Anecdotes from Rover Field Operations

Vandi Verma
vandi@cs.cmu.edu
RIACS Summer Student Resarch Program, 2000
NASA Ames Research Center, CA

Abstract

This document is an attempt to investigate requirements and test scenarios for a rover fault detection, diagnosis and recovery system. It is based on the viewpoint that the problems that have been experienced in past field tests capture the characteristics of the fault detection and diagnosis problems that will be encountered in future missions. Moreover these are problems for which real data is available or may be easily acquired.

1 Introduction

A large number of current fault detection, diagnosis and recovery systems use simulated data. Typical experiments, even with real data, involve removing some sensor information that is easily available on most current rovers. These experiments demonstrate the utility of these methods at an academic level. Most often the experiment is crafted to demonstrate the specific strengths of the method under consideration. The danger with this approach is that, at times effort is spent in developing methods that solve problems that do not actually exist on real rovers while ignoring the ones that do. This document is an attempt to try and focus fault detection, diagnosis and recovery work on the problems that rover operators need solutions for. Each section is based on a particular field test.

2 Hyperion rover

Hyperion [1] (fig: 1) is a robot developed at Carnegie Mellon University. At the time of writing this document this rover was in deployment in the Arctic. Hyperion is designed to explore the terrain while being cognizant of the position of the sun, and the available and required energy levels for exploration. It must optimize a path through the terrain based on all these factors.

Hyperion's control system is designed for *sliding autonomy*. It can smoothly slide from direct teleoperation where a human operator tells it everything to do, through modes of control where the operator and robot share decision making, to full autonomy where Hyperion decides for itself how to perform a given mission, where to go and when. The health monitoring system is at times required to monitor parameters even under teleoperated control. This is because experience has shown that there is always a lot of information

[1] Based on the author's experience with developing the Health Monitoring system for Hyperion

Figure 1: The Hyperion Robot in the Arctic.

being relayed back to the operator and at times critical situations may go unnoticed.

What the Hyperion team wanted was health monitoring capability that would enable the robot to decide when it needed help. If it was unable to find its way, thought the mission was impossible, or detected strange behavior from its sensors, it was required to send a message to human operators about what had happened and and to stop and wait for instructions. When everything was okay it was required to let the operator know that it was ready to pick up and continue on its own.

On hyperion simple limit checking was used to determine abnormal situations. These can easily be detected by even the most simple health monitoring system. But they were found to be important in a field mission that required continuous operation over a long period of time. The variables that were required to be monitored were power level from voltage and current at solar array and battery, temperature (outside, E-box, at motors and battery), vehicle speed, roll, pitch, joint angles, status of each sensor as reported by the sensor, status of each software process as reported by the process.

On Hyperion there were GPS problems. The position estimation system used GPS information and dedreckoning information. When GPS information was accurate, it was used, else dedreckoning information was used. The problem was that at times the accuracy of GPS information would degrade, but this would go undetected. Switching to dedreckoning information at this stage resulted in a jump in position, which posed a problem for the navigation system. When accurate GPS information was obtained, again there was a jump. A simple limit switch based health monitoring system was insufficient for this. *Filtering and integrating information from multiple sensors is required to address subtle performance degradation.*

Another problem encountered by Hyperion was that it had a very large solar panel. In high winds the rover would tip over. The health monitoring system detected that the roll was over threshold, but there was no autonomous recovery action implemented on Hyperion and thus this was useless. A remote operator cannot do much with information that he cannot act on remotely. A number of problems that rover operators deal with all the time, fall into this category. *Unless it is possible to take preventative or corrective action in response to detecting a fault, detecting that fault is worthless.*

3 Dante II

Dante [2](fig: 2) was a terrestrial, semi-autonomous walker designed at Carnegie Mellon University. It was designed to climb treacherous and unstable terrain with an aggressive slope.

[2]Based on discussions with Dave Wettergreen

Dante II successfully climbed down the mouth of Mt. Spurr, a volcano in Alaska in the summer of 1994.

Figure 2: Dante II at Mt. Spur.

While ascending out of the crater, Dante II encountered step slope and cross-slope conditions. The robot was being teleoperated at that time. The operator had a number of parameters to track at that time and failed to notice that due to the steep slope only one of the rear legs of the robot was in contact with the ground. When the weight of the robot shifted as it was climbing out, that leg slipped and Dante II fell on it's side. it was unable to self-right from this position and had to be rescued by helicopter. This anecdote shows that even under operator control it is essential to detect fault conditions. Ideally fault monitoring would be active at all times. The operator would only turn off the monitoring if it interfered with recovery from unanticipated situations.

4 Nomad

In January 2000 the Nomad [3] (fig:3) robot explored the remote Antarctic region of Elephant Moraine in search of new meteorite samples. Nomad autonomously found and classified in situ five indigenous meteorites and dozens of terrestrial rocks.

Figure 3: Nomad in Antarctica.

Anecdotes from the mission are listed below:

- Nomad was equipped with a manipulator arm that placed a visible-to–near-infrared

[3]Based on discussion with Stewart Moorehead

reflection speedometer within a centimeter of a rock face. At one time, the limit switch that detected whether the arm was in position died. This could have been detected by monitoring the torque on the joints.

- When the robot was first deployed, the wheels were set in deployed configuration.One of the tires got stuck in the concrete. A large force was exerted on the rack and pinion gears of the steering mechanism. This caused a gear on one of the wheels to jump some teeth. The pose of the rover wheel from that point on was incorrect for the rest of the mission.

- The compass on Nomad would drift and cause heading problems.

- The motor mount on Nomad became loose on one of the wheels. This went undetected until the rattling sound caught the attention of the operators. There were a number of problems that the operators claim were detected by listening for them. *Maybe tracking nominal operating sound of a rover should be considered.*

- Nomad ran out of oil in the generator. There was a sensor to monitor the oil level in the generator, but in the Antarctic, it froze often. Temperature measurements could have been used to determine the confidence in the oil level measurement.

- Due to glare from the ice, stereo vision performance was poor.

- Nomad ws equipped with a high resolution camera mounted on a pan tilt unit. It enabled it to visually servo to a target and zoom into it to determine if it was worth deploying the spectrometer on the target. On start up the pan tilt unit went through an initialization routine. Often the wires attached to the camera would get stuck when it was initializing and it would not complete the process.

- There were a number of problems with acquiring all the color channels on the pan tilt.

- When the pan tilt unit was started after the rover had been standing all night, the lens would fog up.

5 Marsokhod

During the NASA Ames 1999 Marsokhod field test [4], a model-based technique called Livingstone was used for monitoring and diagnosis. The problems experienced with this technique are listed below:

  - *Keeping up with sensory data* The lesson here is that in the rover domain there is a large amount of data and any system that is tracking the behavior of the system will have to be very fast.
  - *Tracking multiple hypothesis* This is essential because of the uncertain actuators and noisy sensors.
  - *Inability to handle quantitative models* It was determined that qualitative models are not expressive enough for the rover domain.
  - *Inability to handle temporal models* There should be a way to specify temporal constraints.

[4]Based on discussion with Keith Golden and Hans Thomas

– *Inability to express non-deterministic conditional transitions*

– *Hard to create the required models* Ideally we want to be able to use a specification of the underlying physics directly. A model created specifically for diagnosis is cumbersome and defeats the purpose.

During the Marsokhod field test, [5] the two middle wheels of the rover would occasionally rise above the other four wheels. This configuration has lower traction and is therefore sub optimal.

In this configuration, each reasoning *locally* about an individual wheel is insufficient. It is necessary to reason *globally* about the relative position of the six wheels.

Figure 4: Marsokhod Rover in the Mojave desert.

6 K-9

K-9 [6] 5 is a rover being developed at NASA Ames to integrate and demonstrate new enabling robotic technologies for future Mars missions.

The K-9 rover at NASA Ames often stalls when the rover is backing up and hits an obstacle. This makes the vehicle yaw sharply in one direction. This state can be detected by monitoring the motor current, encoder values and rotational speed at each wheel locally. A stalled motor is a transient event.

The potentiometers on the steering actuators on K-9 develop a systematic error over time. They are calibrated for driving straight ahead, but if there is a systematic error the rover will erroneously move along an arc instead. Looking at the gyro data would determine that the rover is turning.

But, a similar diagnosis might result if the rover were driving on a side slope, [7] and it would be necessary to look at the inclinometer to resolve the two modes.

The K-9 team is interested in detecting situations, that faults in the traditional sense since they don't result from a component or software process having failed completely, but result in suboptimal operation. *Detecting non catastrophic faults that result in control changes that dramatically influence the cost of operation is far more important than detecting catastrophic faults from which no recovery is possible.*

[5] This problem also occurs on the rocker bogie platform or any other passive suspension system

[6] Based on discussion with Hans Thomas, Maria Bualat and Anne Wright

[7] K-9 is commanded through an open loop controller

The K9 team is interested in detecting faults such as wheel slippage, wheel sinkage, stuck wheel, bogie angle out of bounds, overexposed images, inaccurate heading estimate due to potentiometer drift, loss of gear tooth, inaccurate estimate from magnetic compass and constant offset in steering angle.

Figure 5: The K9 rover in the Mojave desert.

An issue that came up while attempting to set up an experiment to demonstrate fault detection on the K-9 rover was that *rover engineers do not want the robot to get damaged during testing.* They are therefore unwilling to put the robot in situations that test a fault detection system. It therefore seems that most fault detection and diagnosis work will have to be developed in simulation or with real rover data that has been modified to include faults. Testing will have to be done by putting the system on a real rover during a field test and letting it handle the real faults that arise during these missions. The problem with this approach is that this does not test the complete system.

7 Acknowledgments

These anecdotes were collected from a number of people that were part of the various field missions. Michael Wagner, Stewart Moorehead, Ben Shamah, David Wettergreen, Hans Thomas, Maria Bualat, Rich Washington, Anne Wright and Hamid Aljabri provided valuable information.