Fault Tolerant Action Selection for Planetary Rover Control

Terry Huntsberger

Intelligent Systems Laboratory, Department of Computer Science
University of South Carolina, Columbia, SC 29208
terry@cs.sc.edu

ABSTRACT

The ability of a small rover to operate semi-autonomously in a hazardous planetary environment was recently demonstrated by the Sojourner mission to Mars in July of 1997. Sojourner stayed within a 50 meter radius of the Pathfinder lander. Current NASA plans call for extended year-long, multikilometer treks for the 2003 and 2005 missions. A greater deal of rover autonomy is required for such missions.

We have recently developed a hybrid wavelet/neural network based system called BISMARC (**B**iologically **I**nspired **S**ystem for **M**ap-based **A**utonomous **R**over **C**ontrol), that is capable of such autonomy. Simulations reported at this meeting last year demonstrated that the system is capable of control for multiple rovers involved in a multiple cache recovery scenario. This robust behavior was obtained through the use of a free-flow hierarchy (FFH) as an action selection mechanism. This paper extends BISMARC to include fault tolerance in the sensing and mechanical rover subsystems. The results of simulation studies in a Mars environment are also reported.

Keywords: autonomous rovers, behavior-based control, fault tolerance, control hierarchies

1. INTRODUCTION

Autonomous operation of rovers in remote, harsh environments such as the surface of Mars obviates the need for some measure of fault tolerance in system components and control. Faults can occur based on environmental influences such as cold temperatures, as well as mechanical/electronic failures. Depending on rover mass/power constraints, component redundancy may be an option to address such cases.[6] In general, a control system that is able to adapt to component failures would offer some measure of robustness. Most hard-coded control structures including the FFH used in BISMARC don't have this flexibility.[10] One approach to this problem is to directly include uncertainty in the control mechanism.[11] Another approach is to include the notion of uncertainty in the sensor models.[1] Both of these approaches address the problem of imprecision in the sensors, but not component failure.

The failure modes that could potentially compromise a rover mission include and are not limited to:

- loss of one or both stereo cameras in front and back,
- loss of mobility in one or more wheel sub-assemblies,
- loss of power regeneration capabilities,
- loss of one or more wheel encoders,
- loss of one or more degrees of tilt sensing,
- loss of internal temperature sensing capabilities, and
- loss of CPU and/or memory.

Previous fault tolerant studies for robotics have mostly concentrated on joint failure in manipulators.[8] The proposed solution in such a case generally entails some sort of component redundancy. Mass, power and size limitations for planetary rovers constrain the level of redundancy that is possible.

We previously developed a control system for autonomous planetary rover control called BISMARC (**B**iologically **I**nspired **S**ystem for **M**ap-based **A**utonomous **R**over **C**ontrol) [3, 5], that uses a hybrid neural network[4]/free flow hierarchy[9] architecture for action selection. This system had some degree of built-in fault tolerance in the action selection mechanism, but it could not handle rover sub-component failures. Such failures would result in incorrect rover actions due to the fixed nature of the action selection hierarchy.

The next section describes the overall organization of the BISMARC architecture. This is followed by a discussion of the inclusion of fault detection/tolerance in the action generation system. Short term memory mechanisms are used for the fault detection process and sensory perception modification is used for fault tolerance. In some sense, the modification can be viewed as a type of learning behavior. The results of some failure mode scenarios in simulated planetary rover missions is described next, followed by a final summary section.

2. BISMARC ORGANIZATION

The three level BISMARC architecture (shown in Figure 1) uses a hybrid mix of neural networks and behavior-based approaches. The first level performs a wavelet transform on the rover's stereo image pair, the second level inputs these processed images into an action generation navigation network, and then to a third level action selection mechanism (ASM) network modeled after that of the Rosenblatt and Payton FFH.[9] This type of FFH was recently shown to be optimal within the multiple objective decision making (MODM) formalism, which produces an action through the maximization of a global objective function that includes all possible actions.[7] Some examples of the other external inputs would include internal temperature sensing, relative time of day, sun sensor positioning, and possible communications with other rovers.

BISMARC uses a FSOFM to learn *landmarks* (obstacles and goals). In the operational mode, this network generates membership values to the classes of visual input that the system has previously seen. When coupled with onboard rover components such as accelerometers and dead reckoning inputs, an egocentric *map* of the environment is built using the FSOFM response as an

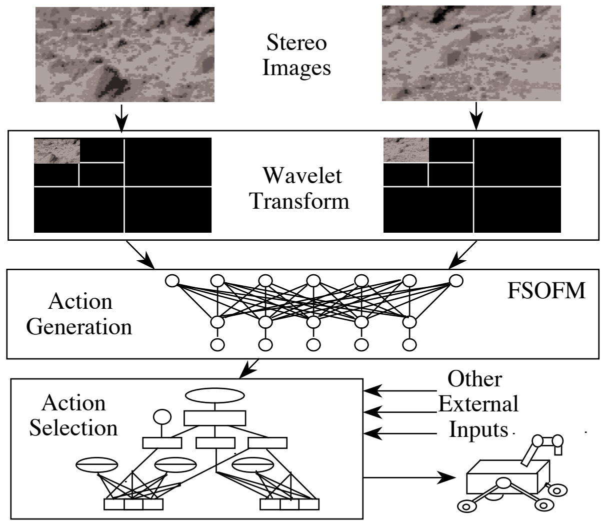

Figure 1: Multilevel system organization of BISMARC for autonomous rover control in planetary environments. Coefficients from the wavelet detail channels are used to generate actions with a FSOFM. An ASM then performs a combination operation on the possible actions for final navigation.

index. Since unsupervised operation or training a system in a planetary environment such as Mars would be costly and potentially dangerous to the rover, BISMARC offers a compromise solution.

There are a number of factors beyond the visual sensory input that influence navigation of the rover. These include the health of the rover (internal/external temperature, battery power levels, accelerometers/gyro), time of day, and homogeneity of the terrain. For example, if the rover is approaching a highly uneven portion of the terrain rather late in the day, the control decision may be made to halt and wait for the next morning in order to recharge the batteries and to have enough light for visual sensing. The FFH automatically handles action selection in the presence of such conflicting behaviors.

A FFH is a directed graph of action and stimulus nodes that are combined using predetermined rules. These rules may include addition, multiplication, or more complicated means of combination. The FFH system for BISMARC is shown in Figure 2. Action nodes are drawn as rectangles, stimulus nodes as ellipses, and those with multi-directional characteristics are indicated using 8 directional bins. The combination rules are additive for a small filled rectangle above the node, multiplicative for a small filled triangle, and a more sophisticated rule is used for plain rectangular nodes. [12] This more sophisticated combination rule was developed by Tyrrell to guarantee the proper transfer of goal and motivational behavior to lower levels of the FFH.

Tyrrell introduced the temporal penalty (T-circle in Figure 2) to control action that will take an inordinate amount of time to complete.[12] The temporal penalty is derived using the assigned value raised to the power of the elapsed time during the current action. Temporal penalty nodes increased the likelihood of satisfying the overall mission goal of maximizing the number of cache containers returned. In addition, the uncertainty penalty (U-circle in Figure 2) is used to control actions that are heavily dependent on external sensor inputs, which are usually noisy and imprecise.

The top level nodes in BISMARC generally relate to rover health (Avoid Dangerous Places, Sleep at Night, Warm Up, Cool Down, Get Power) or cache recovery/navigation (Scan for Cache, Get Cache, Keep Variance Low). These high level actions involve a complicated combination of

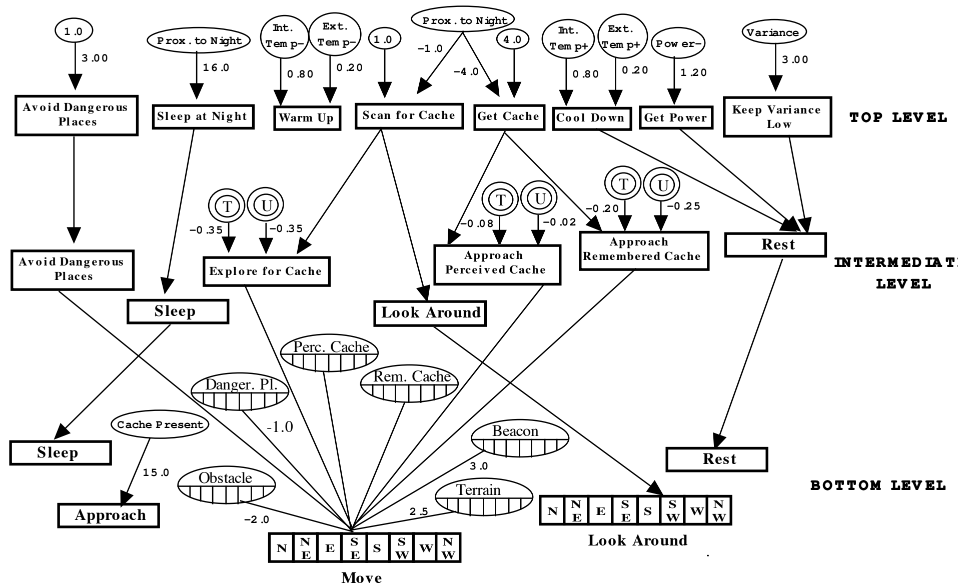

Figure 2: FFH system for BISMARC. All weights on the arcs are 1.0 unless otherwise indicated. Notation for symbols is that of Tyrrell.[12] See text for a detailed discussion.

internal control and assimilation of external sensor inputs. Since potentially conflicting behaviors can arise, the FFH offers a better approach to the low level control problem than direct elimination of lower nodes due to inhibition as found in purely reactive control.[2] There are only five bottom level nodes: Sleep, Approach, Move, Look Around, and Rest. The most sensitive bottom level node to faults is move, since its activation level depends primarily on the combination of sensor inputs with additional input from higher levels in the hierarchy.

The direct integration of the obstacle avoidance behavior into BISMARC's route planning strategy is a necessary component of the design. Since obstacles are used as *landmarks* for sensory/action map making, detailed information about size, relative height, etc. are important. The necessary clearance for a rover to navigate around an obstacle is built into the training sets, and looming is used to differentiate between small obstacles that can be driven over and the larger ones that require course changes. This subsystem would be the most heavily influenced by a fault in the stereo cameras, such as a loss of one camera.

3. FAULT DETECTION/TOLERANCE

Sensing in BISMARC is based on the concepts of memory and perception. All incoming sensor information is compared to a short term memory (STM) value for internal sensors such as temperature, and a map-based memory for external sensor input such as obstacles. These memory traces make fault detection fairly straightforward, since BISMARC uses a 10Hz sample cycle. We are explicitly making the assumption here that large changes in the sensor inputs at this sample rate are indicative of faults. The directional sensor perception activation is given by:

$$A_S = P_d \times (1.0 - dist) \times (1.0 - P_u),$$

where A_S is the activation level for sensor S, P_d is the normalized sensor input, *dist* is the normalized distance to the perceived objects, and P_u is the perception uncertainty. The perception uncertainty is given by:

$$P_u = ABS[P_d(t+1) - P_d(t)],$$

where $P_d()$ refers to the time separated normalized sensor samples. This expression for P_u experiences a maximum when the sensor input undergoes a full range swing. The perception uncertainty is used for fault detection (high values indicate a fault).

Fault tolerance on the other hand, is a more difficult problem. Due to mass and power constraints on the rovers, redundancy of system components is not necessarily an option. The FFH has a fixed set of weights and nodes that doesn't lend itself very well to adaptive reconfiguration in the presence of faults. Our solution to this problem is to adapt the sensor perception based on an uncertainty measure returned during the fault detection stage as shown in the expressions above. This gives us the best of both worlds: a stable control hierarchy coupled with adaptive sensor analysis. Sensors with a high uncertainty will have little effect on subsequent nodes. These sensors are flagged, and will be automatically allowed to come back on-line in the case of intermittent faults.

Since the FSOFM is providing obstacle avoidance information based on the input of the stereo cameras, any damage to them would compromise this input. In addition, these cameras are used for cache acquisition/approach and retrieval. Dangerous places are areas where relatively large gradients in the planetary surface are detected. Damage to any of the accelerometers or tilt sensors would make this information unreliable.

In the event that one or more of the wheels are damaged, the rover would be seriously hampered from any type of movement. We have assumed that any damaged wheel can be made to free roll, and as such, would not stop the rover from moving. The encoder inputs from these wheels however are considered faulty, since they are no longer under control.

4. EXPERIMENTAL STUDY

We ran 200 trials using a randomly generated heightfield. The area encompassed about 1KM by 1KM with a grid decomposition resolution of 5 cm. Each trial had different starting positions and the placement of a cache container was randomized within the area. It was assumed that each cache placement site was known to within a 200 meter radius using a beacon. The top speed on the rover was set at 30 cm/sec, which is consistent with SRR1. In order to simulate wheel slippage, we set a 10% loss of traction when climbing over a rock or traversing rocky terrain. The battery lifetime was set at one week on the rover with a timestep of 0.1 sec. The rover was forced to sleep during the night hours of the simulations, since there were no infrared sensors on the rover. Cache acquisition time using the 5 DOF manipulator on the rover prior to return to the lander site was assumed to be one hour.

In half of the simulations we randomly selected from the list of possible faults enumerated in the Introduction. In the other half, the faults were correlated with the environment (i.e. wheel failure in rocky terrain). We did not test the *loss of CPU/memory* fault since there is no way to run any control software without these components.

The success rate for cache retrieval was only 46%, with death of the rover most common in the loss of tilt sensors and/or stereo camera(s). Redundancy in these sensors might be worth the mass investment, since their loss totally compromised the mission. There are front and back stereo cameras on the current FIDO rover prototype at JPL. We currently don't use this option in BISMARC.

5. CONCLUSIONS

This paper extended the rover control system called BISMARC to include fault tolerance to some types of sensor and mechanical failures. The FFH used for action selection in BISMARC maintained its structure and weights, but the sensor perception models were modified to account for the faults. The results of 200 missions indicate that failure modes have varying impact on the rover. A ranking of failure modes from most to least destructive yields tilt sensor(s), stereo camera(s), battery power level indicator, and internal temperature sensor. Since the stereo and tilt sensors are combined independently, a better model might reduce the strong dependence on rover health from the failure of either subsystem. Preprocessing of the stereo cameras through the FSOFM would have to be modified by training the network with single camera inputs. We are currently investigating better sensor models, that will give a more sensitive heading activation than the eight-directional one currently used in BISMARC. These will be based on the sensors used on SRR1 and FIDO at JPL.

ACKNOWLEDGMENTS

This research was supported in part under ONR Grant N00014-97-1-1163 and ARO Grant DAAH04-96-10326. I would like to thank Paul Schenker for hosting me under the ASEE Visiting Faculty Program at the NASA JPL Planetary Robotics Laboratory during the summer of 1997 and for numerous visits since then. I would also like to thank Hrand Aghazarian of JPL for many helpful discussions about possible rover failure modes.

REFERENCES

1. E.T. Baumgartner and N. A. Klymyshyn. A sensitivity analysis for a remote vision-guided robot arm under imprecise supervisory control. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Distributed Robotic Agents*, pages 218–226, Boston, MA, November 1996.
2. R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.
3. T. L. Huntsberger. Autonomous multirover system for complex planetary retrieval operations. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, pages 221–227, Pittsburgh, PA, October 1997.
4. T. L. Huntsberger and P. Ajjimarangsee. Parallel self-organizing feature maps for unsupervised pattern recognition. *International Journal of General Systems*, 16:357–372, 1990. Also reprinted in *Fuzzy Models for Pattern Recognition*, (Eds. J. C. Bezdek, S. K. Pal), IEEE Press, Piscataway, NJ, 1992, pp. 483–495.

5. T.L. Huntsberger and J. Rose. BISMARC: A Biologically Inspired System for Map-based Autonmous Rover Control. *Neural Networks*, 1998. in press.

6. M.L. Leuschen, I. D. Walker, and J. R. Cavallaro. Robot reliability using fuzzy fault trees and Markov models. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Distributed Robotic Agents*, pages 73–91, Boston, MA, November 1996.

7. P. Pirjanian and H. I. Christensen. Behavior coordination using multiple-objective decision making. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, pages 78–89, Pittsburgh, PA, October 1997.

8. M. Rokonuzzaman and R.G. Gosine. Minimization of the effect of fault-clearance period in the fault-tolerant sensing of an intelligent system. In P. S. Schenker and G. T. McKee, editors, *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, pages 239–249, Pittsburgh, PA, October 1997.

9. J. K. Rosenblatt and D. W. Payton. A fine-grained alternative to the subsumption architecutre for mobile robot control. In *Proc. IEEE/INNS Joint Conf. on Neural Networks*, pages 317–324, Washington, DC, June 1989.

10. Y. Ting, S. Tosunoglu, and B. Fernandez. Control algorithms for fault-tolerant robots. In *IEEE Conf. on Robotics and Automation, Vol. 2*, pages 910–915, 1994.

11. E. Tunstel. Mobile robot autonomy via hierarchical fuzzy behavior control. In *Proc. 6th International Symposium on Robotics and Manufacturing*, Montpellier, FR, May 1996.

12. T. Tyrrell. The use of hierarchies for action selection. *Journal of Adaptive Behavior*, 1(4), 1993.