A Hexapodal Robot for Maintenance Operations at a Future Mars Base

Graham A. Mann[1] and Akapravo Bhaumik[2]

[1] Applied Artificial Intelligence Laboratory
School of Information Technology, Murdoch University,
South St. Murdoch, Western Australia.
g.mann@murdoch.edu.au

[2] Birla Institute of Technology
Ranchi, Jharkand, India.
arkapravobhaumik@ gmail.com

Abstract. The development of a teleoperated, hexapodal robot suitable for maintenance work at a future Mars base is described. In order to reduce the workload on human engineers at the base, it is desirable that the machine quickly acquire the ability to carry out routine tasks without direct manual control. To function most usefully in a hostile environment, it must do this on the basis of very few training trials and in the presence of considerable sensor and motion noise. A mapless navigation task in which the robot periodically visits a circuit of waypoints around the base is selected from among several potential jobs for the maintenance robot. Simulations are used to show that a simple playback of operator commands corrected using a combination of logged non-magnetic compass and bearing to optical landmarks produces good results, and suggests that the method may be generalisable to other maintenance tasks.

Keywords: hexapodal robot, adjustable autonomy, mapless navigation.

1 Introduction

The last decade has seen considerable development of mechanical legs as possible alternatives to wheels or tracks for mobile field robots. Rocker-bogey configurations with six powered wheels have been successfully used on a series of NASA's Mars rovers since the Sojourner in 1997. They have proven to provide a stable, reliable and shock-absorbing ride for the instrument packages those robots carried, but they do have limitations. For example, the MER rovers are costly, only mechanically stable to 45° off the horizontal, and have a top speed of only 50mm/sec [1]. Much higher speeds over very difficult terrain has been demonstrated to be possible with relatively inexpensive, mechanically simple (no jointed leg) hexapods while also affording excellent traction and stability in uneven or broken terrain. One such promising line of development, which has spawned a family of hexapodal machines using a common means of legged propulsion, began with the Rhex prototype [2,3]. Each leg of this system is a curved spring mounted on a single axis. The great advantage of this approach is its simplicity: the rotary motion is relatively easy to control, there are no knee joints and good results can be obtained without complex foot placement models. The main difficulties encountered are power consumption (endurance) and precise motion control – in particular, toe-stubbing and relatively poor steering.

Most practical field robots are still remotely controlled by a human operator at present. Yet if such a robot is to achieve its goal of reducing the workload on astronauts, it must be at least partly automated. To be more specific, adjustable autonomy [4] which provides for multiple control modes manual control, high-level commanding and full automation, should be the ideal. In the Mascot project, the goal is to provide a learning system through which control of a manually-guided task can be naturally and easily be transferred to automatic control.

A maintenance patrol task is chosen as the focus of these experiments. In this task, a robot at a Mars base would periodically visit a number of key waypoints at which time-and-location stamped sensor readings and high-resolution photographs would be taken. These would be copied to a human engineer's station at the conclusion of each patrol. In Section 2, the prototype Mascot robot will be described. In Section 3, an experiment in which simulated one-trial learning process is applied to this maintenance patrol task is described. Section 4 shows the results and discusses the prospects for implementation on a maintenance robot.

2 Mascot Robot

The Mascot field robot [5] (Figure 2), currently being developed by the authors, is designed to demonstrate that reliable, steerable six-legged locomotion that cannot be matched by wheeled machines. In order to avoid the well-known problem of unreliability in complex, jointed leg systems, the mechanism has been greatly simplified. Like its inspiration Rhex, the Mascot has six, simple passive spring legs, each mounted on an independent revolute axis (6 DoF in total) and driven by an 18V Metabo 100W DC motor fitted with a 150:1 planetary gearbox. The six motors are driven by Jeffrey Kerr LLC PIC-SERVO control boards connected to a 32-bit RS485 multidrop network controlled by an onboard Dell Inspiron compact laptop running Windows XP.

This configuration is simple, reliable and robust, yet provides adequate control. The machine is 590mm long, 570mm wide at the middle legs and 700mm from the ground to the top of the current camera mast. It weighs approximately 12kg. As with insects, the machine moves by "tripod walking": at any instant three legs are on the ground, and these alternate between the sides of the robot. The machine is steered by altering the phase relationship between the tripods on either side. The machine is currently able to achieve a speed of about 0.25 m/sec. on uneven ground, and is likely capable of much higher speeds. The main power supply for the motors consists of two 18.5 volt, 4.5Ah lithium-polymer battery packs. A separate 11 volt 1Ah lithium polymer battery supplies logic power. The camera platform is designed around a pair of EO5-380 CCD cameras mounted on a tilt-pan head. Each camera is capable of transmitting 380-line PAL colour video over a 2.4GHz wireless link. At this stage the cameras are not used by the robot as a vision system, but only as part of a low-cost teleoperation control system. This also depends on a commercial 6-channel 36MHz FM wireless remote control system, designed for model aircraft. Two channels of this control the effective left and right steering, and two channels control the tilt and pan motors of the camera head. When completed, the system will enable a remote operator to control high-speed motion of the robot while viewing real-time video from the cameras on a small LCD monitor. Depending on the performance of the machine, the project may progress to a high-level commanding mode or even to full automation.

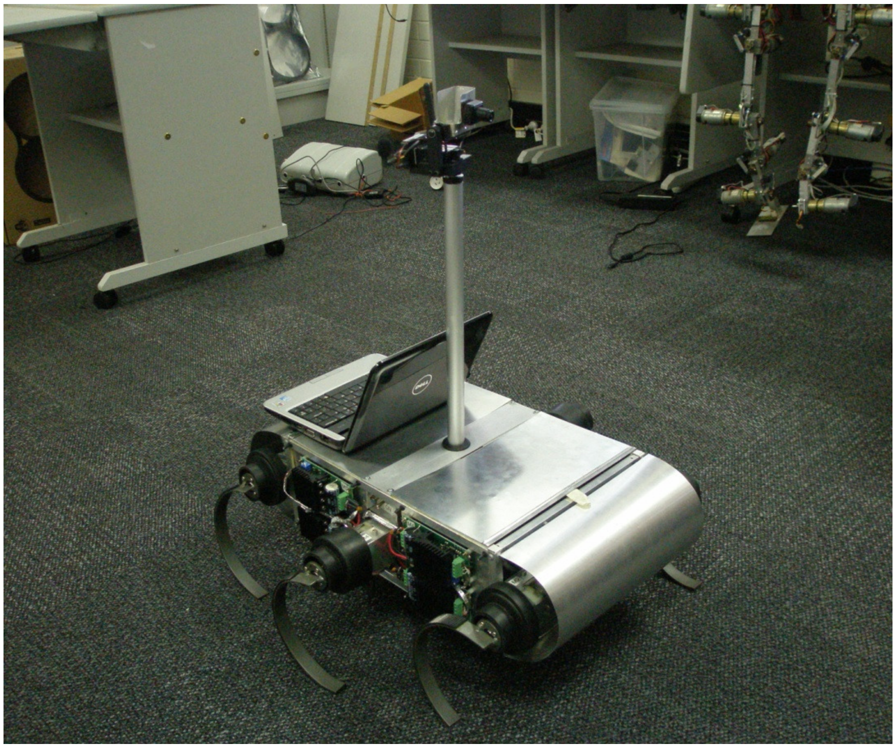

Figure. 1. Prototype Mascot field robot. Video of the machine in a test run is available at http://www.youtube.com/watch?v=OskuPsk9DRI

What tasks could a robot like the Mascot take on while setting up and operating a surface base, and in exploratory work? The answer depends on the task, the mode of control required (teleoperation, high-level commanding or full automation) and the provision of hardware and software for the total robot system (Table 1). We focus on maintenance tasks because of the importance of these for a future Mars station (for discussion, see [5]), as well as for remote stations on Earth. This table should be interpreted as showing increasing, cumulative demands on the equipment and behavioural competence of the robot as we move from the lowest demands in the top left of the table to the greatest demands in the bottom right. Thus simplest operational form of Mascot would be capable of Level 1 tasks if teleoperated, because those tasks only need accurate navigation and photography. At the other extreme, a long, well-funded research effort would be required to provide the all requirements specified in the table, including real-time planning and error-recovery software in order to automatically cope with unplanned, unstructured repair jobs.

Level 1. Location-based non-manipulation tasks (e.g. still and video imaging; transport of tools and consumables; instrument positioning) It is only necessary for the robot to navigate accurately to a location such as a possible trouble spot and take high-resolution photographs of the equipment concerned for transfer to an engineer's station. In teleoperation, the machine is guided by the human operator; in high-level commanding and full automation, the robot must plan a path between waypoints that avoids obstacles. Such a robot also could fetch and carry tools, equipment and samples on command.

Table 1. Cumulative requirements of the Mascot field robot relative to developed control mode and level of task.

	Teleoperation	***High-level Commanding***	***Full Automation***
Level 1 Tasks	functioning basic system allows inspection, photography, fetch and carry	add compass, GPS and obstacle- detecting sensors; add software planning layer on behaviour based reactive layer	connect cameras to vision system; add vision software to frame photographs, recognise humans
Level 2 Tasks	add at least one 6 DoF manipulator arm; routine maintenance manuals for operator	add specialised, detachable tool ends; vision software and touch sensors to guide tool use algorithms	add human-interruptible maintenance scheduling software off-board); algorithms for selection of tools
Level 3 Tasks	Provide more force at manipulator, tool end; more manipulators; detailed troubleshooting manuals/software for operator	add more and better sensors depending on task; voice command software	add best available real-time planning software (off-board); error recovery software

Level 2. Use of manipulators for planned, structured tasks (e.g. repair-by-replacement; spraying of paint, lubricant or sealant; changeout of a dust filter or replacement of a gas cylinder; staking or pegging structures such a solar panels or antennae; connecting and tightening electrical or stay cables; loosening or tightening bolts and nuts; sweeping or blowing dust off solar panels, instruments or cameras). This task requires, in addition to accurate navigation, the provision of one or more manipulators and/or specialised tools and the skill to bring them to bear on a particular work item. In teleoperation, the skill is that of the remote human operator; in high-level commanding it requires sophisticated sensors and intelligent control software. Scheduling would come from human-supervised, overrideable, automated scheduling software working to a routine maintenance schedule (both off-board the robot).

Level 3. Use of movement and manipulators for unplanned, unstructured tasks (e.g. repair on demand, given a diagnosis; disassembly and assembly of machines according to manufacturer's procedure; repositioning of fallen or displaced equipment; opening or closing stuck valves, doors and panels; unfreezing pipes; simple testing of electronic and mechanical components). These tasks require everything required in Level 2, but also presuppose a certain degree of problem-solving, and error recovery. This would come from human intervention, planning and reasoning overriding automated routine maintenance schedules. A larger selection of tools, probably more sophisticated sensors and probably a greater amount of applied force from the manipulators would be required. Such skill is difficult, but not impossible, to demonstrate [eg. 6]

3. Reproducing an Operator-Commanded Path

The simplest way of automating the task of moving around a tour of fixed waypoints would undoubtedly be to equip the robot with a GPS receiver and a compass. The operator would guide the robot to the each waypoint on a map of the base, and signal the robot to record its coordinates and orientation. Later, the robot would duplicate the tour by ordinary coordinate navigation, abetted by simple sensor-based obstacle avoidance. This method has been successfully demonstrated in many field robots [eg. 7, 8]. The principle drawbacks of such an approach are that the positional resolution of low-cost civilian GPS receivers is insufficient for exact placement (typically ~2.5m in devices suitable for mobile robots, although resolutions of ~10cm are theoretically possible) so that accurate placement of the robot may be difficult or expensive; that GPS systems will not work inside buildings, underground or near strong electromagnetic fields; and that (magnetic) compasses can be confounded by ferrous metal structures or electrical equipment likely to be found at industrial sites.

Although these are important considerations for practical Earthbound machines, we may confidently expect a Mars base to be equipped with an accurate GPS-like system for other reasons, which the robot could employ. More seriously, however, a robot that only acquires waypoints during manual control is not fully exploiting the intelligence of the human operator, as may be seen in the case where an apparently non-optimal path chosen by an operator for the purpose of keeping the robot clear of deep holes is not safely reproduced by simple waypoint-seeking navigation algorithm (at least, not without the extra effort of defining artificial waypoints representing safe locations around the obstacle). It is also desirable that the robot not require a map to be created, input to the system and then constantly updated. Furthermore, the acquisition of waypoints is not generalisable to the more general tasks required of maintenance robots at Levels 2 and 3. Simply marking a complex state as a desirable goal is very far from being able to subsequently generate it.

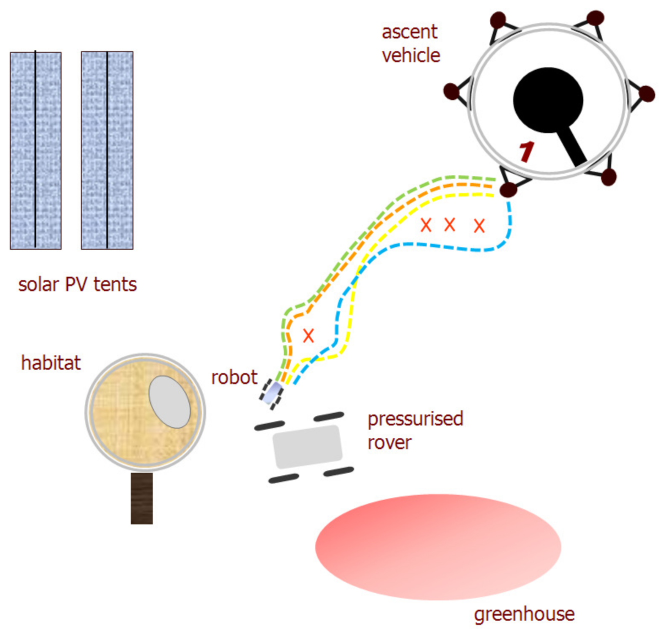

Figure 2. The first leg of a risky maintenance circuit at a Mars base. Over time, the initially human-guided robot tries multiple safe routes past obstacles to the first waypoint.

A different approach will therefore be pursued in this study, even though for this simple task that could probably be more efficiently accomplished by waypoint path finding: that of learning to reproduce the control actions of the human operator. This is, of course, not as simple as playing back a set of recorded controlling commands in sequence; the behavioural reproduction must be robust against positional and sensor uncertainty as well as unexpected disturbances such as moving objects. However, the literature on robot navigation clearly demonstrates that errors in orientation, which have the most serious consequences, can be corrected using local sensory information that provides an independent local bearing to readily-sensed invariant features of the environment, such as vertical lines extracted from doors and walls indoors [9] and trees outdoors [10]. Due to the length of time for recharges relative to operating time, as well as unfinished work on the sensors and steering of the robot, the Mascot machine is not yet ready for experiments of this kind, so despite the well-known risks, a simulated learning experiment will be discussed instead.

Consider a tour of interesting maintenance points around a circuitous route bringing the robot back to its starting point. Although the robot may encounter obstacles at any point on the route, on Mars it may also need to cover considerable distances containing few distinguishable local landmarks. It does not have a map of the area, and no GPS. It is however, equipped with a (non-magnetic) compass and sensors capable of detecting nearby obstacles. Using simple teleoperation as described above, the robot may be manually guided safely around such a route, while taking photographs and other maintenance measurements at nominated waypoints. Only first leg of such a tour will be simulated here (Figure 2), since the complete tour is essentially no more than a series of these. We wish the robot to gather information during only a small number of these guided tours, which it could then use to learn a safe route for autonomous movement. (This constraint of avoiding large numbers of training examples eliminates many potential machine learning techniques that tend to require much training data, such as conventional feedforward neural network approaches. The ideal would be *one-trial* learning.) Therefore, data will be logged during only a few successful manual runs will be available.

This simulation is not designed to model the dynamics of this kind of hexapodal movement; that has already been well-studied by Sanranli et.al.[1]. The motion model used here is simplified relative to the real motion of the Mascot robot in that it provides only a differential drive signal to virtual actuators on either side of the 'body', which turn it as if they drove wheels or tracks. The focus here is rather on the problem of fast, mapless acquisition of routes to waypoints which does

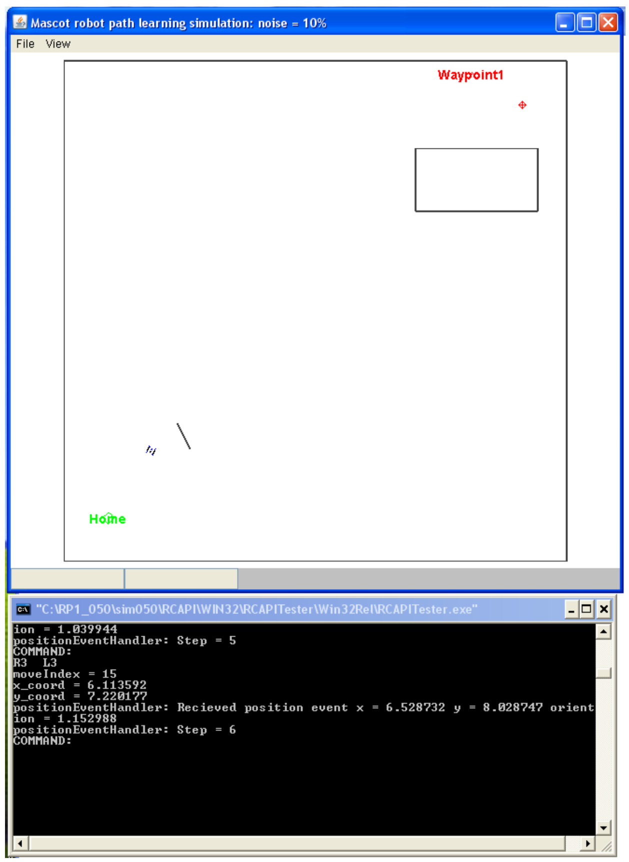

Figure 3. Screenshot of RP1 simulation of a single leg of a maintenance tour by the Mascot robot. Manual control signals input to the simulation Guide the robot initially; after

not crucially depend on the details of locomotion for a particular robot. To keep the simulation honest, only information which could be available to the robot was used, the commanded motions were equivalent to the motion model used by the physical robot, and random noise of 10% of the maximum leg velocities was injected bilaterally to represent the expected uncertainty in position that results from this complex interaction between the legs and unknown surfaces. The implementation is a purpose-written C++ client to the Java server provided by the open-source

Rossum's Playhouse simulation package, version 0.50 [11]. The client passes requests for data or commands to move the simulated robot to the server, which either succeed or fail depending on the interactions within the model environment. The server asynchronously generates expected and unexpected events which are passed back to handlers in the client code. The code ran in real time during the manual guidance training for the operator's sake, but was speeded up by a factor of 5 during learning trails to complete experiments more quickly.

4 Results and Conclusion

The approach favoured here is that of playback-and-correction. This has the advantages of great simplicity, a low requirement for training data, low requirements for map and sensors, and options for generating the corrective feedback. It is also possible that the general principles, though not the specifics, of the method will also be applicable to the acquisition of skills other than navigation. The first step is to characterise the errors that disturb the reproduced path during playback of a successful manual command sequence. The mean Euclidean distances between recorded and current x,y coordinates (displacement error) and the mean differences between recorded and current orientations were taken during multiple playbacks of the first 50 movement units is shown in Figure 4. Each movement unit is one of seven combinations of fixed (forward) left and right leg speeds, yielding a family of curved turns ranging from a straight line 2m in length to sharp curves to left and right.

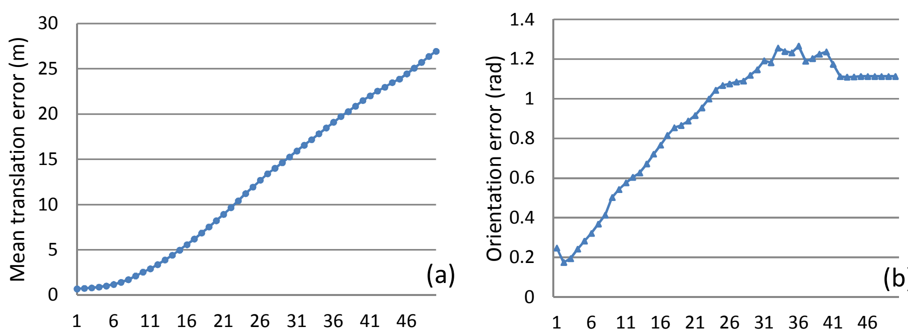

Figure 4. a) Mean x-y translation error (a) and mean orientation error (b) accumulated over the first 50 movement units (10% noise).

At the 10% noise level, small random orientation errors tend to accumulate at a mean rate of 0.02 rad./step; the x-y position of the robot departs monotonically from the ideal at a mean rate of 0.53m per step. Any navigational corrective feedback must exceed these stepwise rates to be effective.

The second step is to find a practical principle for correcting these errors. We assume that a map and coordinate system and that the Cartesian coordinates of the robot are not available for use. (In the simulation, coordinates are not used in the navigation logic, though they are for evaluation purposes). Orientation is the only correction possible at the end of each unit of movement. It might seem that a compass would be unable to make the needed corrections, since although at each interval the current absolute bearing could be compared to its logged equivalent, the x-y translation error will be unknown, and so the needed course correction would be theoretically impossible to calculate, on account of the parallel nature of compass needles over a (locally) wide range of different positions. However, it is possible that in practice, if each unit of movement is kept small that is, the intervals between successive readings is kept short as in this simulation (< 2m), the accumulated error will be tolerable. Also, if possible movement commands are deliberately restricted to a small number of standardized turns (as is the case in this simulation), then it might be possible associate an actual measured characteristic translational offset with each and then use these to modify the compass correction, at least on known, regular surfaces. So "compass-only" correction might still be practical under certain conditions. In this simulation, the difference between the recorded and observed orientations is used as an error signal that corrects the actual orientation after each movement step.

A second option would be to use the bearing to a local landmark to correct the orientation. Because the angles to this from the robot will change for both orientation *and* translation, it is possible to compute a more suitable course correction. The robot would have a sensor capable of measuring the bearing (and possibly range) to at least one fixed target that was constantly visible, such as a rotating laser and a corner reflector on a tower. Using several such landmarks would protect against accidental occlusion or faults without greatly complicating the error calculations. (The RP1 software did not support the use of multiple targets for its omnidirectional sensor, so only one is used in these simulations.) The landmarks could even be natural features of the environment detected by a vision system as in [12]. Early tests of this method showed that the error terms (recorded minus observed landmark bearings) was a powerful steering signal, but suffered from occasional large disturbances as the landmark bearing crossed a discontinuity in the measurement scale from $-\pi$ to π. At these anomalies in the landmark method, the compass error term was substituted.

The third step is to write code to test these competing correcting principles – compass-only and compass-plus-landmarks – by measuring the overall efficiency of navigation using each method. In practice, real compasses are imperfect. The physical Mascot robot uses the magnetic compass within a Phidget Spatial 3/3/3 device, which could be affected by stray magnetic fields to an unknown degree. To model this, random noise of 2% of the maximum observed orientation changes produced by perfect measurements is added to the simulated readings. The same disturbance is applied to the target sensor bearing generated in the compass-plus-landmarks algorithm.

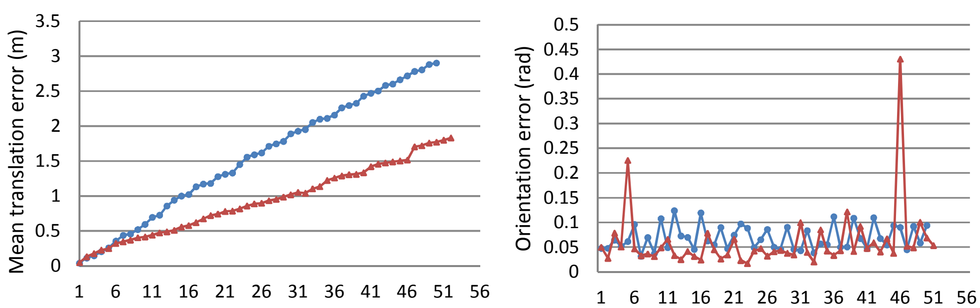

Figure 5. a) Mean x-y translation error (a) and mean orientation error (b) for compass-only (●) and compass-plus-landmark (■) orientation corrections accumulated over the first 50 movement units (10% motion noise, 2% sensor noise).

Figure 5 compares the two orientation correction methods. Compass-only correction shows an approximately an order of magnitude improvement in orientation accuracy relative to Figure 4b and the errors do not accumulate dramatically. The translation errors accumulate but are reduced by about an order of magnitude. The landmark-plus-compass corrections method accumulate translation errors at a slower rate, as expected, achieving a final error of 62% of the compass-only value. Orientation errors were generally of similar magnitude in the both methods, except for the two large peaks, which represent easily-detectable anomalies in the landmark corrections, and which the algorithm avoids in favour of compass-correction.

The resulting efficiency of navigation was assessed by comparing the robot's path to an optimal path (an approximately straight path from Home to Waypoint 1 with minimal skirting of obstacles) using a simple Sum of Squared Errors of estimation (SSE), here based on the Euclidean distance between corresponding pairs of path points. If the number of steps in the robot's path exceeds that of the optimal path, the distance between each extra point and the target position is used, thus imposing a penalty for redundant motion. A perfectly executed optimal path would return a SSE of zero; a far from optimal path, moving the robot around the walls of the simulated floor space to the waypoint might return an SSE of about 30,000, while a random path might result in an error of

100,000. The mean SSE of ten paths corrected by compass only was 234.9, while the corresponding mean SSE for compass-plus-landmark method was 87.58.

Although the landmark-plus-compass based method shows superior navigation performance, there are other important reasons to employ visual landmarking in this application. First, the system will need to employ cameras to take maintenance photographs, which to be taken from similar poses on each patrol. Therefore, employing visual navigation does not require the addition of more sensory equipment, and the visual landmarking approach can potentially position the camera repeatably. Second, as Govannangeli et.al. [12] have shown, distinguishable invariant features of the visual environment suitable landmarking can be automatically extracted during navigation using standard convolutions of camera imagery, and that their appearances from different positions alter in systematic ways that can be used to guide a robot close to a target object.

Generalising beyond the navigation task, we can envisage applying playback-and-correction to Level 2 tasks such as using a manipulator to open an access panel, remove a faulty component and replace it with a working one: i) characterise the errors which would accumulate during playback of an ideal manipulator sequence; ii) identify practical correcting principles which could reduce those errors iii) build sensory and software systems that use these to estimate and correct these errors.

Acknowledgements. This research was supported by RCF grant 07177 from The Division of Research and Development, Murdoch University.

References

1. Scammell, R. Mars Exploration Rover Technical Data. Available at http://hobbiton.thisside.net/rovermanual/
2. Saranli, U., Buehler, M., & Koditschek, D..: Rhex – A Simple and Highly Mobile Hexapod Robot. International Journal of Robotics Research, 20, 7, pp.616–631 (2001).
3. Saranli, U. & Koditschek, D. Template Based Control of Hexapedal Running. Proc. *of* the 2003 lEEE International Conference on Robotics & Automation, Taipei, Taiwan, Sept. 14-19 (2003)
4. Dorais, G. A., Bonasso, R. P., Kortenkamp, D., Pell, B. and Schreckenghost, D. Adjustable Autonomy for Human-centered Autonomous Systems on Mars. Proc. of the First International Conference of the Mars Society, Boulder, USA (1998)
5. Mann, G.A.: Quantitative Evaluation of Human-Robot Options for Maintenance Tasks during Analogue Surface Operations**.** Proceedings of the 8th Australian Mars Exploration Conference, University of South Australia, Adelaide, Australia. (2008)
6. Nickels, K., Kennedy, B., Aghazarian, H., Collins, C., Garrett, M., Magnone, L., Okon, A. & Townsend, J. Vision-Guided Self-Alignment and Manipulation in a Walking Robot. Proc. of the 2006 IEEE International Conference on Systems Engineering, Los Angeles, USA April 24-26. (2006)
7. Ohno, K. et. al. Outdoor Navigation of a Mobile Robot Between Buildings Based on DGPS and Odometry Data Fusion" Proc. of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Tiawan, September, 1978-1984. (2003).
8. Trepagnier, P., Nagel, J., Kinney, P., Koutsourgeras, C., & Dooner, M. KAT-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain. Journal of Field Robotics, 23, 509–526. (2006)
9. Guerrero, J.J., Martinez-Cantin, R. & Sagües, C. Visual Map-less Navigation Based on Homographies. Journal of Robotic Systems, 22, 10, 569-581. (2005)
10. S. Maeyama, S., Ohya, A. & Yuta, S. Positioning by Tree Detection Sensor and Dead Reckoning for Outdoor Navigation of a Mobile Robot. Proc. of IEEE International Conference on Multisensor Fusion and Integration for Intelligent systems, 653-660. (1994)
11. Wilson, J. Cyber Robotics: Experimenting with Robot Simulation. Circuit Cellar, 141, April, pp.34-46. (2002)
12. Giovannangeli, C., Gaussier, P. & D´esilles, G. Robust Mapless Outdoor Vision-Based Navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, October, 3293 – 3300. (2006)