MONTHLY REPORTING CHECKLIST

Submitted by: University of Miochigan Report Month: NOVEMBER 2002

TTP No.: ALO-7-C1-61 (UMichigan)

EARNED VALUE ANALYSIS DATA. *Check* ***one*** *box in Line A and* ***one*** *box in Line B.*

A. How is your project's schedule doing compared to your TTP baseline?

☐ 1. As planned. ☐ ☐

B. How is your project's total cost doing compared to your TTP baseline?

☐ 2. As planned ☐ ☐

PTS NARRATIVE INPUT CHECKLIST. *Check that you have prepared the following narrative inputs:*

☐ 1. SIGNIFICANT ISSUES/PROBLEMS/CONCERNS:
Note if there are any problems; otherwise, state "None." Report only problems considered "showstoppers" or fatal flaws (i.e., a lack of funding will cause the project to be shut down).

None

☐ 2. CORRECTIVE ACTION:
If a significant issue/problem/concern in Section 1 above is described, this section is required; otherwise state "None needed."

None needed

☐ 3. SUMMARY ASSESSMENT:
This should be a BRIEF paragraph summarizing the overall status of the project. This section is a synopsis of the entire report.

During this past month, work has continued in robust navigation, sensing, and radiation imaging. The novel OmniPede vehicle is progressing well, with a new innovation utilizing treads. Work using the Sick laser range scanner for map-building resulted in a better understanding of sources of artifacts from images. Work to upgrade an earlier generation gamma camera is proceeding rapidly with all 16 elements functional, and new efforts are exploring more efficient imaging techniques.

☐ **4. COST VARIANCE:**
If you checked number 4 or 6 in the Earned Value Analysis section, ***you must provide an explanation here.*** **Explain funding issues such as variances, carryover, commitments, incorrect FIS data. Avoid using only the words "Within budget." Some narrative is preferred.**

Within budget.

☐ **5. SCHEDULE VARIANCE:**
If you checked number 3 or 5 in the Earned Value Analysis section, ***you must provide an explanation here.*** **Note if the project is on schedule, ahead of schedule, or behind schedule. If behind, explain what is being done to bring the project back on schedule.**

On schedule.

☐ 6. TECHNICAL STATUS:

This is likely to be the longest section of the narrative and describes the technical accomplishments during the reporting period. Provide enough detail to inform, yet avoid extensive details that can confuse the reader.

6.1 Obstacle avoidance

6.1.1 Elevation Map-building with the Sick laser rangefinder (LRF)

This month we performed another series of experiments to refine our map building methods. In order to complete the data sets for a journal paper, we added five additional surveyed obstacle courses. In previous months we had already conducted meticulous tests on 17 surveyed obstacle courses. In all of the experiments, we moved the laser rangefinder, mounted on our roll-pitch-linear motion table, through the 22 obstacle course in four different modes: (1) translation only, (2) translation and roll, (3) translation and pitch, and (4) translation and roll and pitch. Roll and pitch rotation was limited to fixed rates of 37.5°/sec, while translation was fixed at 1 m/sec. The results obtained with the five new obstacle courses showed in all cases that our mapping algorithm, called Certainty Assisted Spacial (CAS) filter, outperformed the other, existing filters.

6.1.2 Artifacts and noise investigation

We ran further experiments under Windows XP (see section below) and verified that the dominant type of artifacts were caused indeed by an overflow problem in the 16-meter mode of the Sick laser rangefinder. We also confirmed that this overflow problem did not affect our earlier collected data sets, since we discarded in those sets all range readings in access of 8 meters.

With regard to the analysis of noise we ran experiments to try and isolate different causes of noise. However, these experiments were inconclusive. The difficulty in isolating causes of noise is exacerbated by the fact that not all physical functions of the Sick laser rangefinder are disclosed to the end user. We concluded that we will have to use the rangefinder “as is” and leave the further improvement of noise issues to the manufacturer. We should point out that the performance of the Sick laser rangefinder “as is” is excellent, and that noise is not a substantial problem.

6.1.3 Sick and Windows XP

Since the beginning of our work with the Sick laser rangefinder we were unable to use the vendor-supplied, Windows-based interface software. That’s because it did not run reliably at the Sick’s maximum interface speed of 500 kbps under Windows 98. Instead, we wrote our own interface drivers from scratch for DOS and Linux.

We now found that under Windows XP the 500 kbps interface works much better than under previous versions of Windows. This fact proved helpful in our analysis of noise and artifacts, since the vendor-supplied software provides additional information that our own interface drivers did not provide.

6.2 Position Estimation

During most of November we paused all research work on position estimation. This allowed our researchers and programmers to collaborate on the development of a unified software architecture and its object oriented implementation. This effort is described in more detail in the Infrastructure Section in this report, below. Research activities in position estimation resumed toward the end of November.

6.3 Novel mobility concepts

As reported in the last two progress reports, we have developed a break-through improvement to the OmniPede concept. This improvement, called OmniTread, is based on treads instead of legs to provide large and constantly moving surfaces all around the vehicle. For the time being we are continuing in parallel the work on the OmniTread and OmniPede (collectively called “Omnis”).

6.3.1 New OmniTread design

We finished dimensioning the main mechanical unit of the OmniTread, called a “Segment,” and we created a fully dimensioned 3-D model in a CAD program. We have also begun manufacturing the individual mechanical components that, once assembled, will make up the prototype segment. We further ordered the all components that will not be manufactured in house, such as gears, shafts, and bushings. We expect to have a prototype segment finished by early January 2003. We expect the prototype segment to be fully operational with a temporarily attached motor (we recall that all segments of the OmniTread will be powered from a single drive motor via the drive shaft spine). The current design of our prototype segment implements a simple clutch, which will allow one tread to jam without impeding the motion of the other treads in the robot.

6.3.2 OmniPede design

We made the changes to the OmniPede’s propulsion system that we proposed in last month’s report. That is, we (1) changed the ratio of the worm gears, (2) increased the footprint area of the feet, and (3) covered the bottom of the feet with high friction material. The effect of (1) is an almost two-fold increasing in the speed. However, there is also an almost two-fold increase in power consumption. The new feet with improvements (2) and (3) also work fine, but only on flat and hard floors. The OmniPede still fails to move properly on gravel and sand. The apparent reason is that the step height is too small. The step height, defined as the distance between the fully extended and fully retracted position of a foot, is ~14 mm. However, because of the geometry of walking with two feet per segment, the *effective* step height is only $14x(2)^{-½}$ = 10 mm. Yet, even the new feet with larger footprint areas sink into gravel or sand by more than 10 mm. As a result the feet push the gravel or sand while doing their return stroke (i.e., during the retracted phase). This undesirable effect substantially reduces the efficiency of motion.

6.3.3 Embedded control modules

We performed a detailed comparison of microcontrollers for the control of the pneumatic actuators and their position feedback for the joints of the Omnis. Our intention is to have a distributed control system, in which all actuators and feedback sensors belonging to one segment are controlled by a microcontroller on that segment. Microcontrollers are wired to a simple serial bus, called CAN bus, that runs the length of the Omni. Based on our comparison we selected the PIC18F458 from Microchip for our application. This microcontroller has four hardware-supported PWM channels, which is exactly what’s needed to operate the four pneumatic valves on each segment. Eight 10-bit A/D converter are available for joint-angle feedback and possibly pressure feedback. Furthermore the PIC18F458 controller has the fastest machine cycle, yet the costs of the development tools is the lower than that of its competitors.

Based on this selection of microcontroller we analyzed the spatial arrangements for the microcontroller itself and the associated electronic components. The required space is a parallelepiped of 1”x2.2”x2.2”. Since the segments of the OmniTread are essentially hollow, they provide sufficient space for the controller components. However, free space on the OmniPede segments is more limited. However, with some effort we believe that the electronic control components can be rearranged into a form factor of 1”x1.1”x4.4”, which can be fitted to the segments of the OmniPede.

6.3.4 Active pneumatic suspension vehicle

We finished assembling the pneumatic suspension vehicle test-bed. We also installed all pneumatic and electric connections. For the purpose of evaluation we mounted two different types of pneumatic valves. The two types differ from each other in their flow rates by a factor of 3. Next month we expect to conduct some dynamic tests to determine experimentally which type of valves produces more desirable results.

6.4 Infrastructure

6.4.1 Common software structure

For the past 4-5 years most of our mobile robot software development was done for the Pioneer AT as the target platform. More recently, however, we began work on several new projects, which use several different mobile platforms. These platforms are (1) the Pioneer AT, (2) our in-house-built Mars Rover clone Rocky-8B, (3) the Gorilla robot used exclusively for our URPR project, and (4) an ATRV platform stationed at CMU, for which we are developing code. We even expect to get a Segway personal transport into the lab, sometime next year.

With the increase in the number of different robotic platforms, as well as the number of programmers developing software for these platforms, we identified the need for tighter software standardization and organization. To meet these needs we spent significant efforts on analyzing our existing code, and on reorganizing this code in an architecture that is applicable to all our current mobile platforms. In practice, the relationship between modules and the method of communication between them will thus be the same for all our robots, while the properties and functionalities of the modules may differ depending on the specifics of each robot platform.

An ideal way for implementing this kind of modular approach and our new architecture is the so-called “Object Oriented” programming that is widely used in software development. We have now completed the design of our new unifying architecture and implemented most of its modules or “objects.” We have also tested and debugged most of the modules. Throughout this work we paid special attention to portability of the code across operating systems.

We also designed and implemented an Object Oriented utility for providing debugging facilities to all the programs currently being used. This utility generates debugging messages, displays warnings, and flags errors. We tested the utility and it appears to work well.

6.4.2 High-end Analog-to-digital Converter

We procured a high-end PCMCIA-based A/D converter card made by National Instruments. This powerful multifunction card, the DAQCard-AI-16XE-50, offers 16-bit resolution, 16 analog inputs, and a sampling time of 200 kS/s (kilo-samples per second). We installed drivers for the card under real-time Linux.

6.4.3 Control Code for the Gorilla

We implemented additional motion control functions for the Gorilla vehicle, as well as a joystick interface based on these functions. The set of completed motion control functions, most of which are accessible under the new joystick control, are:

1) Moving forward/backward.
2) Changing speed on the fly, in cruise control fashion.
3) Turning the steering wheels left/right.
4) Applying the brakes.
5) Calibrating the steering.
6) Emergency stopping.

6.5 Radiation Sensing and Imaging

The emphasis of this month has been the IEEE NSS/MIC Conference, help in Norfolk, VA. A fair amount of time has been spent on preparing the poster presentation, conference record and manuscript to be submitted to IEEE Transactions of Nuclear Science archival journal.

We have also been working on setting up the PSPMT-VAMCR system:

- A housing design has been finished and will be sent to Physics Workshop for manufacturing.

- The VA-MCR readout system was tested and the preliminary results seem promising. We are currently working on deriving the characteristics such as gain, noise level and threshold settings.

A new iteration of detector design/evaluation was outlined. It will be based on the use of the 5” PSPMT and scanning multiple hole collimator (or time modulated coded aperture).

☐ 7. MAJOR ACCOMPLISHMENTS:
Note MAJOR accomplishments during the reporting period; "None" is a valid, *occasional* entry.

We met with a representative of the University of Michigan Intellectual Properties Office to discuss the possibility of extending the soon-to-be-issued OmniPede patent to the OmniTread design. The Intellectual Properties Office has not yet made a final decision on how to proceed but is likely to submit an extension that will add the OmniTread design to the OmniPede patent.

MILESTONES. *Check that you have updated the status of your milestones.*

☐ 1. MILESTONE STATUS UPDATES:
Make sure you have provided a brief, one- or two-sentence comment on each active milestone and completion/new forecast date as appropriate.

pm/11-03-02