on-screen command inputs, which are described later in the report.

Communication Link (UDP)

The user commands are provided to the cRIO through three software components that are executed over an IP network between the cRIO and OCU. The first component handles the transmission of commands to the robot. The second is a loss-of-signal (LOS) process that monitors both the presence and the timeliness of the command data stream, and indicates the absence of good communication to the robot. The final component transmits the state of the system, which consists of wheel velocities and positions, the Pan-and-Tilt state, and the metrical power consumption data, to the user interface.

The selection of an IP network and a WiFi communication link between the onboard processors and the OCU, led to the use of either transmission control protocol (TCP) or user datagram protocol (UDP). The TCP option poses a liability in that it conducts lossless transmission of an ordered data-stream. In a command-and-control task, the data integrity is not nearly as important as timely delivery. This makes TCP a very bad fit because of the non-deterministic lag associated with verifying the communication channel. With the intent of optimizing the end-to-end reliability of the communication link, rather than making each of its elements reliable, as outlined by Blumenthal and Clark, [63] UDP is used as it devotes little attention to data loss in exchange for operating overhead

and a lower latency communication. The actual implementation involves generating a packet and sending to a specified port on the fixed IP address of the embedded cRIO controller and vice versa.

To compensate for the lack of reliability, the LOS monitor is implemented, which relies on a 10Hz switching pulse generated on the operator's computer and included in the robot-bound packet as a single bit that reverses state with every packet. The packet is decoded on the robot controller, where the switching pulse is read at 10Hz. If no change in the switching signal is detected in five consecutive cycles, the connection is assumed lost. The count to loss-of-signal provides a level of resilience against the expected intermittent nature of UDP. On loss-of-signal, a trigger is set which is interprets the control system with an immediate stop-motion command. When the signal reads three consecutive switch cycles, the LOS trigger is released and control is restored to the user. The count to acquire-signal is used to give a level of certainty that the connection to the controller has been reestablished with a level of reliability.

Returning the robot data to the operator's consol is much like the command sending operation. The main difference being that the operator's computer is not subject to the same level of configuration control as the embedded computer, i.e., an embedded controller has a predefined IP address. To avoid placing constraints on the IP address of the OCU, the command UDP packets contain the IP address of the client computer. When the robot controller receives this

command stream, it applies the OCU's IP address as the destination address for the packetized return data.

Real-Time Task Processing

With limited processing resources, it is important to optimize the load of the Real-Time processor. The main tool used to perform these optimizations is the LabVIEW timed-loop structure. This structure executes a process sequence against an execution deadline with an associated process priority. The priority defines the criticality of each of the parallel process, such that the available processor time is allocated appropriately. If the processor is unable to complete the specified tasks, the OS will allocate resources so that low-priority processes will be skipped or terminated before high-priority tasks suffer performance loss.

Determinism throughout the design of the different processes was the objective, but the goal was not universally reached due to state changes corresponding to the various operational modes and external timing constraints. The process priority was set by criticality, and then by processing cost. This philosophy is derived from the notion that the safety of the vehicle should come first. This is followed immediately by the priority of the core of the control functionality. In the implementation, the LOS monitor receives the highest priority followed by the operational modes of the system-controller. Pending the proper generation of actuator commands by the system controller, the closed loop control law is next in priority, followed by the packet management for the UDP channels. Aside from

the power management IO to/from the FPGA, which received the lowest priority of the all the timed structures, the remaining processes operate on external time constraints.

External timing constraints are imposed on all I/O functions. The incoming UDP stream's timing is dependent on message arrival, and the outgoing UDP stream is defined by a port transmission frequency. The Pan-and-Tilts RS232 serial communication has a transmission frequency specification that is set in the definition of the VISA session. A further timing constraint is imposed by a 'wait for message arrival' function that is used to read the return data from the device. Because of the external timing constraints further management of IO process timing results in loss of efficiency manifesting in degraded performance and higher processor load. Thus, the UDP and serial processing tasks are executed in standard loop-structures.

1.18.9 User Interface

The user interface carries out two functions: command-and-control of the system and display of sensory feedback. The interface allows control if the embedded processors, the sensor devices, the power-system, the rolling chassis, and the Pan-and-Tilt unit. Remote operation of the robot is enabled by the displaying the sensor data from the omnidirectional camera, the forward-looking camera, and the SICK laser range-finder. An integrated graphical display provides indication of the actuator states and vehicle orientation.

Sensor, Processor and Power Control

Because the sensors are operated through embedded computers, their management involves accessibility to the processor and control of the power system. The configuration of the individual sensors and their respective data is more or less fixed in the compellation of the device executables. The executables start on boot, and immediately roll into reading from the sensors and broadcasting data. Thus the accessibility needs of the sensors are limited to the occasional restart of the device executables.

Basic accessibility to the embedded LiPPERT processers is made convenient to a user with a front panel dialog, and a corresponding server-management VI. The panel gives the user shortcut access to the windows Remote Desktop Connection that open directly to the specific LiPPERT; the connection allows easy access to the desktop of the embedded Windows OS. Additionally, a LiPPERT status indicator is included in the user interface showing when the OS is up and running. The OS status light for each LiPPERT is implemented over a two-way UDP connection that communicates with a server-management VI. The client-server communication is implemented using the same system described in the communication link section.

Power to the computers and the sensors is handled by a panel of relay toggles that are displayed on the user interface. This panel operates all the relays of the

PDU, with the exception of those associated with the drive system, which are linked together under a single toggle.

Sensor Data Feedback

The sensor display, shown in figure 51, consists of four view-fields organized in a scheme designed to emulate an automobile dashboard. The principal view is the unwrapped omni-cam image, which offers a 360° field-of-view where the ±180° discontinuity at the edges of the scene is aligned with the rear of the robot. This display spans the width of the consol, much like a windshield of an automobile, and provides visual coverage of the vehicle's surroundings, as shown in green in figure 52. Centered directly below the 'windshield' is the forward-looking camera that is mounted on the Pan-and-Tilt unit, and whose field-of-view is shown in blue in figure 52. This camera earned its central location because it is by far the most intuitive to understand by an operator, and is therefore expected to receive the most attention. The innate paradigm of a directed gaze under direct control of the user engenders a sense of presence to the operator. The intuition surrounding this sensor is reinforced by mapping the joystick motions to the motion of the field of view, via the Pan-and-Tilt interface.

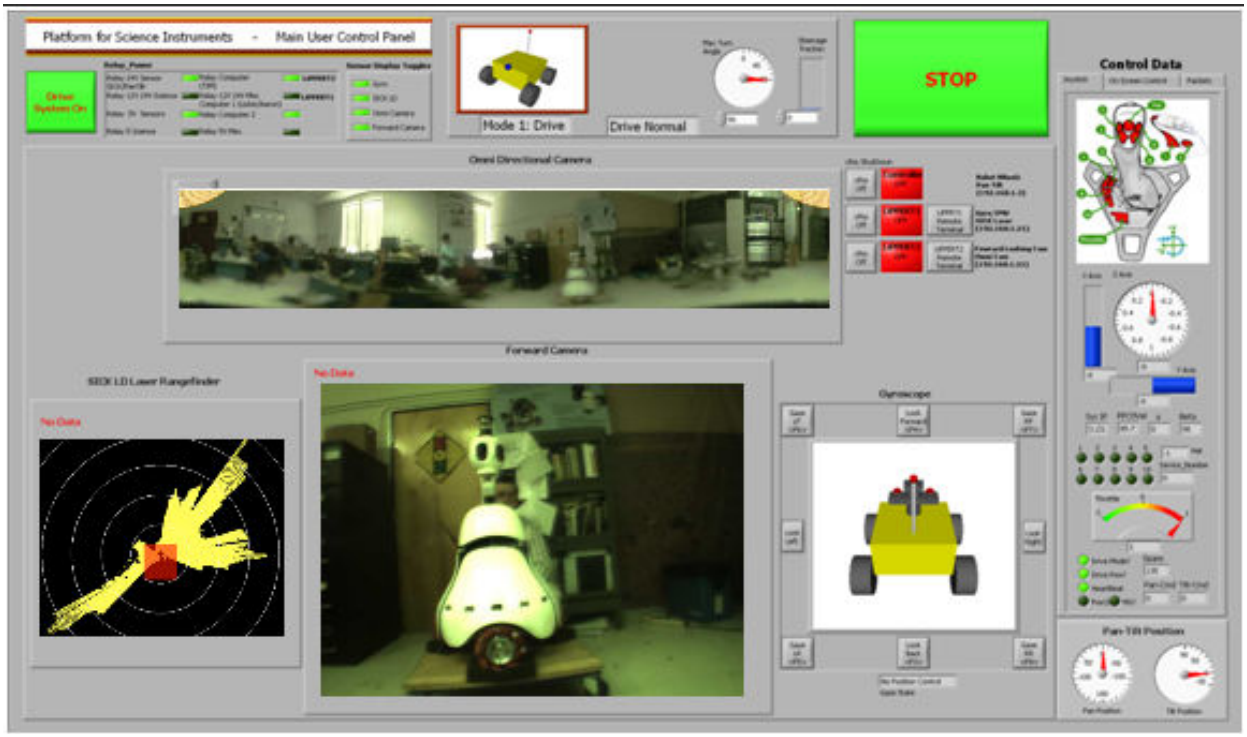

Fig. 51. Dashboard of the user interface.

In the lower left and lower right, in place of what might normally be engine gauges in an automobile console, the respective range data and the integrated graphical display are positioned. The range data display presents the readings from the SICK laser range-finder, as shown in figure 52 in red. The data is displayed from a top-down view, along with an outline of the robot chassis, for user orientation. The integrated graphical display is an animated figure that visually depicts the rotation components of the vehicle's orientation and actuators. The animated axes include the vehicle pitch, roll, and yaw; the angular positions of the four wheels; and the azimuth and inclination of the Pan-and-Tilt servos as is indicated in figure 52.

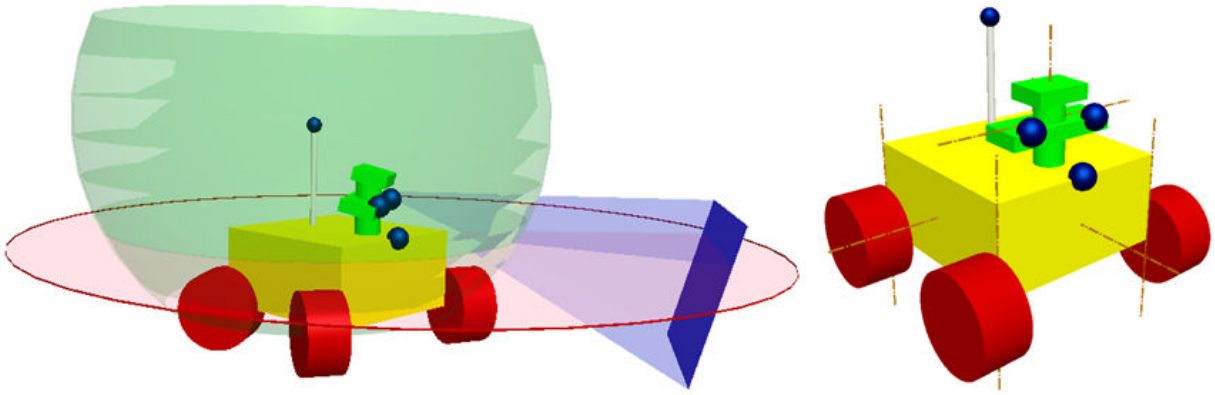

Fig. 52. Field-of-view for the omnidirectional camera, the laser range scanner, and the forward camera are displayed in green, red, and blue respectively (left); and pivot axes of roll, pitch, wheel angles; and Pan-and-Tilt azimuth and inclination are dynamically animated in the integrated graphical display (right).

As the sensor data is collected and processed by the host machines on the robot, it is published to the network shared-variables. The LabVIEW network shared-variables have a routing functionality built in so that data is not openly broadcast across the network. When a call is made to a shared-variable, the host sends the data directly to the IP address of the requesting client. This is important for transmitting the sensor data across the wireless link to the client computer.

The sensors themselves are capable of very high data rates, and are sampled a good deal below capacity, based on the minimum frame rate requirements of a remote user. However, even at this reduced data rate, the 802.11g connection cannot support simultaneous transmission of data from all sensors. To get around this problem, a time-domain, multiple-access construction is implemented with the use of the LabVIEW timed sequence structure. The structure allows for the sequenced execution of multiple sensor data-acquisition functions to be

completed by inside of an execution deadline. In the implemented front panel, the LiPPERT hosted shared-variables sensor data is accessed, post processed, and displayed to the front panel in a timed sequential fashion.

Processing the sensor data in this way, as it is displayed, has two favorable outcomes. First, in the event the data read takes too long, an execution deadline is reached, and the particular task in the sequence is skipped and attempted again on the following cycle. The more significant result is that the wireless network link remains open and usable, because only one sensor's data is being transmitted at any given time.

Command-and-Control

Primary control of the vehicle is accomplished with a single joystick and an on-screen command counterpart. The particular joystick used is a wireless ten-button, three-axis unit with a throttle on the base, and a thumb operated nine-position hat-switch as shown in figure 53. The joystick command of the system is carried out under two general modes that alter the function of joystick axes, such that a look-mode controls the Pan-and-Tilt, and the drive-mode controls wheel actuation of the motion base. As the Pan-and-Tilt is essential for the directing both the view of an operator, and the pointing of a remote sensing instrument, the joystick ‘hat switch’ gives the user eight typical gaze directions that can be accessed while in drive mode. In addition to the joystick input, front panel access

to the command values of the motion platform and the Pan-and-Tilt are made available.

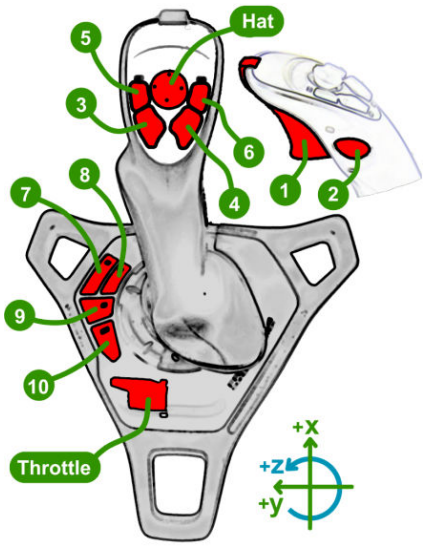

Fig. 53. Three-axis, ten-button wireless joystick with base throttle tab on base used to control the robot.

The look-mode provides the operator with analog rate control of the Pan-and-Tilt position. The mapping uses the x and y axes of the joystick to drive the inclination and azimuth servos of the Pan-and-Tilt respectively; the backward joystick motion corresponds to a positive inclination command. The y-axis is used to pan, even though it is more intuitive to control the azimuth of the Pan-and-Tilt with a z-axis rotation of the joystick when operating the Pan-and-Tilt under direct observation. When interacting with the robot from the heads-down vantage point of a Pan-and-Tilt directed image, it is most intuitive to ‘pan’ the image left and right with the y-axis of the joystick. To deliver precise control to the user, the magnitude of the axis displacements correspond to the rate of motion. To avoid

inadvertent joystick input, the trigger is used to verify intent from the user before sending commands to the actuators.

A second scheme of Pan-and-Tilt operation allows the user to select from eight preset gaze directions with the joystick hat-switch that guide the Pan-and-Tilt to the cardinal points of the robot. This ‘glance’ functionality is implemented with horizontal gaze directions (i.e., zero inclination) of directly forward, ±90° to the left and right, and directly backwards. The minor cardinal points direct the gaze, such that the field-of-view of the forward-looking camera is centered over each of the four wheels. In addition to the hat-switch, the gaze directions are accessible through an on-screen button interface, as shown in figure 54. The buttons are positioned around the perimeter of the integrated graphical display to intimate their function.

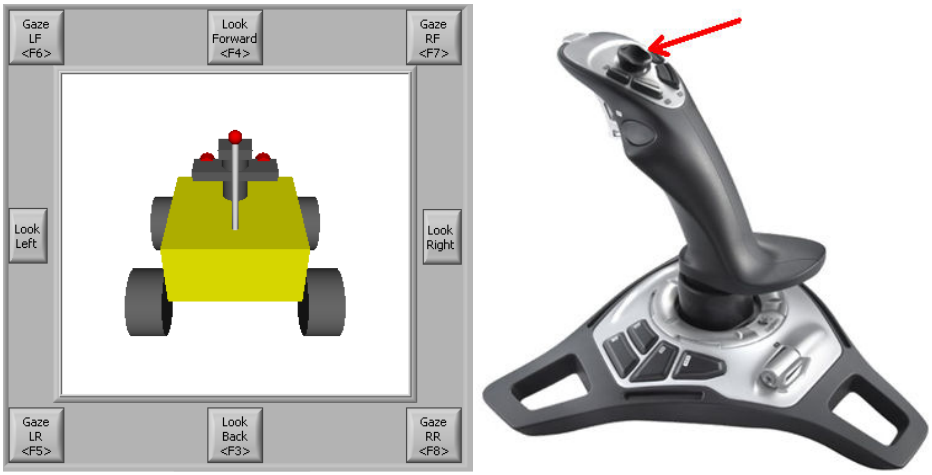

Fig. 54. Input for position control of the Pan-and-Tilt with onscreen interface (left) and joystick hat-switch (right).

When in drive-mode, the joystick takes on the behavior described in the operational modes section. The Drive Normal mode is the default mode, with the

z and x axes corresponding to the steering and velocity of the vehicle. The z-axis is used for the steering rather than the y-axis, because it is easier for a user to decouple the steering and the velocity inputs if both input axes are not linear displacements. The use of the translate and rotate modes are expected to be brief and intermittent, and are activated by depressing thumb buttons 3 and 4, respectively. When button 3 is depressed, the combined direction of the x and y displacements set the heading for the vehicle, and the magnitude displacement sets the velocity command. Button 4 enables the rotate mode, whose angular rate is specified by the magnitude of the z-axis displacement. In all drive mode configurations, the throttle key scales the maximum velocity output, but does not alter the scale of the steering/pointing commands.

An input panel shown in figure 55 is displayed to the user to aid in navigating the layered input modes. The panel gives graphical indication of the current input mode, drive or look, as well as the operational modes for the drive system. Also included on this panel are controls for the steerage-ratio and the steering angle-limit, which modify the behavior of the Drive Normal operational mode.

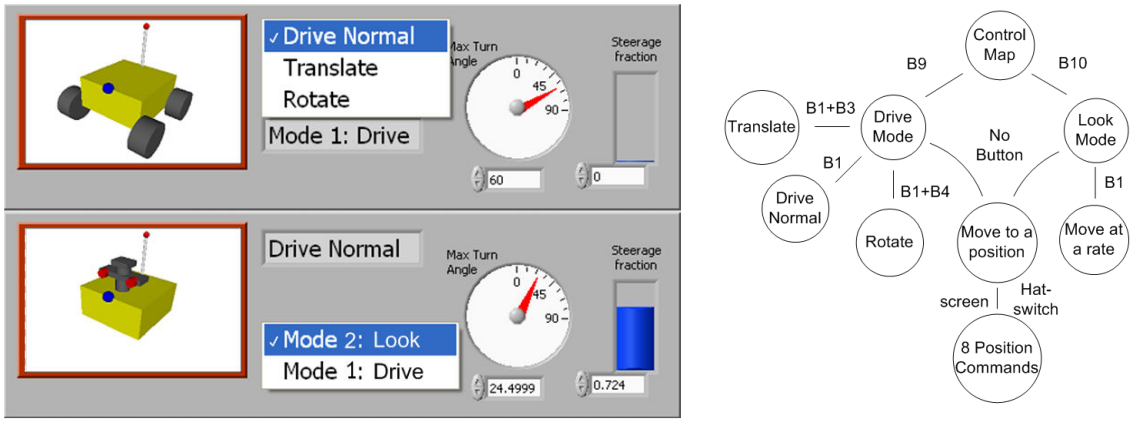

Fig. 55. Input mode indicator (left) and input mode (right) change mapping.

The final element of command-and-control that is delivered to the user is a three-pane panel positioned to the right for the screen, which serves mainly as a debugging tool. The panel gives a positive indication of the data being sent to and from the Real-Time controller. The panes, shown in figure 56, display the raw values constituting the UDP traffic and the exact data being posted to the joystick packet. If desired, a direct-input control is available in which the user can manually specify values.

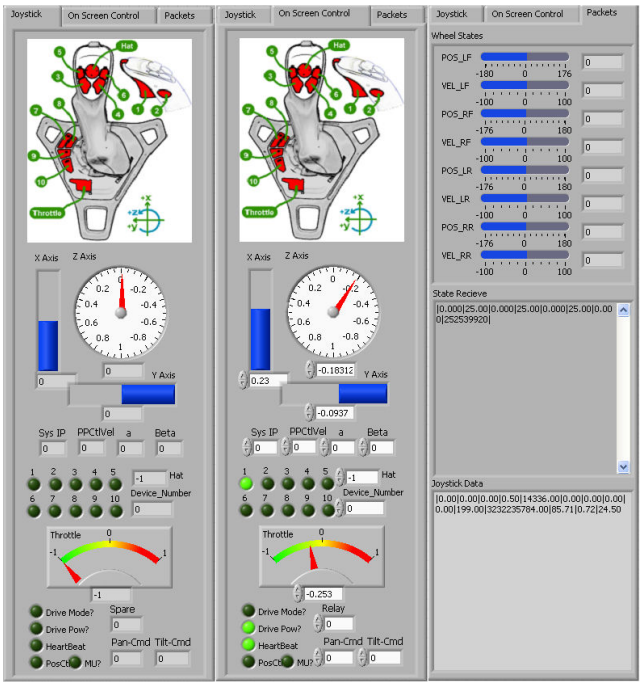

Fig. 56. Command-and-control access panel with a joystick data pane (left), a direct input pane (center), and a data received pane (right).

5. RESULTS AND SYSTEM PERFORMANCE

This section examines the primary elements of the as-built system and their performance as an integrated robotic system. The examination covers the performance of the software/computing architecture, the effectiveness of the mechanical system in operation, and the performance of the control system.

1.19 System Architecture

The networked computing architecture has proven extremely effective in that it allows for asynchronous and parallel development by multiple users. As the system is a platform for robotics research and experimentation, Ψ is under development a significant fraction of its existence. The considerations given to the differences between the needs of an operational demonstration/experiment, and the less glamorous needs of laboratory development, have played a critical role in the effectiveness of the system. During development, it was found that the benefits of an IP network could be expanded by connecting an external switch to Ψ's onboard network. This expansion allowed for multiple lab workstations and developmental PC104 embedded processors to be connected to the robot from elsewhere in the lab. Trivial as it may sound, the ease of access for testing of sensor and algorithm software, without the encumbrance of physically interacting with the robot, have proven invaluable. As various devices make their way to the robot, the burden of a shared power system and space is incurred. Nevertheless,

a high level of parallel development is still very achievable—even when the system is fully integrated.

While LabVIEW itself has been found to be somewhat fragile as a network-distributed computing framework, the value of a consolidated code base must not be underestimated. The use of LabVIEW executable files as sensor interface modules has proven immensely advantageous. The consolidation of the interface executables into a single file make moving the sensor interface between processors quite easy— connectivity issues aside.

Finally, in an attempt to document the success of Ψ's control architecture, the following sequence of events is related without claim or comment. In March 2008, a tour of the Robotics and Intelligent Machines Lab was given to National Instruments' Vice President of Development, Tony Vento, with particular emphasis on the use of the Compact Rio as the central element of an IP-based distributed robotic control system. In the April 2008 issue of NI News/Academic Edition, an article was published titled, 'FIRST Selects NI CompactRIO for Robot Control System.' [65 NI News] The agreement between National Instruments and FIRST Robotics (For Inspiration of Science and Technology) resulted in the use of a control system in the 2009 FIRST Robotics competition. The control system that was announced, which was eventually used on than 1500 high-school robots in an international robotics competition, carried a striking resemblance to the control system that had been implemented on Ψ.

1.20 Rolling Chassis

Field testing of the platform demonstrated good performance of the rolling chassis and drive systems. The mobility base was tested for a number of characterizations listed in table 1. The kinematic suspension allows for excellent tractions, which, as shown in figure 57, makes steep-hill and obstacle navigation possible. Reasonably fast actuator-response times make direct teleoperation practical. With minimal training, an inexperienced user can guide the vehicle with little thought to the function of the mobility base. As obstructions and extreme terrain are encountered, the operator must possess an increased awareness of the drive mechanism and operational modes to successfully navigate.

Maneuver	Condition	Measurement
Max ascent grade (experimental)	Grass slope	30°
	Loose dirt	29°
Max grade of decent & cross slope navigation	Validated	35°
	Theoretical	45°
Max right angle step navigation	Direct approach (90°)	12 in
	Indirect approach (45°)	18 in
Max speed (any direction)	Flat & level ground	1.7 mph
Time to rotate 360° in place	—	0.8 s
Max steering transition time (full forward to full turn)	—	2.8 s
Time to pure rotation from full forward	—	1.1 s
Transition time between full forward and 90° lateral translation	—	2.8 s

Table 1. Vehicle performance metrics

The most common failure occurs when the vehicle is maneuvered into a situation where the side of a wheel is driven into an obstacle, as an attempt is made to reorient the wheel. The position actuators do not possess sufficient control authority to shift the full weight of the vehicle in such a condition. In this case, a stall condition is reached. The servo position controllers are sufficiently robust that good positional control is returned to the user once the obstruction is cleared. Avoidance and extracting from such a situation is somewhat of an art, and is entirely dependent on the skill of the operator.

Fig. 57. Ψ moving down a 35° slope of extremely lose dirt (left) and navigating an 18-inch right-angle step (right).

In the design of the box beams of the rocker arms, the analysis failed to adequately predict the rotational deformations of the structure. This was due to an oversight in the specification of the load cases, where the wheel-pointing accuracy was not linked to the system steering loads. This was compounded by a lack of understanding of the role played by the mating tolerance between the steering shaft and its interface in the worm-drive gearbox. The reality was that

the greater source of mechanical slop was not from the anticipated gear-train backlash, but from the low-tolerance fit between the steering shaft and key. The slop was reduced some by getting custom keys made to match the precise dimensions of the shaft and gearbox socket. The ultimate correction to the misalignment problem, however, came from the trimming behaviors in the control system that took advantage of the non-uniformity in torsional stiffness of the rocker-arm beams.

The overall serviceability of the robot has proven excellent. As the vehicle has been taken to a number of demonstrations and tests, the removable leg interfaces have been exercised almost continuously without issue. The design attention given to maintenance has proven invaluable while working on the robot in the lab. The combined three-side access to the electronics enclosures, and the chassis' vertical servicing configuration, has significantly reduced the effort required to perform the routine wiring and computer modifications.

1.21 Integrated Control System

To evaluate the performance of the IC-transtion controller, three test cases were conducted. In each of these cases, the robot was cycled through a predefined manuever, while monitoring various parameters of the individual wheels. The experiment was designed to measure the performance of the combined closed-loop and IC-transit controllers, test the system's sensitivity to perturbations in the

step angle, and provide an overall qualitative sense of the control-system's function.

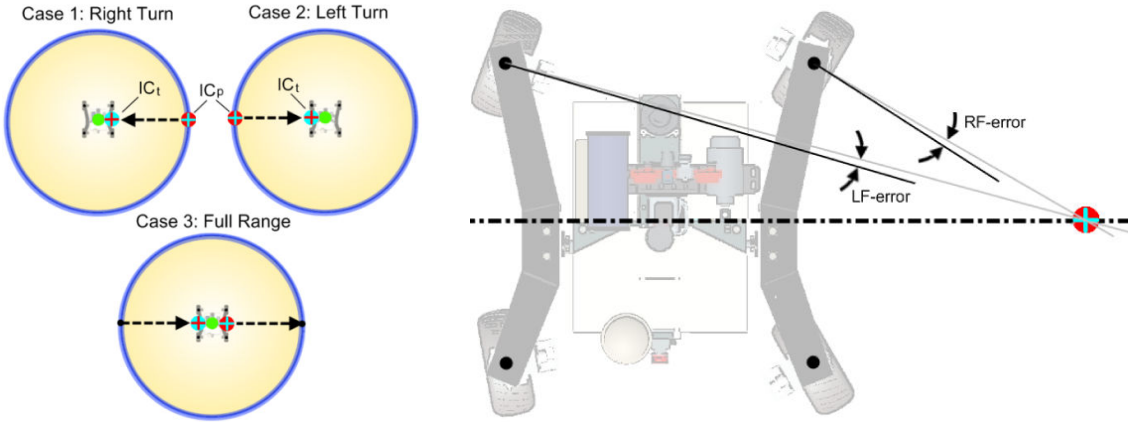

Fig. 58. Three cases used to evaluate the control systems performance (left), and angle measured as the system error for the front left and right wheels (right).

The maneuvers for the experiment, illustrated on the left of figure 58, commanded the robot to execute a full right turn, a full left turn, and a transition through the full range of motion from the full-right to the full-left configuration. The first two cases were used to evaluate the precision with which the controller could execute a basic transition maneuver. Both case one and case two were conducted to isolate effects of nonuniformity in the physical components. Both of the these cases were repeated for three different angular step limits in both while the robot was at rest and when it was suspended off its wheels.

Test code was written and run during these maneuvers that monitored the IC motion, the magnitude error of the actuators, and the commanded and measured wheel positions. The test was conducted with a unity steerage ratio and a 90° steering-angle limit. Because of the symmetry resulting from the steerage ratio,

measurements were only taken for the front two wheels, as indicated on the right of figure 58. A sample representative of the data taken for cases one and two is shown on the left of figure 59. The measurements taken over the full dynamic range of the drive normal mode are shown on the right of the same figure.

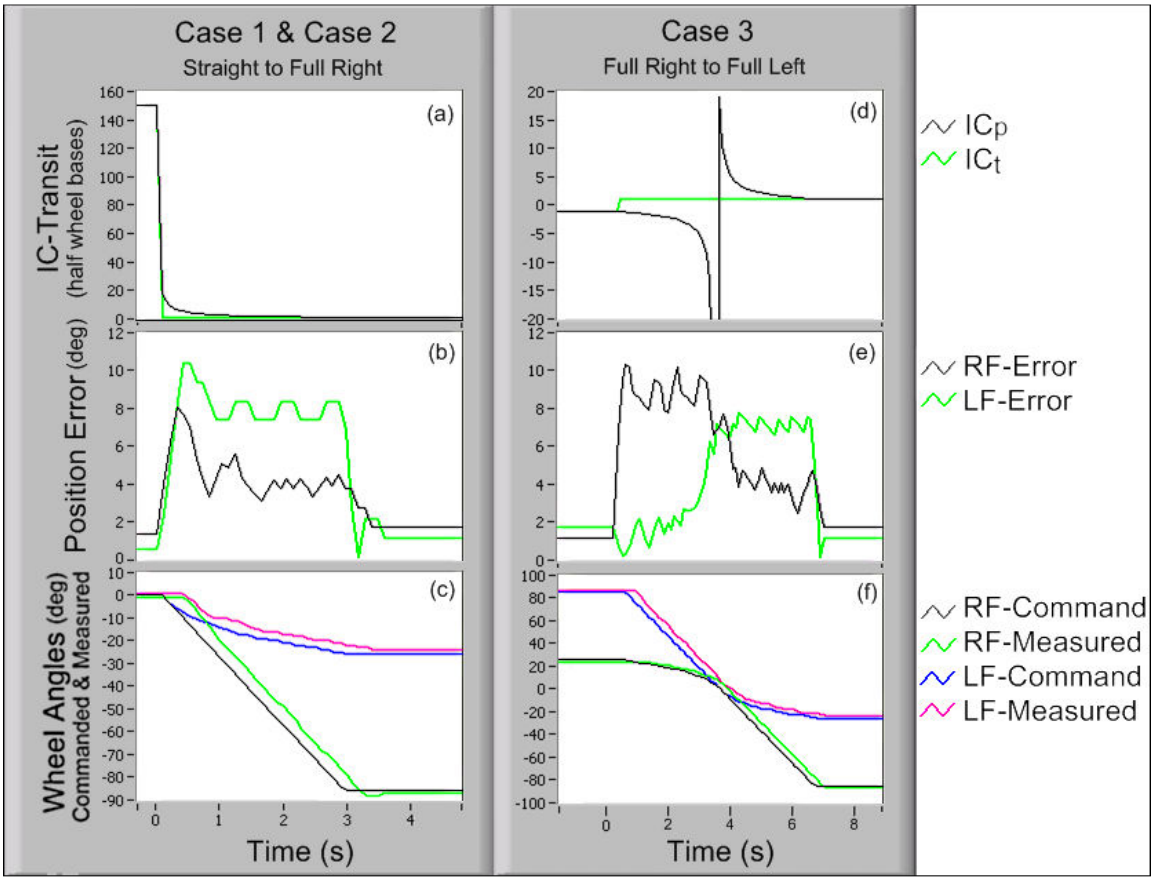

Fig. 59. Error measurements made to evaluate the performance of the IC controller.

In figures 59-(a) the function of the IC-transition control is clear in that the IC can be seen approaching the limiting radius R_{lim}. When exercising the full range of motion in case three, the R_{lim} value is particularly prominent in that it steps from one side to the other. As seen in figure 59-(c), the path-planning behavior takes the present-IC first outward to an infinite radius R_{∞} before closing on the target

from the opposite side. Plots (b) and (e) show the errors associated with several maneuvers. Plots (c) and (f) are included to give as a comparision between the actual commands generated by the IC-transit controlller, and the motions executed by the physical servos. These plots show clearly the characteristic overshoot, rise times, and transient and steady-state errors of the over-all control system. Figure 59-(f) shows the effectiveness of the IC controller in its ability to maintain a drivable state through a state transition while being restricted only by the actuator slew-rate. This effectiveness is evident by the continuity of the slope as the IC passes through the R_{∞}.

As seen in figure 59-(b) and (e) the control/drive system exhibited surprisingly large positional errors seeing that the system has proven itself effective in field testing and demonstrations. To understand these errors, a comparison of the peak error values for each of the 12 experimental tests are organized in chart format by test case in figure 60. The comparison reveals two interesting trends that give insight into the source of this error.

The first and most noteable feature is the proportionality between angular step limit and wheel error. This points to the possibility of further optimization, which involves matching the transition rate with the angular step limit and the process execution period, such that the per-step error is minimized. Recall that the angular step limit was experimentally approximated, and that the 10Hz process execution time was arbitrarily selected based on a user-interface design requirement.

The second feature of note is the discrepency between the right and left wheel error groupings and their reversal with turn direction. To understand at least part of this feature, recall that the angle calculation, using the dot-product, carries a significant role in the performance of the IC-transit behavior. The calculation relies on an arccosine to measure the angle between two points. As the distance of the points increases, the machine precisions becomes the limiting factor. As it is, the angle calculation is performed with 64-bits of floating point precisions to accommodate the very small angles that result at the radius of infinity R_{∞}. This machine error can be seen by the initial spike in error.

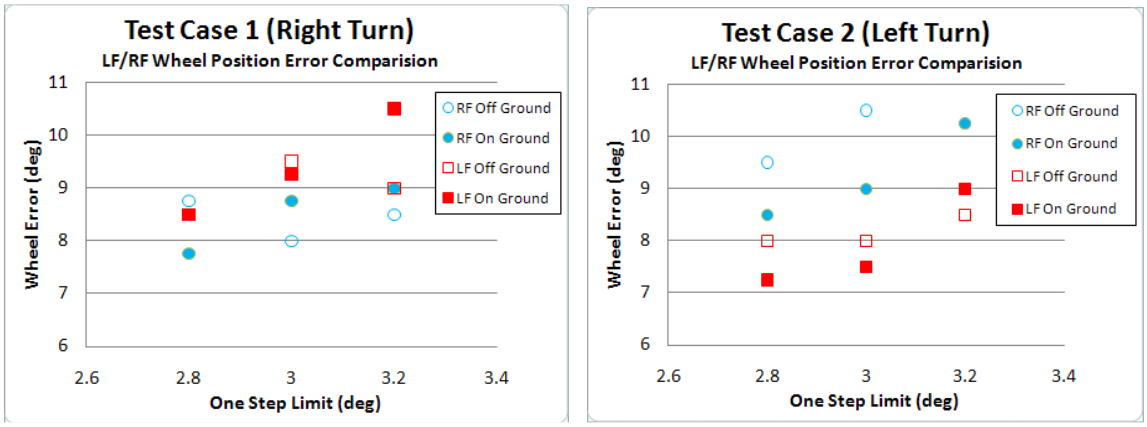

Fig. 60. Comparative errors for cases one and two for loaded and unloaded wheels with angular step limits of 2.8°, 3.0°, and 3.2°.

Other possible contributing factors are the approximate tuning of the PD controller, or the lack of optimization between the lower three levels of the controller: the FPGA signal processing, the Real-Time closed-loop controller, and the IC-transit controller. At more than one time during development, control system lag from cluttered communication channels has resulted in poor dynamic

response. A final system-level analysis and optimization effort was not performed on the digital controller, as it is not required to meet the basic mobility objectives.

To test adaptability of the control system, an experiment was conducted to give a level-of-fault tolerance to the control system. In this experiment, test code was written that reconfigured the orientation of the controller to accommodate complete failure of an arbitrary drive actuator. The assumptions of the experiment were that actuator failure would result in the inability of a single caster to reorient and to power the wheel, but still be able to be free to roll. The experiment was to see how much work it would take to modify the behavior of the control system to satisfy the constraints of a wheel fixed in an arbitrary orientation and AWDAWS drivability.

When framed in the context of the IC-transit controller, the solution was simply to cause the IC to reside on normal axis of the failed wheel. To used the vocabulary developed in the implementation chapter, the modificaton to the controller makes the normal axis of the failed wheel the effective line-of-zero motion. The experiment took a very simple approach where the IC was linearly projected onto the new line-of-zero motion as shown in figure 62.

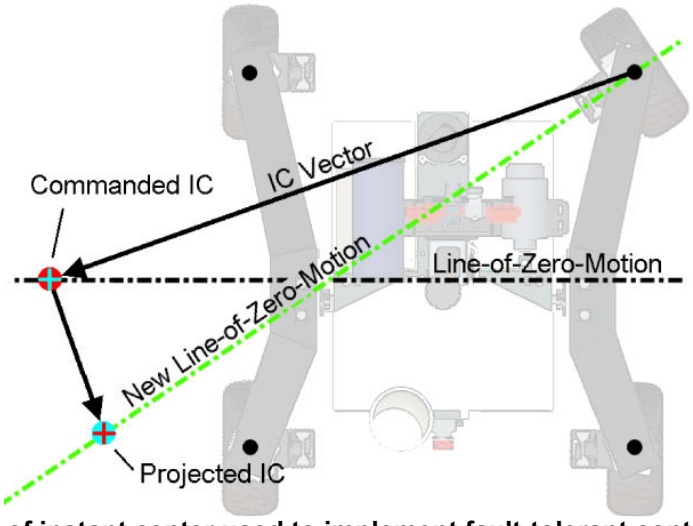

Fig. 61. Projection of instant center used to implement fault-tolerant controller modification.

For purposes of expedience, no attempt was made in the experiment to detect the wheel failure. The code was tested by first disconnecting the wheel, then telling the program which wheel had failed. As long as the wheel 'failed' within about 60° of the home position, the loss of system control was minimal to imperceptible to the user, aside from the obvious change in forward heading. Improvements to the failure compensator could be made by radially projecting the IC, but overall the experiment was effective in showing the adaptability of the control scheme. The entire process, from concept to implementaton, took less than an hour and yielded functionality that could effectively cope with limited system failure.

6. FUTURE WORK

Ongoing and future work summarily described in this section.

1.22 Control System

Future work on the control system will focus on reaching complete omnidirectional mobility based off a generalized instant-center control scheme. The development efforts will expand the capability of the drive normal mode to accommodate the full range of the IC-plain, as illustrated in figure 57. This expansion is expected to comprehend the nuances associated with the ergonomics of transferring and evolving the 3 DoF command of an operator's joystick to motions of an instant-center point on the IC-plane. Planning of these motions is believed to imply some level of artificial intelligence. Such an 'intelligent' algorithm must comprehend intonations of the user's input, the constraints of the vehicle, and the hazards/targets existing in the robot's environment.

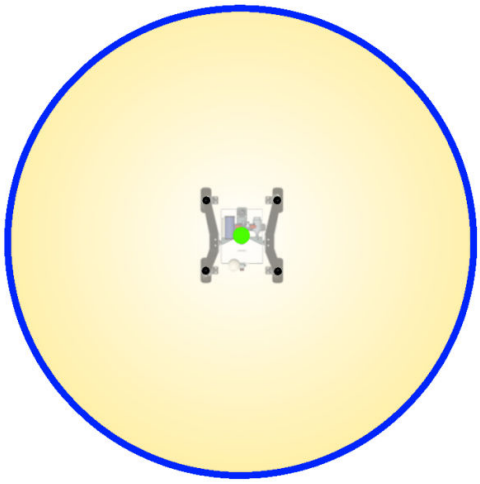

Fig. 62 Rotation (green dot), translation (blue circle), and intended allowable range of combined translation and rotation (yellow).

This instant-center based approach to control of AWDAWS systems is believed to be worth pursuing because it abstracts a component of the vehicle navigation in an unstructured real-world environment to a more controlled, path-planning problem situated in well-defined and tightly constrained virtual space. The preservation of basic physical geometries in the control system bears application to AWDAWS systems with variable wheel geometries, potential transparence in interactive calculation of actuator power consumption, and an ability to easily modify system behavior to cope with actuator failure.

1.23 Laser Multi-sensor

An effort is currently underway to miniaturize and demonstrate a Raman Florescence and Lidar Multi-sensor (RFLMS) instrument, while being remotely operated on-board Ψ. The RFLMS is being developed under Mars Instrument

Development Project (MIDP) as a candidate for future missions to Mars. The device, when implemented, will be capable of detecting minerals, organic and potentially biogenic materials, as well as atmospheric aerosols and clouds distribution, as illustrated in figure 58.

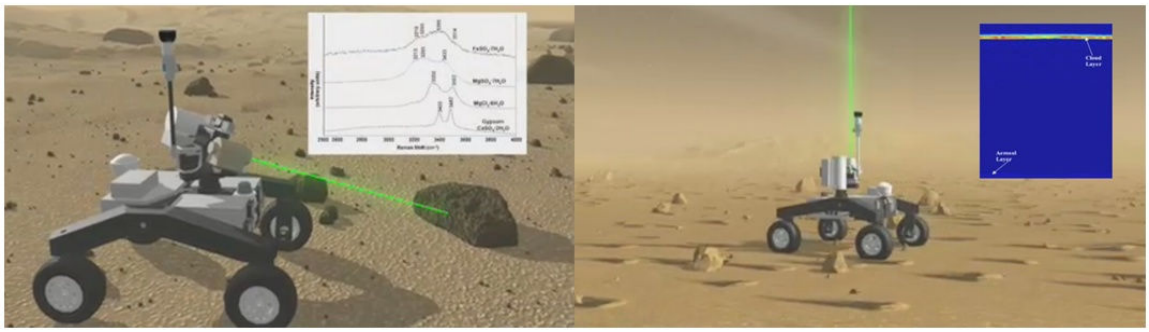

Fig. 63. Illustrations of a ground-based chemical measurement (left), and an atmospheric aerosol and cloud measurement (right).

The effort involves taking a prototype that was implemented as a feasibility study, and developing it into a mobile and self-sufficient system that will then be integrating onto the robotic platform for science instruments. This process involves a control system that will provide an interface to a custom laser control unit, a data-acquisition system, and a basic power distribution system for the various instrument components. The power, control, and data-acquisition systems needed for the demonstration will be housed onboard Ψ in a rear-mounted cabinet. The laser, optical devices, and sensing instruments will be mounted on Ψ's Pan-and-Tilt unit, as shown in figure 59.

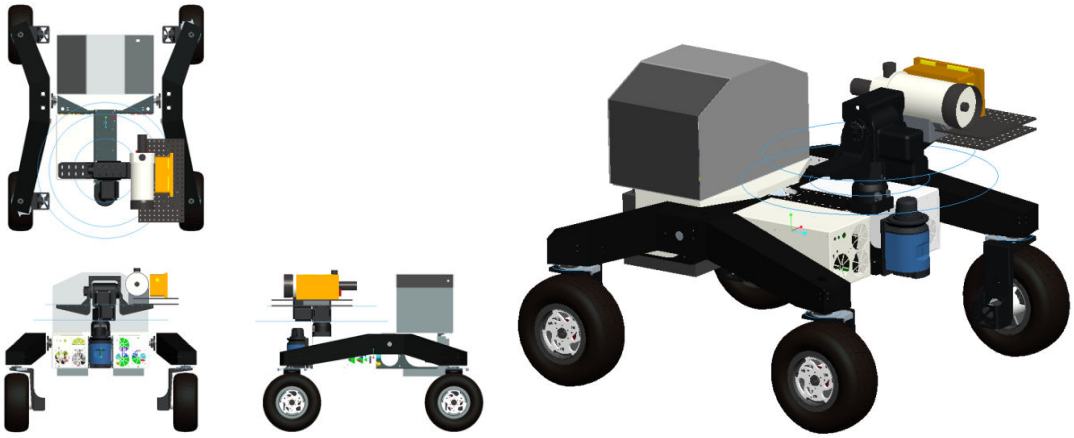

Fig. 64. Preliminary design from mounting the RFLMS instrument to Ψ.

1.24 Behavior-based Autonomy with Carbon Nanofiber Biosensors

A joint proposal between University of Tennessee, Oak Ridge National Labs, and Langley Research Center's Robotics and Intelligent Machines Lab has been accepted to proceed in the development of two separate, innovative concepts that will be combined to explore the possibility of searching for life with autonomous robots. The first concept is a sensor based on a technology that has been used to carry out electrochemical analysis to detect enzymes in biological material with the use of carbon nanofibers. The effort seeks to create an instrument, illustrated on the left in figure 60, which will demonstrate the feasibility of electrochemical sensing for investigating the presence of life in a given environment.

Fig. 65. Artistic rendering of biosensor and supporting equipment (left), engineering model of the arm in development (center), and artistic rendering of Ψ executing the demonstration with the biosensor (right).

The second innovative concept is the use of behavior-based navigation to explore a robot's environment in search of a science target, e.g., microbial life. The implications of having a competent and autonomous machine, whose actions are directed by the findings of a self-actuated science instrument, are tantalizing. The plan is to fix the biosensor to an arm attached to the front of the robot, as shown in figure 60, where it will be positioned to examine various targets. Both the science instrument and Ψ's sensor suit will provide the input to the behavior-based algorithm that will cause the rover to navigate terrain and obstacles in search of a science target.

1.25 Borehole X-Ray Fluoroscope

A proposal has been submitted to perform field trials of an X-Ray Florescence (XRF) spectrometer, which is being prepared as a Mars borehole probe. The instrument is capable of conducting elemental analyses along the walls of a

prepared borehole, which has significant implications toward *in situ* sub-surface planetary exploration.

Fig. 66. COTS X-Ray source and detector (left), and the flight qualified borehole XRF probe (right).

The XRF instrument consists of two parts: an x-ray excitation source and a detector, as shown on the left of figure 61. These two instruments have been miniaturized and integrated into a flight-grade configuration for borehole examination, as seen on the right of figure 61, The expectation is that the borehole probe, along with a drilling mechanism, would be fitted to the Ψ for a remotely operated demonstration of the instrument.

1.26 Immersive Virtual Human Environments

Studies are underway to explore ways in which immersive virtual environments can be used as an element in an ongoing initiative to enhance tools for interaction and public access of scientific data. The study seeks to explore ways of extending to the general public the experience of missions to the Moon, Mars, and other celestial bodies. The increase in accessibility through a rich 3-dimensional environment, populated with imagery and scientific data, is expected to engage the general public, while simultaneously giving academic researchers and analysts enhanced tools through which they can conduct their studies. An implemented virtual exploration environment may open the door for school students from around the world to pour over the vast repositories of data being collected from every corner of the solar system.

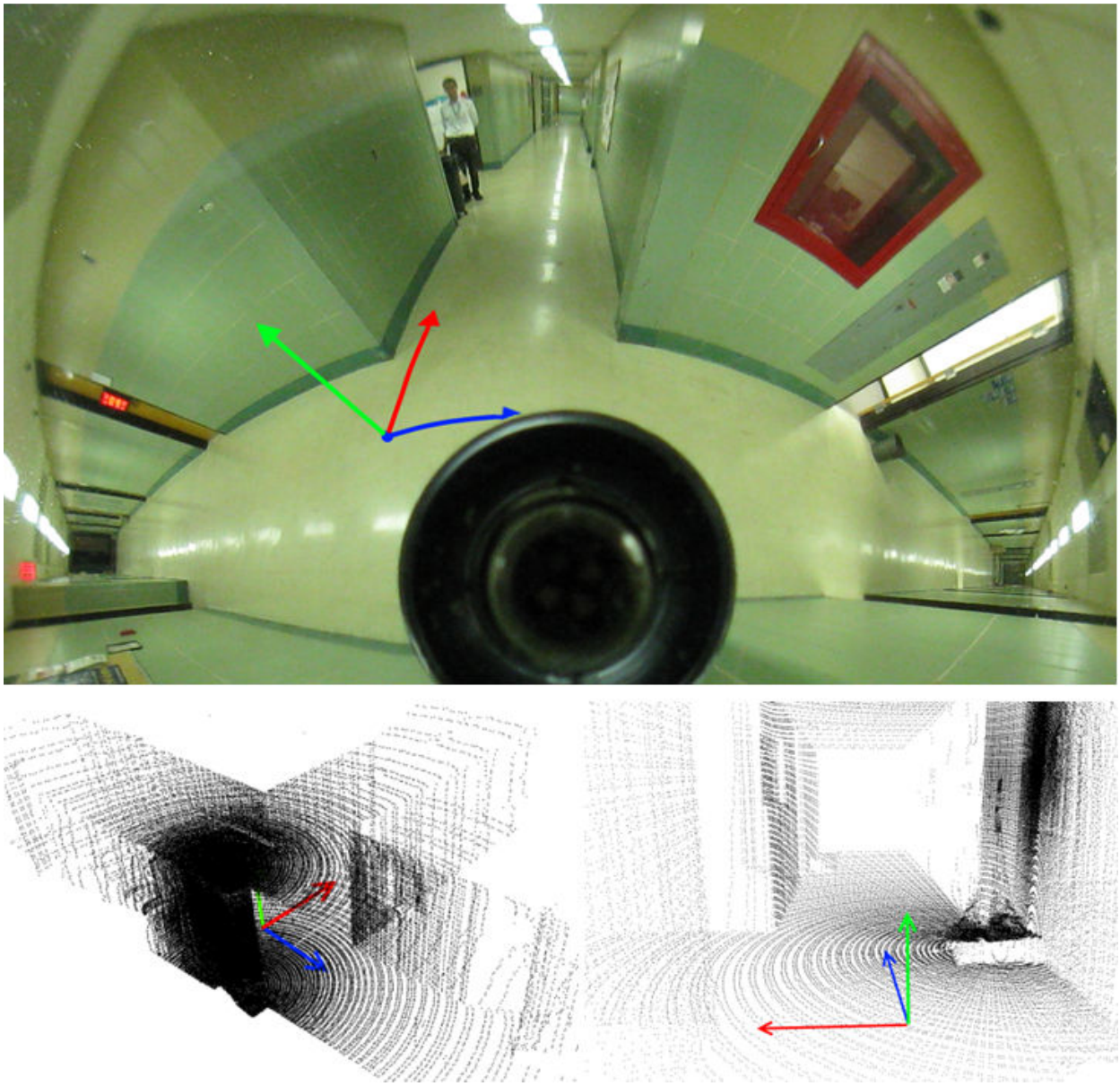

Fig. 67. Example of disparate data types for the same scene. The pivoting laser range-finder, which generates a cloud of points (bottom) and the omni-camera image (top).

Ψ has gained the interest of this Immersive Virtual Human Environments (IVHE) study as a sensor platform for collecting standard environmental data, and potentially representative scientific measurements. Using the remote data collection capabilities of the platform, an experimental demonstration is being

developed that is intended to explore end-to-end implications of a data-driven, immersive virtual environment that factually simulates a real environment. A physical survey and virtual reconstruction of a test-site, using Lidar generated range geometries, shown in figure 62, overlaid with visual and thermal images, along with data from a developmental science instrument, is expected to give insight into virtual exploration.

7. CONCLUSION

This thesis presented the development of a robotic system created in the Robotics and Intelligent Machines Lab at NASA Langley Research Center. The platform for science instruments (PSI or Ψ) presented herein is capable of supporting the demonstration needs of new and developing science instruments, as they seek to increase their technology readiness level between TRL levels two and four. These demonstrations are for the purpose of furthering various science technologies as they compete for membership in the sensor suits of future planetary surface exploration missions. Additionally, Ψ provides an outdoor platform from which to conduct research and experimentation in the areas of remote presence and autonomy.

To this end, a differentially suspended all-wheel-drive-all-wheel-steer (AWDAWS) robotic frame was designed, built, and tested. The conceptual design received inheritance from a number of successful robotic exploration and experimentation rovers, as well as an in-house robotic system. In a cost-savings effort, the vehicle was designed around COTS components, which were selected as a step toward finalizing the system design. The non-COTS portions of the system were rapidly constructed using modern prototyping technologies. The specification of the prototyped components was backed by parsimonious structural and thermal analysis that aligned with the ‘just-enough-design’ approach ascribed to in the overall system development. The resulting system met nearly every design

objective including obstacle navigation, slope climbing, and flat-ground transition speeds.

In contrast with prior art, the AWDAWS mobility base was controlled using a geometrically derived behavioral control scheme. The control scheme relied on the fundamental geometries of planar-motion, and the instantaneous center of rotation, to coordinate the motions of the drive actuators. The dynamics of the vehicle were captured in a geometric state-estimator. The estimator handled the transition from a given vehicle configuration, or drivable-state, to a different drivable-state of the operator's specification. The control by instant-center approach required the experimental derivation of mapping functions to convert the users input into their corresponding instant-center positions. The approach also required a set of behaviors that mapped the location of the instant-center to actual wheel commands. An elementary demonstration of the characteristic benefits expected from a behavioral control scheme was then demonstrated. The demonstration consisted of an implementation of an experimental algorithm that, upon failure of a given wheel, could quickly and effectively modify the sense the controller’s orientation with little effect to vehicle operation. Future development of the control scheme is expected to yield an omnidirectional controller for an AWDAWS vehicle that exhibits extreme responsiveness to the operating environment.

A flexible computing/software architecture was developed and implemented on the robot system. This architecture was based on IP networked PC104

processors, running Microsoft ® Windows XP Embedded and National Instruments' LabVIEW software. The architecture was presented as a reconfigurable robotic experimentation environment. With plug-and-play IP-based connectivity, reconfigurable stackable processor units, and portable software to interface to the various sensors, a satisfactory level of flexibility was reached.

To deliver a viable platform in support of a laser-based science instrument demonstration, a suit of four sensors was assembled, and a user interface was designed. The interface integrated remote control of the robot with dynamic sensory feedback. The sensor suit consisted of a rate-gyro, omnidirectional camera, a laser range-finder, and a targeting camera whose gaze was directed with Pan-and-Tilt unit. In preparation for a science instrument demonstration, the sensor system was integrated into the robot to allow intuitive 'heads-down' remote operation of the vehicle. A single command-and-control user interface was ported to a laptop-based operator control unit that communicated to the robot via a high-power WiFi link with down-range capabilities of up to a mile.

Ψ continues in its mission of robotic experimentation and science instrument demonstration as it is currently supporting two demonstration efforts that may experience mutual benefits as a result of the endeavor. The Raman Florescence and Lidar Multi-Sensor (RFLMS) is to carry out *in situ* and atmospheric measurements. The integration of the Raman spectral and Lidar data from the RFLMS instrument into the Immersive Virtual Human Environment study will amplify the quality of the study's results. At the same time, this presentation will

likely further the efforts of the RFLMS team by increasing awareness of their work and bringing their instrument to a broader audience. In the end, Ψ, in fulfillment of original vision, is making an ever-so-small, but none-the-less definite, contribution to the exploration of our solar system.

8. REFERENCES

[1] Wolfram Alpha. Wolfram Alpha LLC. Search “(earth land mass) (Mars surface area)”. 09 Oct 09 <www.wolframalpha.com> 09 Oct 09

[2] A. Bradley, S. Miller, G. Creary, N. Miller, M. Begley, and N. Misch. “Mobius, an omnidirectional robot utilizing mecanum wheels and fuzzy-logic control.” *Proc. 28th Annual AAS Rocky Mountain Guidance and Control Conference*. Breckenridge, CO, USA. Feb 2005: pp. 251-266.

[3] Sam. A. Miller, Mobius. “An Omnidirectional Robotic Platform and Software Architecture for Network Teleoperation.” Thesis for Masters of Science in Electrical Engineering, Electrical and Computer, Old Dominion University. May, 2005.

[4] B. E. Ilon. “Wheels for course stable self-propelling vehicle movable in any desired direction on the ground or some other base.” April 8, 1975. US Patent 3,876,255.

[5] S. Miller, A. Bradley, N. Miller, R. Hodson. “Trinity, an Omnidirectional Robotic Demonstrator for Reconfigurable Computing.” Advances in the Astronautical Sciences. Breckenridge, CO, USA. 2006: Vol 125, pages 145-162, AAS 06-096.

[6] R. Hodson, K. Somervill, J. Williams, N. Bergman, R. Jones. “An Architecture for Reconfigurable Computing in Space.” 2005 MAPLD International Conference. Washington D.C., USA. Sep 2005.

[7] A. Bradley, S. Miller, G. Creary. “Omnidirectional Robots for Off-Planet Applications.” Nippon Kikai Gakkai Robotikusu, Mekatoronikusu Koenkai Koen Ronbunshu, 2005: L0318B. pp 1P2-S-058,

[8] A. Bradley, S. Dubowsky, R. Quinn, N. Marzwel. “Enabling interoperable space robots with the joint technical architecture for robotic systems (JTARS).” Proc. ISAIRAS 2005 Conference. Munich, Germany. Sep 2005: ESA SP-603

[9] I. Nesnas, A. Wright, M. Bajracharya, R. Simmons, T. Estlin, W. Kim. “CLARAty: An architecture for reusable robotic software.” *SPIE Aerosense Conference*. Orlando, FL, USA. 2003:

[10] I. Nesnas, R. Simmons, D. Gaines, C, Kunz, A. Diaz-Calderon, et. al.. “CLARAty: Challenges and Steps Toward Reusable Robotic Software.” International Journal of Advanced Robotic Systems. 2006: Vol. 3, No. 1, pp. 023-030.

[11] S. Miller. “Innovation Report on Robot-Human Teaming” Records of the Robotics and Intelligent Machines Lab, Internal NASA LaRC. Oct 2007: **

[12] R. Munroe, S. Miller, A. Bradley. “Behavioral Mapless Navigation Using Rings.” Accepted to 4th International Conference on Autonomous Robots and Agents. Singapore, 2006:**

[13] S. A. Miller. “Statistical Image-Histogram Matching for Topological Robot Localization.” Final Report for ECE 651, Old Dominion University ECE department, Mar 2005:**

[14] W. F. Wood, A. T. Bradley, S. A. Miller, N. A. Miller. “Robot Self-Localization Using Omnidirectional Histogram Correlation.” Accepted to Intelligent Control Systems and Optimization. Milan, Italy. Nov 2008:**

[15] S. Miller. “Final Report: Stereo Viewing for Robot Teleoperation, Innovation funding report.” Internal NASA LaRC, Oct 2007:**

[16] J. J. Biesiadecki, M. W. Maimone. “The Mars Exploration Rover Surface Mobility Flight Software: Driving Ambition.” IEEE Aerospace Conference Proceedings. Big Sky, MO, USA. Mar 2006:

[17] G. E. Reeves J. Snyder. “An Overview of the Mars Exploration Rovers Flight Software.” IEEE Conference on Systems, Man, and Cybernetics. Big Island, HI, USA. 2005:

[18] H. W. Stone. “Mars pathfinder microrover: A low-cost, low-power spacecraft.” *Proc.* 1996 AIAA Forum on Advanced Developments in Space Robotics. Madison, WI, USA. 1996:

[19] R. Volpe, S. Peters. “Rover Technology Development and Infusion for the 2009 Mars Science Laboratory Mission.” *Proc.* International Symposium on Artificial Intelligence, Robotics and Automation in Space. Japan. 2003:

[20] J. Krajewski, K. Burke, C. Lewicki, D. Limonadi, A. Trebi-Ollennu, C. Voorhees. “MER: From Landing To Six Wheels on Mars… Twice.” IEEE International Conference on Systems, Man & Cybernetics. HI, USA. Oct 2005:

[21] Richard Volpe. “Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype.” The *Int’l.* Journal of Robotics Research. Jul 1999: Vol. 18, No. 7, pp. 669-683.

[22] P. W. Bartlett, D. Wettergreen, W. Whittaker. “Design of the Scarab Rover for Mobility & Drilling in the Lunar Cold Traps.” *Int’l.* Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS). Los Angeles, CA, USA. Feb 2008:

[23] K. Zacny, J. Wilson, A. Ashley, C. Santoro, et. al. “Geotechnical Property Tool on NASA AMES K-10 Rover.” Abstracts and Presentation at Joint Annual Meeting of LEAG-ICEUM-SRR. Cape Canaveral, FL, USA. Oct 2008:

[24] G. Ishigami, A. Miwa, K. Yoshida. “Steering Trajectory Analysis of Planetary Exploration Rovers Based on All-Wheel Dynamics Model.” *Proc.* of the 8th *Int’l. Symp.* on Artificial Intelligence and Mathematics. FL, USA, Jan 2005:

[25] B. Shamah, D. Apostolopoulos, E. Rollins, W. Whittaker. “Field validation of Nomad’s robotic locomotion.” *Conf.* for Mobile Robots XIII and Intelligent Transportation Systems. Boston, MA, USA. 1998: vol. 3525, pp. 214-222.

[26] T. Estier, Y. Crausaz, B. Merminod, M. Lauria, R. Piguet, R. Siegwart. “An Innovative Space Rover with Extended Climbing Abilities.” *Proc.* 4th *Int’l. Conf.* and *Expo.* on Robotics in Challenging Environments. Albuquerque, NM, USA. Feb 2000:

[27] Brandi Dean. “NASA's Newest Concept Vehicle Takes Off-Roading Out of This World.” NASA release: H08-068. Feb 2008: <http://www.nasa.gov/centers/johnson/news/releases/2008/H08-068.html>

[28] K. Tadakuma, M. Masatsugu, S. Hirose. “Mechanical Design of Horizontal Polyarticular Expandable 3-wheeled Planetary Rover: “Tri-Star3"." *Int’l Conf.* IEEE on Mechatronics and Automation. Jul 2005: vol.1, pp. 236-241.

[29] B. Wilcox, T. Litwin, J. Biesiadecki, J. Matthews, et. al. “Athlete: A cargo handling and manipulation robot for the moon.” Journal of Field Robotics. Apr 2007: vol. 24, iss. 5, pp 421-434.

[30] R. Brooks. “A robust layered control system for a mobile robot.” IEEE Journal of Robotics and Automation. Mar 1986: vol. 2, iss. 1, pp 14- 23.

[31] J. J. Bryson, (2003). “The behavior-oriented design of modular agent intelligence.” In R. Kowalszyk, J. P. Müller, H. Tianfield, and R. Unland, editors, “Agent Technologies, Infrastructures, Tools, and Applications for e-Services.” pages 61–76. Springer. 2007:

[32] Rondey A. Brooks. Flesh and Machines; How Robots Will Change Us. First Vintage Books Edition, New York, NY: Feb 2003, © 2002 by Rodney A. Brooks.

[33] Rodney A. Appendix. “How Genghis Works, Brooks, Flesh and Machines.” How Robots Will Change Us, First Vintage Books Edition. pp242, Feb 2003, © 2002 by Rodney A. Brooks.

[34] R Murphy, J Sprouse. “Strategies for searching an area with semi-autonomous mobile robots.” *Proc.* of Robotics for Challenging Environments. 1996: pages 15-21.

[35] R. Emery, T. Balch. “Behavior-Based Control of Non-Holonomic Robotics in Pushing Tasks.” IEEE *Int’l Conf*. on Robotics & Automation. Seoul, Korea. May 2001: pp 2381

[36] J. D. Crisman, J. Ayers. “Implementation Studies of Biological-Based Controller for a Shallow Water Walking Machine.” *Proc.* SPIE *Conf.* Mobile Robots VII. 1992: Vol. 1831, pp. 212-222

[37] Crandall, Karnopp, Kurtz, Pridmore-Brown. “Dynamics of Mechanical and Electromechanical Systems.” Malabar, Florida. 1985, © Krieger Publishing.

[38] T. B. Lauwers, G. A. Kantor, R. L. Hollis. “A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive,” *Proc*. IEEE International *Conf*. on Robotics and Automation. May, 2006: pages 2884-2889.

[39] M. Wada, H. Asada. "Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs." Robotics and Automation, IEEE *Trans.* 1999: vol. 15, no. 6, pages 978-989.

[40] M. West, H. Asada. “Design of a Holonomic Omnidirectional Vehicle.” *Proc*. IEEE *Int’l. Conf.* on Robotics and Automation. Nice, France. May 1992:

[41] H. Asama, M. Sato, L. Bogoni, H. Kaetsu, A. Mitsumoto, I. Endo. "Development of an omnidirectional mobile robot with 3 DOF decoupling drive mechanism." *Proc.* IEEE *Int’l. Conf.* on Robotics and Automation. 1995: vol. 2, no. 2, pages 1925-1930.

[42] O. Fiegel, A. Badve and G. Bright, et al. “Improved mecanum wheel design for omnidirectional robots.” *Proc*. Australasian *Conf.* on Robotics and Automation. Nov 2002: pages 117-121.

[43] R. Holmberg, O. Khatib. “Development and Control of a Holonomic Mobile Robot for Mobile Manipulation Tasks.” *Int’l*. Journal of Robotics Research. 2000: vol. 19, page 1066.

[44] M. Wada, S. Mori. "Holonomic and omnidirectional vehicle with conventional tires." *Proc.* IEEE *Int’l. Conf.* on Robotics and Automation, Apr 1996: vol. 4, no. 22-28, pages 3671-3676.

[45] D. S. Apostolopoulos. “Analytical Configuration of Wheeled Robotic Locomotion.” Ph.D. Dissertation at Carnegie Mellon University. Pittsburg, PN. 2001: © by Dimitrios S. Apostolopouslos. All rights reserved. CMU-RI-TR-01-08,

[46] M. Udengaard, K. Iagnemma. “Design of an Omnidirectional Mobile Robot for Rough Terrain.” IEEE *Int’l. Conf.* on Robotics and Automation. Pasadena, CA. May 2008:

[47] B. Gurol, M. Dal, S. M. Yesiloglu, H. Temeltas, "Mechanical and Electrical Design of a Four-Wheel-Drive, Four-Wheel-Steer Mobile Manipulator with PA-10 Arm," IEEE *Int’l. Conf.* Electric Machines & Drives Conference. May 2007: vol. 2, no. 3-5, pages 1777-1782.

[48] D. Wang, F. Qi. “Trajectory Planning for a Four-Wheel-Steering Vehicle” *Proc.* IEEE *Int’l. Conf.* on Robotics and Automation, Seoul, Korea, May 2001:

[49] B.C. Besselink. “Computer controlled steering system for vehicles having two independently driven wheels.” Computers and Electronics in Agriculture. 2003: vol. 39, pages 209-226.

[50] M. Makatchev, J. McPhee, S. Tso, S. Lang. “Modeling and Control of a Four-Wheel-Steering Mobile Robot.” IEEE *Trans.* on Automatic Control. 1999: pages 100–108.

[51] H. Xu, W. Huang, F. Peng, et al. “Maneuver Control of Mobile Robot Based on Equivalent Instantaneous Center of Rotation in Rough Terrain.” IEEE *Int’l. Conf.* on Mechatronics and Automation. Harbin China. Aug 2007” vol. 1, pages 405-510.

[52] E. Rollins, J. Luntz, A. Foessel, B. Shamah, and W. L. Whittaker. "Nomad: A Demonstration of the Transforming Chassis." *Proc.* of the *Conf.* on Intelligent Components for Vehicles. March, 1998:

[53] G. Ishigami, K. Nagatani, K. Yoshida. “Path planning for Planetary Exploration Rovers and Its Evaluation based on Wheel Slip Dynamics.” *Int’l Conf.* on Robotics and Automation. Italy. 2007:

[54] T.Bertl, T. Miller, S. Rock, J. Latombe. “Climbing Robots in Natural Terrain.” *Proc.* of IEEE *Int’l. Conf.* on Robotics and Automation, May 2003:

[55] T. Bretl, S. Lall, J. C. Latombe, S. Rock. Algorithmic Foundations of Robotics VI, Chapter: Multi-Step Motion Planning for Free-Climbing Robots, pp 59, Berlin Heidelberg: Springer-Verlag, 2005

[56] R. C. Hibbeler. Engineering Dynamics. Sixth Edition. Don Mills, Ontario Canada: Maxwell Macmillan Canada, Inc. 1992. © 1992 by R. C. Hibbeler,

[57] R. Siegwart, I. Nourbakhsh. Introduction to autonomous mobile robots. Boston, MA: MIT Press, 2004. © Massachusetts Institute of Technology.

[58]T. Mashita, Y. Iwai, M. Yachida. “Calibration of HyperOmni Vision based on Conic Curve.” *Conf.* on Machine VIsion Applications, Tsukuba Science. Japan. May 2005:

[59] R. A. Lindemann, C. J. Voorhees. "Mars Exploration Rover mobility assembly design, test and performance." *IEEE Int’l Conf.* on Systems, Man and Cybernetics. Oct. 2005: vol.1, no. 10-12, pages 450-455.

[60] M. Carroll, J. Johnson, J. Yong. “Design of Compliant Wheel for a Miniature Rover to be used on Mars.” Final Report. Univ. of Texas Austin. Fall 1991:

[61] Katsuhiko Ogata. Modern Control Engineering. 4th Edition. Upper Saddle River, NJ: Prentice-Hall 2002. © 2002 by Prentice-Hall Inc.

[62] Directed Perception Inc.. Pan-Tilt Unit Command Reference Manual. Burlingame, CA: Directed Perception, Inc., 2008. © 2008.

[63] M. Blumenthal, and D. Clark, 2001. “Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world.” ACM *Trans.* on Internet Technology. August 2001: vol. 1, no. 1, pages 70-109.

[64] National Instruments Corp., “FIRST Selects NI CompactRIO for Robot Control System.” NI News/Academic Edition, April 2008, <http://www.ni.com/news/academicnews_archive/ni_academic_news_apr08.htm>

** Available in the Records of the Robotics and Intelligent Machines Lab, Internal NASA LaRC.