# A Simple Drive Load-Balancing Technique for Multi-wheeled Planetary Rovers

James C. Finnis and Mark Neal

Computer Science, Aberystwyth University, Penglais, Aberystwyth SY23 3DB
(jcf1@aber.ac.uk, mjn@aber.ac.uk)

**Abstract.** A simple method for balancing the motor driver load across a six-wheeled rover is presented. This method uses the concept of inflammation to model the load on each motor driver by its temperature, decreasing the load as the local temperature increases. The method is compared with both the base case, where all motors run at a fixed load; and a relatively unintelligent method involving shutting off all motors while the temperature is above a given level.We show that load balancing in this manner has the beneficial effect of avoiding overheating in individual motors, while neither overly decreasing the traversed distance nor increasing the energy used.

**Keywords:** Robot, rover, temperature, PID, load balancing, inflammation.

## 1 Introduction

Autonomous robots need to be able to function for extremely long periods of time without any operator intervention. This applies particularly to planetary rovers, which cannot be repaired or maintained after their initial deployment.

Faults can develop over time which can cause unforeseen variations across duplicate components. For example, the drive motors of multiwheeled rovers should ideally respond in an identical manner, but variations will occur — from minor changes in response to outright failure. This is compounded by the inevitable slight variations in manufacture between the components.

An example of such a failure occurred on the Mars Exploration Rover Spirit in 2004, when the right front drive motor began to draw approximately twice as much current as the other wheels due to a problem with the distribution of lubricant[1]. This problem persisted intermittently and slowly worsened, despite attempts to exacerbate the load by dragging the wheel, until it finally failed in March 2006[2].

This case demonstrates one key area in which variations can occur: motor load. A typical planetary rover has six wheels, each of which may develop faults which could cause the load at a given speed to change — lubrication problems, gear wear and so on. In addition, each wheel is positioned at different heights on a different patch of terrain with different properties. This can have a very large effect on the friction and slip experienced by each wheel, and therefore the load on that wheel's motor and its driver.

Such variations can cause one motor to become overloaded, which can eventually cause damage due to thermal and mechanical stress on the components. This can be exacerbated by the harsh environments in which planetary rovers must function: a lunar environment has no atmosphere to provide convective cooling, and no "cleaning events" in the form of fortuitous winds to clear away dust particles, which block radiative cooling. This led to the loss of the early Lunokhod-2 lunar rover[3]. The low-pressure Martian environment is little better for convective cooling.

The method described in this paper is designed to limit overloading individual motor drivers by using an analogue to a biological response to damage: inflammation. As part of the inflammatory response, substances called "cytokines" are released by damaged cells. These are small, short-lived proteins released by cells as signalling agents which are used for many purposes throughout the body, typically acting on nearby cells (i.e. in a *paracrine* manner). They are complex to classify and study: one task may be performed by several cytokines (redundancy), and one cytokine can have many unrelated functions (pleiotropy).

One fairly well-established effect, however, is the role of certain cytokines in the body's response to damage[5]. Cytokines are released by damaged cells, causing a complex series of changes leading to inflammation — the most important, perhaps, being the migration of white blood cells out of the bloodstream and into the injured area, where they can deal with potential infection and remove the damaged cells.

Another cytokinetic inflammatory effect directly changes the behaviour of the animal. Cytokines bind to pain receptors, sensitizing them so that normally painful stimuli become more painful (hyperalgesia) and those not normally experienced as painful evoke pain (allodynia.) This offers the organism a clear advantage: it will avoid exercising the damaged part of the body[7].

Although cytokine receptors such as the primary thermal nociceptor TRPV1 have sigmoid activation functions[4, 11, 9], we are currently using a ramp activation function. This is justified because we are simply using the temperature as a convenient analogue to a cytokine-like quantity, not as temperature *per se*. If we did not have temperature sensors, we could instead use the current to increment an exponentially decaying value in software, to the same effect. Thus we do not use temperature as the input to the receptor, but use it to change the threshold of the receptor so that more non-noxious stimuli become painful.

Our rover has low-resolution temperature sensors attached to the motor driver chips, which respond in an analogous way to cytokine release in the inflammatory response: as the motor is overloaded and "damage" increases, the temperature slowly increases, in the same way that cytokine concentrations would increase in a biological organism. Similarly, if the load is removed or decreased, the temperature falls; just as in biology the cytokine concentrations fall as the damage is repaired.

This similarity occurs because both processes can be modelled by a recurrence relation of the form

$$x_{n+1} = ax_n + b \tag{1}$$

In both cases, a quantity is being added to a value which is subject to an exponential decay[10]. We can therefore say that we are using the temperature of the driver as a physical analogue of the concept of a cytokine-like quantity: an exponentially decaying value which increases with load.

It might be thought more useful and direct to use the temperature directly, perhaps passing it through a sigmoid activation function such as would exist in a biological nociceptor (neuron sensitive to painful stimuli)[4]. However, the temperatures we are monitoring are non-noxious, well below any putative "pain threshold" for the rover. Again, we are simply using the temperature as a convenient analogue of a cytokine.

In our system, this temperature is used to sensitize a notional receptor, to the extent that allodynic pain might occur. If this sensitization is linear, we can argue that the *allodynic* pain response of nociceptors to the temperature is also linear, because the proportion of normally non-noxious stimuli which now cause pain will rise linearly. To model this, the temperature of each driver is passed through a simple linear activation function to produce the amount of allodynic pain.

With the pain response increased at the driver, the system will attempt to decrease the pain by decreasing the load. In our system, a short series of experiments showed that simply decreasing the required speed of a given motor slightly relative to others reduced the load on the driver enough to allow the temperature to start to fall.

## 2 Hardware

The rover used in the experiments is shown in Fig. 1, positioned on the simulated Martian regolith (see Sect. 3).

A half-sized ExoMars Concept-E chassis forms the basis of the rover. Each of the six wheels has three degrees of freedom: drive, steer and lift. The steer and lift motors are not used in these experiments and are commanded to the centre position. The numbering of the wheels and the organisation of the control system is shown in Fig. 2. The Concept-E chassis uses a "three-module concept," in which there are three independent suspension modules with two wheels each, each module having a central freely rotating pivot point [8] as shown in Fig. 3.

Our control system consists of nine off-the-shelf motor controller units based around an ATMega328p microcontroller and an L298 dual H-bridge driver. Each controller drives two motors. These controllers are themselves controlled by another ATMega328p, in the form of the very popular Arduino Uno board, via an $I^2C$ bus. This "master controller" receives commands from a small on-board Linux PC (a commodity netbook) via USB.

The motor control PID loops run on the individual motor controller units. Commands from the PC, sent via the master controller, change parameters,

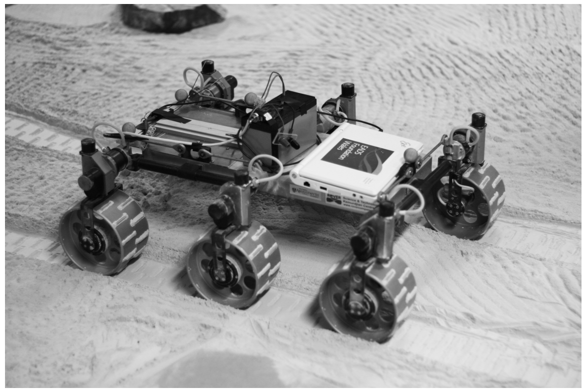


**Fig. 1.** The rover used in the experiments

required speed, etc. and can also read registers holding actual speed, odometry and similar data. Three of the motor controllers also read chassis orientation potentiometers.

Temperature monitoring is done via a network of DS-1820 1-Wire® sensors, all read by the master controller. Each sensor has an accuracy of ±0.5°C, and is read every 10 seconds (to allow for parasitic power usage). One sensor was placed against each of the L298 driver chips, while a tenth sensor was mounted outside the enclosure to monitor ambient temperature. Since each controller is responsible either for one drive and one steer motor, or two lift motors; and the lift and steer motors are not loaded, any temperature increase must be from the drive loads.

### 2.1 Control

The drive motors are controlled using proportional control only — tuning a complex system of six wheels on a highly variable terrain proved intractable in the time available, particularly considering the complex interactions between the different wheels through the body of the rover. Using standard techniques such as Ziegler-Nichols did little to reduce the oscillations while maintaining a good response time to setpoint changes.

## 3 Environment and Experimental Setup

All experiments took place in Aberystwyth's Planetary Analogue Terrain laboratory, on a surface of Mars Soil Simulant-D from DLR Germany, which is geophysically analogous to Martian regolith. This surface is similar to talcum powder in consistency, and has a particular tendency to cause wheel slip.

A track was established on the simulant terrain, consisting of a short (200 distance units, $200 \times 256$ odometer ticks) — approximately 4 metre) run, with

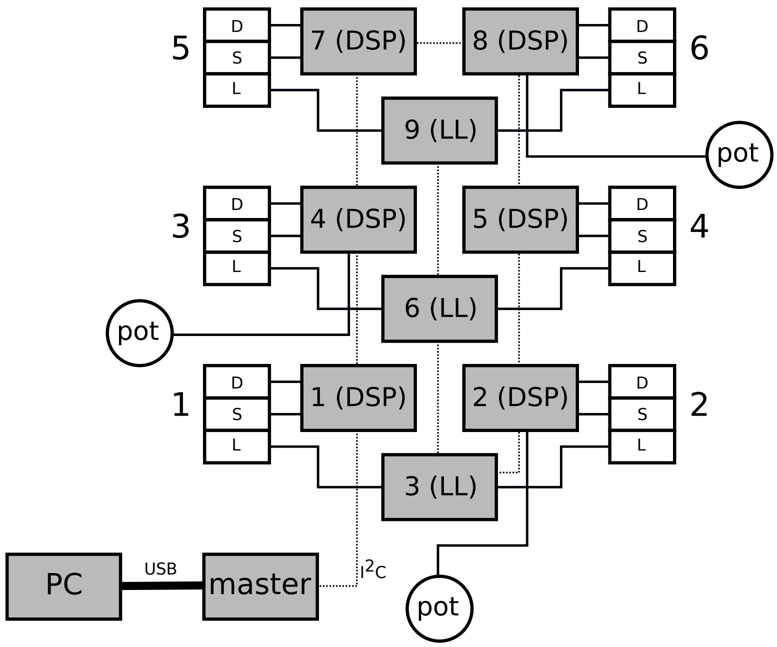



**Fig. 2.** Rover block diagram. Each numbered unit is a wheel, with 5 and 6 at the front. Each wheel has three motors for three degrees of freedom: Drive, Steer and Lift. Each pair of wheels has three controllers: one for each wheel's drive and steer motors (with optional bogey angle potentiometer), and one for both wheels' lift motors.

a slight incline of about 10° for just over half its length. In all experiments, the rover was programmed to change direction every 200 units by negating the desired speed at the end of each length, with the speed being set according to the algorithm being run. Each experiment ran for one hour, and was repeated five times.

Three different control methods were compared:

- a baseline, where the motor speed was set to 700 ticks per second with no variation;
- a so-called "dumb" method, where all the rover motors were shut down for a minimum of 5s if any motor driver temperature rose above a threshold level of 16°C above ambient (as measured by the external sensor);
- the method which is the subject of this paper, which was parameterised to keep all driver temperatures below 17°C above ambient.

## 4 Establishing a Baseline

Our first task was to establish a baseline heating curve. As stated above, this was done by simply driving the rover at the arbitrary maximum of 700 ticks per second. The results of all five runs are plotted in Fig. 5.

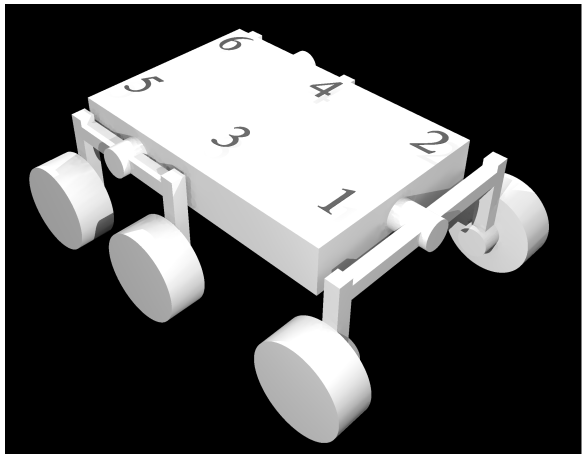



**Fig. 3.** Configuration of the suspension modules and wheels in the ExoMars Concept-E chassis, with wheel numbering.

**Summarizing the Data** Processing the large amounts of data produced by several runs of a single technique was a complex task given that we were interested in the maximum driver temperature each technique achieved, and that each run produced its results at different timepoints, due to the variability of the loop length in the the top-level PC control software.

To deal with this, we first took each run and found the maximum temperature across the drivers at each time point; and then we took the maxima for each run, extracted all the time points, and for each time point interpolated the values of all the runs' maxima. We then calculated the mean and standard deviation of the maxima thus found. This procedure is shown diagrammatically in Fig. 4.

Therefore, Fig. 5 shows the mean of the maximum temperature at each time for each run (since it is the maximum temperature we are trying to limit), with error bars showing 1 standard deviation. A curve of the form

$$y = \frac{a^x((a-1)c+b)-b}{a-1} \tag{2}$$

(the solution to the recurrence given in (1)) was fitted to all the temperature maxima, not just the mean, using Levenberg-Marquardt non-linear least squares and plotted using matplotlib[6].

## 5 Testing the load reduction method

Once this was done, tests were made to ensure that the cooling idea would work — that running a motor at a slightly lower speed relative to the others would reduce the load. The effects we see require that the terrain should have a high degree of slip (as the simulant terrain does).

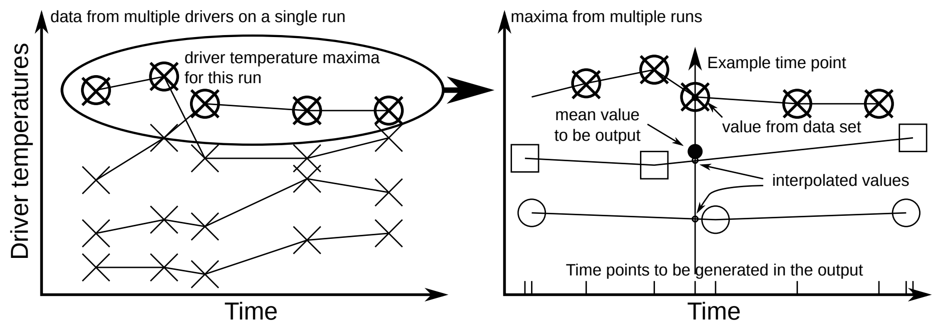



**Fig. 4.** Summarizing the data. In each run, the maxima of all driver temperatures is found at every given timepoint. Then, a set of all timepoints in all runs is found, and the mean and standard deviation of the maxima for all runs at all these timepoints is found, using linear interpolation where necessary since the intervals between the timepoints vary across the runs.

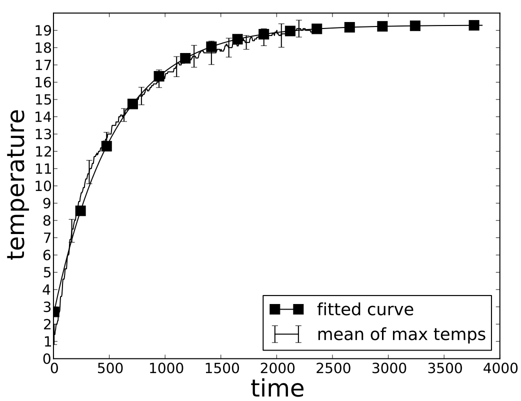
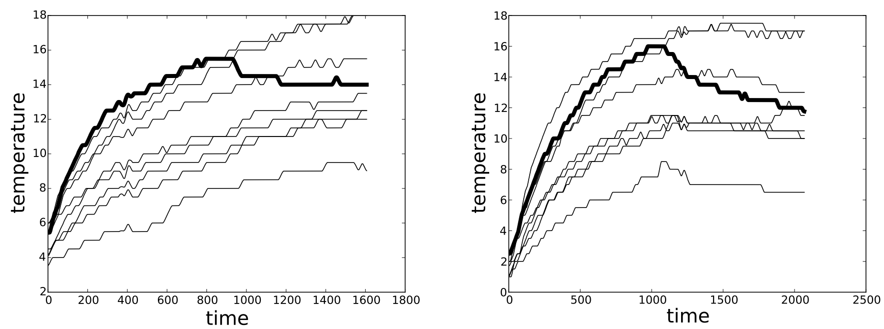



**Fig. 5.** An "baseline" run with no cooling algorithm — the motors always run at the set speed. The data is processed according to the data processing technique described in Section 4.

The test itself was straightforward: the rover was run as in the baseline test, and after a reasonably high temperature is reached, the set point of a hot motor was set to 0.7 of the set point of the other motors ($700 \times 0.7 = 490.$) Two typical runs on different motors are shown below in Fig. 6, indicating a prompt reduction in temperature when a motor is commanded to run at a lower required speed. Motor 3 is the centre left motor, while motor 2 is the back right motor. There was no observed deviation in heading when any motor was set to a lower speed, over a distance of approximately 4m repeated $> 50$ times, as indicated by the wheel tracks in the soil simulant.




(a) (b)

**Fig. 6.** Two runs testing the load reduction method. The temperature of the test motor is shown in bold. This motor is commanded from 700 to 490 after (a) $\sim$ 850s and (b)$\sim$ 1000s, leading to fairly prompt reductions in temperature through load reduction. In (a) it is motor 3, in (b) it is motor 2.

This is the effect we would like to see reproduced in the case of a motor with a persistently high load, perhaps due to a lubrication failure or terrain-related problem.

## 6 The "Dumb" Method

The most direct method for maintaining a low maximum temperature is simply to stop moving the robot when the temperature reaches a certain threshold level, only allowing it to continue once the temperature has gone below that level. Our experiment constantly monitored the temperature, stopping the rover if the temperature of any motor driver exceeded 16°C, given that we arbitrarily set 16.5°C as our maximum permitted temperature and our temperature sensors had a long latency (up to 10s). The rover was only permitted to continue once no driver had been above 16°C for at least 5s. The purpose of this method was to ensure that the "obvious" solution to the problem of overheating was no better that the final method.

## 7 The "Inflammatory Model" Method

**The Method** The desired outcome is a maximum temperature not exceeding about 16.5°C. To obtain this, we used a linear ramp from 12 to 18°C and a maximum reduction by a factor of 0.4 in the desired speed of each motor:

$$r(t,a,b) = \begin{cases} t \leq a & 0 \\ t \geq b & 1 \\ \text{otherwise} & \frac{t-a}{b-a} \end{cases} \tag{3}$$

$$f(t) = 1 - 0.4r(t, 12, 18) \quad (4)$$

where $r(t)$ is the ramp function between edge values $a$ and $b$, giving the pain response to temperature $t$; and $f$ is the multiple of the required speed to be fed to motor controller $i$ for a given temperature. Given that the experiment has a constant required speed of 700, we determine the motor speed with

$$s(t) = 700f(t) \quad (5)$$

This function is shown in Fig. 7.

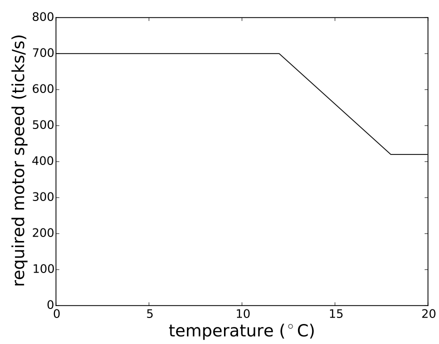


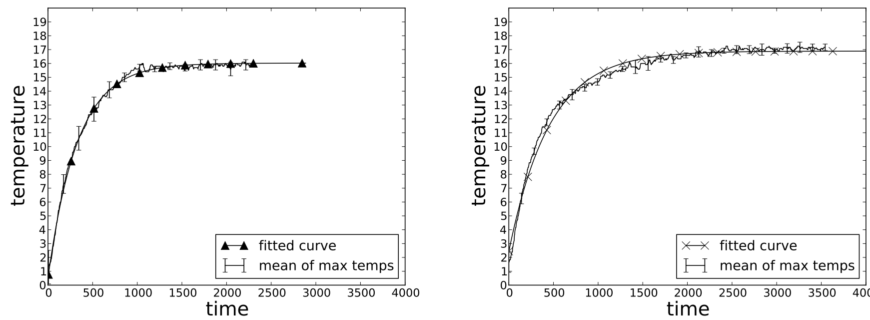

**Fig. 7.** The linear function mapping temperature to required motor speed, given a required vehicle speed of 700 ticks/s.

## 8 Results

The results of the two methods — the "dumb" method and the inflammatory method — were summarized in a similar manner to how the baseline was processed in Section 4, and exponential curves were fitted to the temperature results for all three experiments. The results are shown in Figures 8 and 9.

The "dumb" method — stopping the rover whenever the temperature becomes too high — keeps the temperature lowest, as one would expect. However, the load balancing method which is the subject of this paper (labelled here as "simple") does succeed in maintaining a fairly low temperature. While it does not quite remain below 16.5°C, it is certainly lower than the baseline. Better tuning of the parameters would give a better result.

These results should be viewed in conjunction with Figure 10, which shows graphs of distance and energy against time for the three experiments. Figure 10a shows that the strategy of just stopping for a brief time, while it does keep the temperature down, results in an unacceptable drop in distance traversed;




(a) "Dumb" (stopping) method (b) "Simple" (inflammatory) method

**Fig. 8.** The results of the two experiments, processed according to the method presented in Section 4. The mean maxima of the driver temperatures are shown with error bars, and an exponential curve following the recurrence relation in Equation 1 has been fitted. The graphs show that both methods effectively keep the maximum temperature down, well below the levels seen in Figure 5.

whereas the biologically inspired technique travels almost as far as the baseline in the same time. Figure 10b shows the mean energy usage of each method over time, showing that the energy usage of our method is nearly identical to baseline, although it is considerably higher than the "dumb" method. This is to be expected, since it is merely a method for load balancing, not load reduction.

It's worth noting that in the actual experiments, motor 3 was always the first motor to show a rise in temperature and was therefore the first to be set to a lower speed. This likely to be because motor 3 had recently had some maintenance which required disassembly and reassembly. It should also be noted that after this initial phase, other motors begin to develop a high temperature: after about 2500s, typically three motors are running hot and are slowed by the algorithm. Only motors 5 and 6 did not run hot consistently.

## 9 Conclusions

By considering the biological phenomenon of inflammation, we have developed a very simple method of load-balancing for multiwheeled robots on high-slip surface. Although the method is simple, it maintains a low temperature while not sacrificing total distance travelled, and expending no more energy. This shows that biologically inspired techniques need not be complex or dependent upon emergent behaviour to provide real benefits. Further work will:

1. develop a more accurate model of the sensitisation phenomenon rather than the current ramp function, to see if this improves the behaviour;

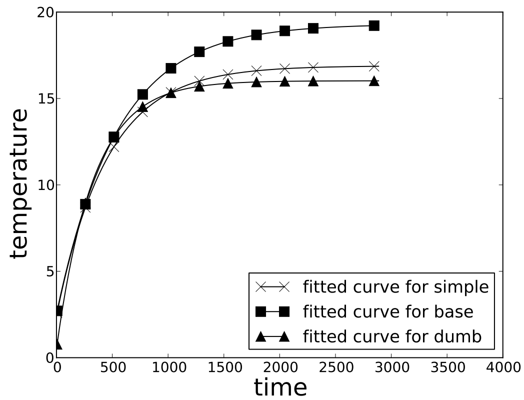



**Fig. 9.** The results of the baseline and the two methods, fitted to exponential curves of the form given in Equation 2. These show that the inflammatory "simple" technique keeps temperature down almost as well as the "dumb" technique of stopping when a motor gets hot.

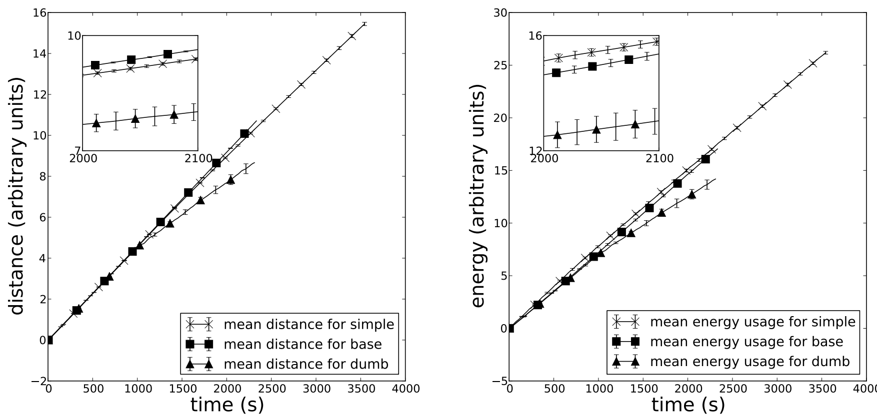
2. look at the paracrine behaviour of cytokines, to see if a high cytokine concentration at one wheel can be diffused to another wheel to produce an effect there even if that wheel does not yet have a problem;
3. look at the endocrine behaviour of cytokines, to see if a slow buildup of cytokine concentration at a central location can produce useful behaviour;
4. develop a mechanical model of the rover and its interactions with the surface so that multiple simulated runs can be made simultaneously, speeding up the development of new techniques.

## 10 Acknowledgements


The authors would like to thank Dr. Hannah Dee for her valuable comments and suggestions on an earlier version of this paper, and Professor Dave Barnes for his valuable support and for the use of the facilities of the Planetary Analogue Terrain Laboratory at Aberystwyth University. This work was partly supported by the Technology Strategy Board's ENDOVER project.


## References


1. Spirit Mission Updates, July 2004. `http://www.jpl.nasa.gov/missions/mer/daily.cfm?date=7&year=2004`, accessed: 2013-4-30
2. Spirit Mission Updates, March 2006. `http://www.jpl.nasa.gov/missions/mer/daily.cfm?date=3&year=2006`, accessed: 2013-4-30
3. Blair, S.: Rovers return. Engineering & Technology 6(3), 48–50 (2011)




(a) Distance (b) Energy

**Fig. 10.** Distance and energy against time for the baseline and both load reduction methods. These show that the distance traversed by the inflammatory technique ("simple") is much better than using the stopping ("dumb") technique and only slightly worse than the baseline (no temperature mitigation at all); while the energy usage for the inflammatory technique is only slightly higher than the "dumb" stopping technique.


4. Caterina, M.J., Schumacher, M.A., Tominaga, M., Rosen, T.A., Levine, J.D., Julius, D.: The capsaicin receptor: a heat-activated ion channel in the pain pathway. Nature 389(6653), 816–824 (1997)
5. Dinarello, C.A.: Proinflammatory cytokines. CHEST Journal 118(2), 503–508 (2000), +http://dx.doi.org/10.1378/chest.118.2.503
6. Hunter, J.D.: Matplotlib: A 2D graphics environment. Computing In Science & Engineering 9(3), 90–95 (2007)
7. Kress, M.: Nociceptor sensitization by pro-inflammatory cytokines and chemokines. Open Pain J 3, 97–106 (2010)
8. Kucherenko, V., Bogatchev, A., Van Winnendael, M.: Chassis concepts for the exomars rover. In: Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation, Noordwijk, The Netherlands (2004)
9. Macpherson, L.J., Geierstanger, B.H., Viswanath, V., Bandell, M., Eid, S.R., Hwang, S., Patapoutian, A.: The pungency of garlic: activation of TRPA1 and TRPV1 in response to allicin. Current Biology 15(10), 929–934 (2005)
10. Medeiros, I.M., Castelo, A., Salomao, R.: Presence of Circulating Levels of Interferon-g, Interleukin-10 and Tumor Necrosis Factor-a in Patients with Visceral Leishmaniasis. Revista do Instituto de Medicina Tropical de São Paulo 40(1), 31–34 (1998)
11. Montell, C., Birnbaumer, L., Flockerzi, V., Bindels, R.J., Bruford, E.A., Caterina, M.J., Clapham, D.E., Harteneck, C., Heller, S., Julius, D., Others: A unified nomenclature for the superfamily of TRP cation channels. Molecular cell 9(2), 229 (2002)