Reconfigurable robots for all terrain exploration

P. S. Schenker, P. Pirjanian, B. Balaram, K. S. Ali, A. Trebi-Ollennu, T. L. Huntsberger, H. Aghazarian, B. A. Kennedy and E. T. Baumgartner, Jet Propulsion Laboratory; K. Iagnemma, A. Rzepniewski, and S. Dubowsky, Massachusetts Institute of Technology; P. C. Leger and D. Apostolopoulos, Carnegie Mellon University; G. T. McKee, University of Reading (UK)

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive/MS 125-224
Pasadena, California 91109-8099
paul.s.schenker@jpl.nasa.gov

ABSTRACT

While significant recent progress has been made in development of mobile robots for planetary surface exploration, there remain major challenges. These include increased autonomy of operation, traverse of challenging terrain, and fault-tolerance under long, unattended periods of use. We have begun work which addresses some of these issues, with an initial focus on problems of "high risk access," that is, autonomous roving over highly variable, rough terrain. This is a dual problem of sensing those conditions which require rover adaptation, and controlling the rover actions so as to implement this adaptation in a well understood way (relative to metrics of rover stability, traction, power utilization, etc.). Our work progresses along several related technical lines: 1) development a fused state estimator which robustly integrates internal rover state and externally sensed environmental information to provide accurate "configuration" information; 2) kinematic and dynamical stability analysis of such configurations so as to determine "predicts" for a needed change of control regime (e.g., traction control, active c.g. positioning, rover shoulder stance/pose); 3) definition and implementation of a behavior-based control architecture and action-selection strategy which autonomously sequences multi-level rover controls and reconfiguration. We report on these developments, both software simulations and hardware experimentation. Experiments include reconfigurable control of JPL's Sample Return Rover geometry and motion during its autonomous traverse over simulated Mars terrain.

Keywords: mobile robots, reconfigurable robots, modular robots, Mars rovers, robot architectures, robot control, intelligent control, sensor fusion, fused state estimation, field robotics

1. INTRODUCTION

Recent development of planetary rovers [1] has provided capabilities for semi-autonomous robotic traverse over relatively benign terrain. For a conventional wheeled rover, this usually means mobility over continuous natural surfaces having area rock densities of 5-to-10%, modest inclines (<30%), and a hard base with modest soft debris or sand pack (i.e., good floatation properties relative to wheel pressures of current rigid chassis designs). "Semi-autonomous" operation means the rover is sequenced by remote commands--with extensive time delay, in planetary cases--at a level of abstraction consistent with a human operator designating a target, a nominal path of traverse to same, and the rover then executing sensor-guided local obstacle detection/avoidance and overall path navigation via on-board computation, possibly carrying out locally optimal path planning in route. (See [2, and refs. therein] on related topics in semi-autonomy.)

Figure 1. JPL *FIDO Rover* during a terrestrial mission simulation of Mars science activities [3, Schenker et al.]

Rover navigation capabilities vary with system design and application objectives, but typically include some form of inertial guidance, and for planetary surface use, reference to celestial coordinates by sun tracking. Collectively, the above capabilities provide a basis for planetary mobile science in which the rover acts as a "remote geologist"--e.g., as illustrated in **Figure 1**, can perform multi-kilometer explorations with a coordinated suite of imaging and in-situ instrumentation (panoramic color viewing, navigational imaging, pointing and/or deployed spectroscopy, instrument arm mounted cameras, rock sampling tools and the like [3, and see also http://robotics.jpl.nasa.gov/tasks/etrover/homepage.html regarding related FIDO tests]).

There remain major rover design challenges which include increased autonomy of operation (e.g., capability for long ranging, accurate and uninterrupted autonomous traverse), self-guided traverse and mobility in more challenging terrain, and operational fault-tolerance under long, unattended periods of use, with failure recovery modes. Complementing these thrusts are altogether different system paradigms for planetary mobility, e.g., cooperative robot work by multiple platforms (cf. robot work crews [4]). We have begun research that addresses some of these issues. *Our focus in this paper is on the problems of "high risk access"-- extending autonomous roving to highly variable, rough terrain*. A primary motivation for this work is to enable exploration of potentially important science sites currently beyond reach of conventional rover design. As one example, recently reported remote imagery of the Martian surface suggests that water resources, if they existed, may be concentrated near cliff edge outflows that will require aggressive mobility strategies to explore in depth. To some degree, the "high risk access" challenge can be addressed by mechanization—scaling vehicle geometry or motility design to finesse the problem (within mass/volume/power limitations)—examples include robots employing legged locomotion, large inflatable tires, et al. Here, we look at a more fundamental issue: having the robot, whatever its design, recognize adverse terrain conditions beyond its nominal operational envelope, and intelligently adapt its mobility strategies. This is a dual problem of sensing the conditions that require rover adaptation, and controlling the rover actions as to implement this adaptation in a well understood way (relative to metrics of rover stability, traction, power utilization, etc.). Our work progresses along several related technical lines: 1) development of a ***fused state estimator*** which robustly integrates internal rover state and externally observed environmental information to provide accurate "configuration" information; 2***) detailed physical analysis of such configurations*** (kinematics and dynamics) so as to determine predicts for a needed change of control regime (e.g., traction control, active c.g. positioning, rover shoulder stance/pose); 3) definition and implementation of a behavior-based ***control architecture and action-selection strategy*** that autonomously sequences adaptive multi-level controls and changes in robot physical configuration. We report on these developments, both software simulations and hardware-based experimentation. Experiments include reconfigurable control of the JPL Sample Return Rover during its autonomous traverse over Mars-like terrain (SRR, as later described, is a small, 10 Kg class planetary rover with actively controlled shoulder pose, a positionable arm/c.g., and high performance real-time stereo navigation and obstacle avoidance.) Section 2 describes the all-terrain system concept. Section 3 overviews the system architecture and underlying sensing and controls. Section 4 reports some recent experimentation. Section 5 concludes with comments on some further planned directions of development.

2. SYSTEM CONCEPT

In this section we discuss our research agenda in terms of specific system concepts we are seeking to develop, and the technical design issues that underly them. Our thrusts are two-fold, one for the shorter term focused on development of a single robot for improved mobility at decresased operational risk, and the other targeting a longer range vision of a task-adaptive multi-robot reconfigurable system. In both cases we are working toward access and *in situ* science at high-value, high-risk planetary environments; in each case, reconfigurability pertains to re-mapping of both software and hardware resources. The near-in objective is, in summary, to develop an advanced science rover concept, one that by its sensor-based control adaptation and reconfiguration faciliates access to challenging locations such as escarpments, fissures, breakout channels, cliffs, etc.--places thought to be very goelogically rich for Mars exploration. The longer range objective is a broader-based, resuable modular system framework for closely coupled operations of heterogeneous agents—a "*networked robotics*" design [5]—a system enabling functions such as wide-baseline fused sensory observations, distributed search-mapping-and-navigation, and collective physical tasking (cliff rappells, tethering, etc.) such as those shown in **Figure 2**. The longer-range technology concept is thus in summary a high-level modular system (see, for example [6]) consisting of physically separable, reconnectable robotic agents with distributed, reprogrammable cooperative control and coordination capabilities. These new capabilities, either a single or multi-robot system, would clearly have multiple applications in field robotics beyond space exploration, e.g., to military surveillance & reconnaisance and urban search & rescue.

Overall, our two developmental thrusts emphasize *robot-level* modularity and reconfigurability. This can be contrasted and compared to more conventional notions of low-level modular reconfigurable robots [7], systems that are implemented at the granularity of primitive components/functional blocks. Both approaches are potentially important to space applications, the latter with reference to design and repair from inventory, as well as inherently restructurable mobility & actuation (e.g., the robot, with appropriate lower-level controls, recasts/self-transforms itself at large into a new class / shape / drive-train).

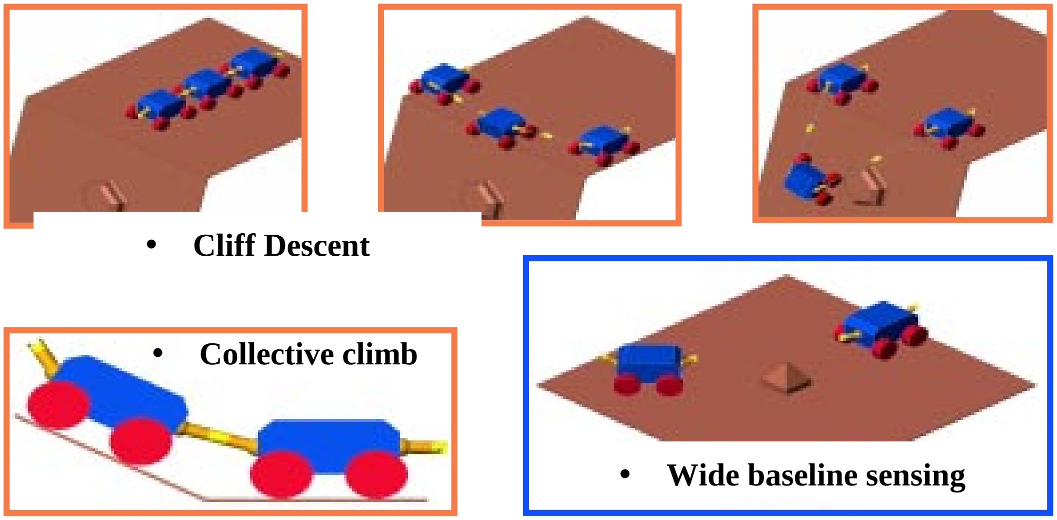
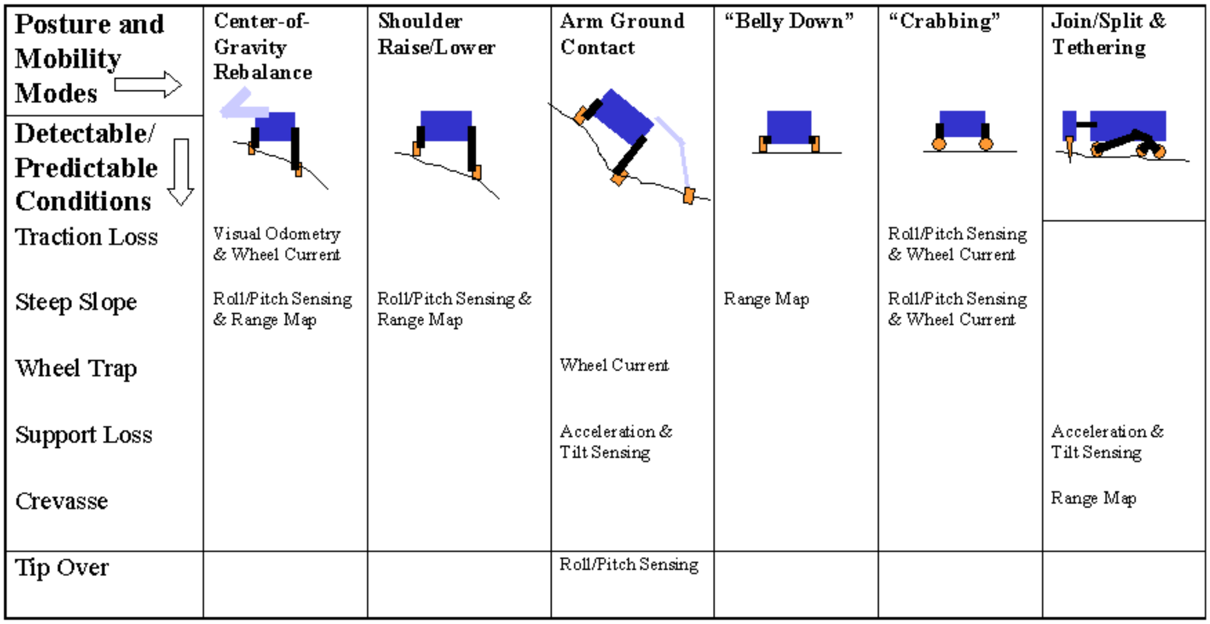


Figure 2. System concept illustrations: Tethering for collaborative cliff descent. Push-Pull mobility to enable access to rough and hilly terrain. Distributed sensing and wide-baseline observation.

For the majority of work that we report here, our focus lies with two problems, as illustrated in **Figure 3**: 1) <u>On-board integrated state estimation</u> that robustly identifies the internally measured and externally sensed parameters of vehicle motion/pose/articulation (including any multi-robot cooperative interactions), and relates this information to stable regimes of vehicular operation in natural, complex terrain; 2) <u>real-time reconfigurable control</u> that is behaviorally coordinated (based on predictive state estimates, and on-board physical-based planning, as available) to alter vehicle geometry and motion parameters in response to changing environmental conditions.



Posture and Mobility Modes ⇨	**Center-of-Gravity Rebalance**	**Shoulder Raise/Lower**	**Arm Ground Contact**	**"Belly Down"**	**"Crabbing"**	**Join/Split & Tethering**
Detectable/ Predictable Conditions ⇩						
Traction Loss	Visual Odometry & Wheel Current				Roll/Pitch Sensing & Wheel Current	
Steep Slope	Roll/Pitch Sensing & Range Map	Roll/Pitch Sensing & Range Map		Range Map	Roll/Pitch Sensing & Wheel Current	
Wheel Trap			Wheel Current			
Support Loss			Acceleration & Tilt Sensing			Acceleration & Tilt Sensing
Crevasse						Range Map
Tip Over			Roll/Pitch Sensing			

Figure 3. System conditions motivating rover reconfiguration (sensory "trigger" conditions)

3. SYSTEM ARCHITECTURE

In this section we first describe a sensor-based control architecture that is the basis for our longer ranging development of modular and reconfigurable robotic systems. We then overview our ongoing development of a fused state estimator and behavior based control for a single rover, as implemented on the JPL Sample Return Rover (SRR) [3], shown in **Figure 4**.

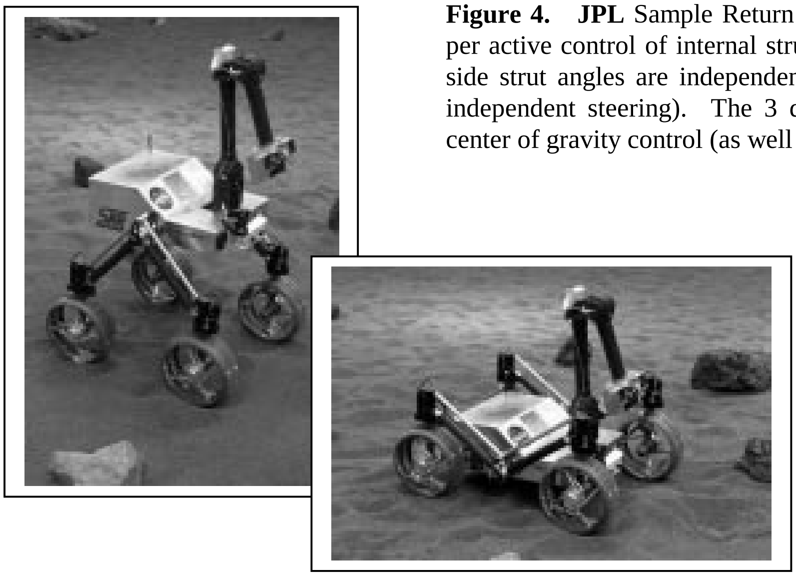


Figure 4. JPL Sample Return Rover (SRR) shown in two extreme poses, per active control of internal strut-axis differential drive. The left and right side strut angles are independently controllable, as are all wheel axes (for independent steering). The 3 d.o.f. composite arm can be positioned for center of gravity control (as well as usual science/sample transfer functions).

3.1 System Architecture

The distributed control and closely coupled coordination of a set of modular, re-taskable hardware/software entities and components requires an architecture and developmental infrastructure that is flexible, very reliable, and scales well with task-system complexity. For this purpose we are developing and extending a design called ***CAMPOUT*** (Control Architecture for Multi-Robot Planetary Outposts), as motivated by some related research we are conducting on cooperating *robot work crews* [4]. CAMPOUT provides a structured approach to the design, specification, implementation and validation of a complex control system and its subsystems, indeed, generalizes to problem solving in the control of a diverse class of heterogeneous multi-agent systems. CAMPOUT is built on a behavior-based philosophy for structuring control problems in a modular set of action-producing modules called behaviors [8], and imposes compositional constraints that guide the way the control problem can be solved. **Figure 5** gives a high-level overview of the control architecture. The diagram does not/cannot represent the whole architecture. Further details about the architecture, its components, interfaces between components, etc., and applicable references for its design rationale (including relationship to other published architectures) are given in [9].

CAMPOUT is a *hybrid reactive/deliberative architecture* incorporating higher-level constructs for task-level planning/ decomposition of activities under finite resource and goal constraints, and lower-level composition, coordination, and sequencing of behaviors for reactive control in tight perception-action feedback loops. The following is a brief description of the main characteristics of CAMPOUT:

- **Hybrid**: The architecture that we propose is characterized as "hybrid" within the realm of current (behavior-based) approaches to robot control. Hybrid architectures are viewed to provide the most general type of control due to their combining low-level reactive components with high-level deliberative planners.
- **Behavior-based:** Due to its generality and demonstrated success to date, a hierarchical behavior-based paradigm was chosen as the focus for our design of reactive/real-time cooperative controls; this appears a realistic starting point for the computation-and-memory constrained environment of space robotics. While a behavior-based approach is best suited to implementing lower-level reactive aspects of an architecture, the above-noted layering of planning over a behavioral level of control is gaining more acceptance within the behavior-based system design community [8].

- **Distributed:** The approach we are proposing is highly distributed. First, behaviors within a single robot operate in a distributed manner thus allowing for concurrent and/or parallel execution of several tasks. Second, each robot can operate on its own, independently of other robots, based on its embedded faculties of perception and action. Cooperation between multiple robots occurs through active collaboration, *with no centralized planning or decision-making component to dictate explicit commands*. The advantages of such truly distributed control and coordination include efficient use of system resources, parallel execution of multiple tasks, reliability and fault-tolerance to failure of individual components (including failure of single robots at large).

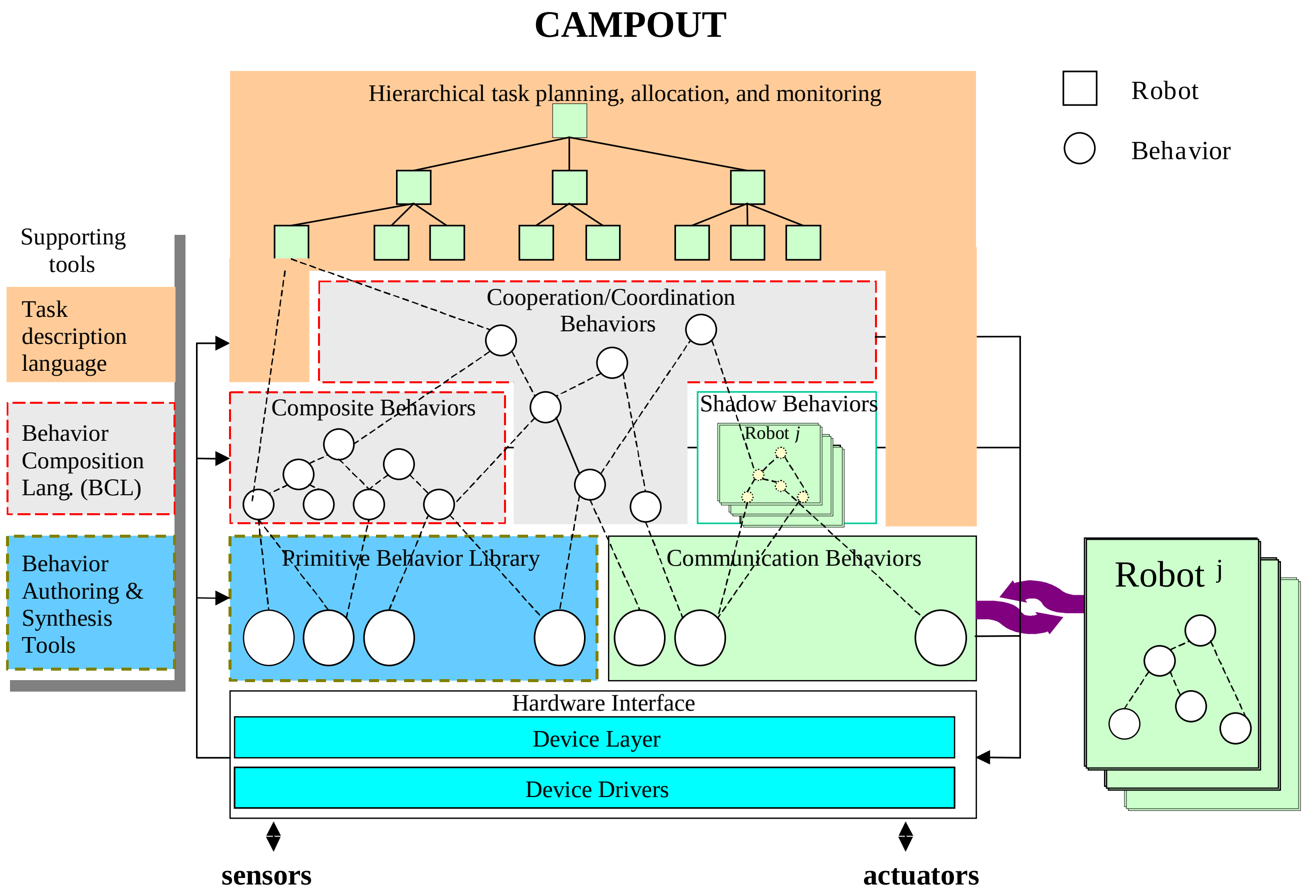


Figure 5. A logical block diagram of the Control Architecture for Multi-Robot Planetary Outposts (*CAMPOUT*)--its components, interaction between components, interfaces, and developmental tools

Beyond its philosophical design framework, CAMPOUT provides a set of tools that support development of implementations that conforming to the CAMPOUT methodology. These tools include:

1. **communications infrastructure** for information exchange between system components/robots, sharing of information (e.g., sensory data) across robots, and for behavior coordination across a network of robots
2. **facilities for construction of behaviors** (a fuzzy inference engine is currently provided for rapid prototyping of behaviors)
3. **behavior coordination mechanisms** (see [10] for an overview) for behavior arbitration and command fusion between inter- and intra-robot behaviors
4. **support tools** for interactive test and monitoring of system state.

3.2 State Estimation

State Estimation can be Current, Smoothed or Predictive in nature. We have discussed obtaining Current and Smoothed robot state estimates using Extended Kalman Filtering elsewhere [11, 12]; here, our interest is in predicting the future state of the rover based upon look-ahead stereo range imaging. This information is used to extract a *robot tip-over stability, slip and traction* ***Locomotion Metric***, which is then in turn used to determine possible and appropriate reconfigurations of the rover geometry and center-of-mass. The approach to obtaining this metric consists of the following steps:

1. Determine the surface shape of the terrain ahead of the rover.
2. Solve the Configuration Kinematics to predict the rover kinematic configuration on the terrain i.e. roll, pitch, yaw, internal angles and wheel contact points.
3. Given a friction coefficient characterizing wheel-ground interactions, determine if the span of the nominal frictional and normal forces at the predicted contact suffices to resist the gravity wrench (and any other disturbance forces) in both nominal and re-configured rover configurations (here the re-configuration consists of independent left-right shoulder angle changes and center-of-gravity shifts using the on-board manipulator).
4. Determine the minimum coefficient of friction in Step 3 which can then be taken to be a Locomotion Metric indicative of the quality of the configuration (or reconfiguration).

Step 1 is achieved by stereo imaging, correlating Laplacian images along epi-polar lines to establish disparity, and consequently the range, via a camera model. Step 2 is by means of an iterative Newton Solver [13]. The solution to Step 3 involves setting up polyhedral inequality approximations to the friction cone at each contact point, and expressing as inequalities the unidirectional constraints on the wheel normal forces and the wheel torque constraints. These linear relationships are then transformed to the vehicle frame using the vehicle Locomotion Matrix. An equality constraint characterizes the manifold of contact forces able to resist the applied wrench without regard to constraints. A linear programming solution uses these inequality and equality constraints to determine if a feasible set of friction and normal forces exists to resist the applied wrench. A binary search algorithm then computes the metric by determining the smallest value of friction coefficient that suffices to resist the applied vehicle wrench.

Note that this metric is sensitive to the fact that *the most stable configuration (which implies using an infinite coefficient of friction) may not be the most advantageous from the viewpoint of slip or traction*. Indeed, configurations that concentrate the weight on the "flatter" parts of the terrain are to be preferred, trading stability for slip resistance. If the vehicle is unstable, then even an infinite friction coefficient is unable to generate the resisting forces resulting in the Locomotion Metric being infinity. A finite value of the Locomotion Metric indicates that *sliding (or loss of traction) is inevitable* if the terrain/wheel coefficient of friction drops below the value indicated in the metric. Availability of the metric allows the current configuration of the vehicle shoulders and center-of-mass to be compared to adjacent configurations. The configuration with the lowest possible metric is a candidate for a vehicle reconfiguration and is recommended to the vehicle on-board controller.

As the sequence of operations indicated above is numerically quite demanding (though well within the capacity of our SRR test vehicle, whose computational features are summarized in Section 4), we have successfully experimented with a table-lookup formulation of the Locomotion Metric which is computed off-line and stored on-board for real-time use. In this approach we pre-compute the value of the metric for a broad range of terrain shapes (quadratic spline with 5 basis functions), shoulder configurations (left and right shoulder angle), and center-of-mass location (x and y). The resulting table is used on-line in the following manner, as summarized in **Figure 6**:

1. Obtain a range map of the terrain in the direction of rover movement.
2. Fit a spline to the range data in the area surrounding the next desired rover location and along the desired heading angle of the desired motion.
3. Perform a search over the look-up table to find the rover configuration that gives the lowest value of the Locomotion Metric.

The rover uses two front-mounted cameras to obtain stereo images of the forward terrain. A range map of the terrain is generated from the images in the camera frame. The points in this map where no range data is available are discarded and the remaining range data is transformed into a frame that is coincident with the rover frame, and vertically aligned with the gravity vector using the accelerometer derived roll and pitch information. The range map of the forward terrain is used to generate a 5x5 grid of elevations at points immediately surrounding the next specified rover location. The elevation at each of these points is estimated by interpolating over the available range data generated in the previous step. The interpolation routine creates a Delaunay triangular grid with the existing range points as nodal points. It then generates a function (with C^1 continuity) that estimates the elevation at any point. The data in the 5x5 grid is rotated from the rover frame (with the z-axis pointing downward) to a similar frame with the z-axis pointing upward. A spline of degree 2 in the x and y-axis is fitted to this 5x5 elevation grid.

This spline, along with a representation of a rover configuration, can be used as inputs to the table look-up which returns the Locomotion Metric for that configuration and terrain. Interpolation over the table is performed to provide a return value when the inputs do not correspond exactly to the values used for constructing the table. A search is performed to find the rover configuration that minimizes the metric. This search examines configurations that are the combinations of some set points for each of the degrees-of-freedom for the configuration. The configuration with the minimum metric is recommended to the higher-level rover controller.

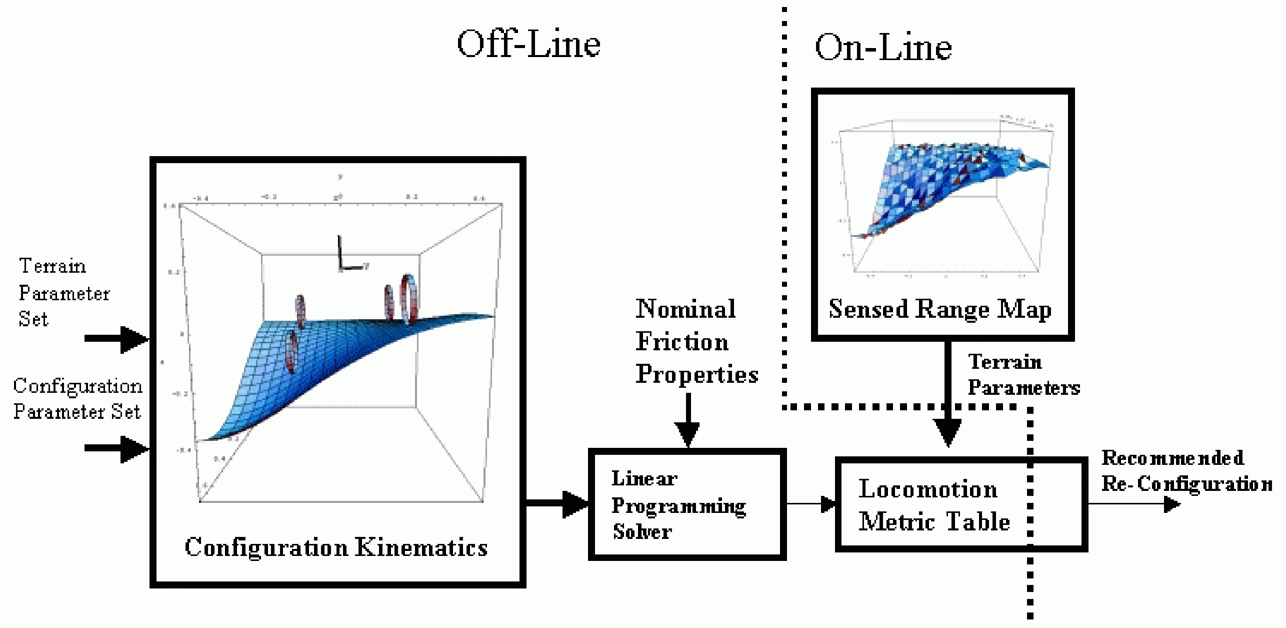


Figure 6. *Locomotion Metric* generation and use

3.3 Control Synthesis and Performance Evaluation

The first objective of our experimentation is to verify useful reconfiguration of rover motion parameters and geometry in response to a sensed terrain of highly variable, known topology and surface characteristics (viz., ground truth is available). A simple behavior-based control strategy is used for control and reconfiguration of the SRR (Sample Return Rover)—responding to both unstable pose and slip—as based on above described predictive state estimates of terrain slope from elevation maps produced by a binocular vision algorithm. The state estimator predicts the roll and pitch of the vehicle and calculates its stability. When approaching terrain for which the current configuration is not suitable, the behavior-based controller stops SRR, selecting the perceived most suitable configuration, and reconfigures the robot arm and rover shoulder postures accordingly before it proceeds. If the system fails to find a stable configuration, it will stop and report same.

As has been described, the predictive state estimation and rover reconfiguration employs off-line training using simulations to learn appropriate reconfiguration strategies for a representative set of states (a sparse representation of the reconfiguration space). The learned tables are used on-line to sequence reconfiguration behaviors. The terrain spline parameters, rover configuration, and X-Y/heading variables are inputs for look-up table generation, with the overall behavioral sequencing strategy implemented per **Figure 7** as a Finite State Machine (FSM). At the time of this report, we implement only a discrete state-based control of the rover configuration, as opposed to continuous-time adaptation.

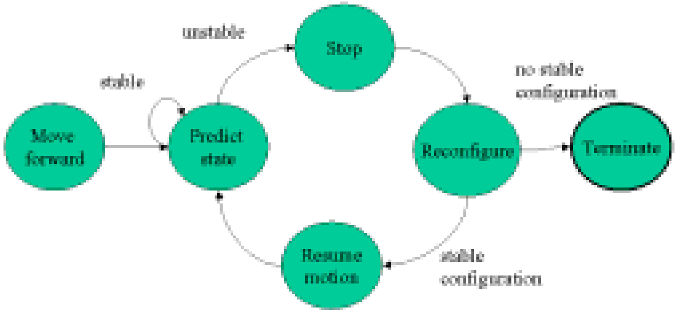


Figure 7. Basic control sequencing strategy for tilt-and-slip-based stability reconfiguration of SRR

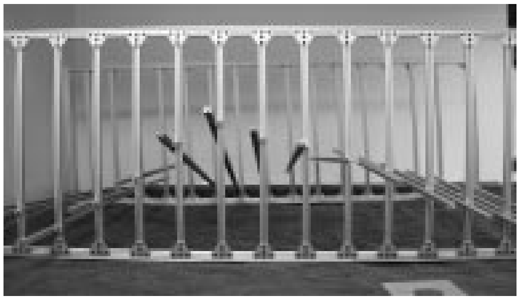


Figure 8. Section of adjustable track used in performance evaluation

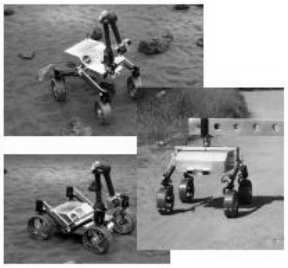
The performance of the above system is being experimentally evaluated (as to accuracy, repeatability, sensitivity analysis, etc.) and compared to that of the same rover running under conventional fixed parameter control. A user-reconfigurable test-track consisting of three eight-foot-long segments is employed for testing, as illustrated at left in **Figure 8**. The cross-bars can be configured to construct a “terrain” with known variable pitch, roll, and combined pitch and roll angles. The track is covered with a synthetic surface having appropriate markings to drive a high-fidelity visual depth map estimation process. This reconfigurable track thus can be used to construct test scenarios that systematically investigate system capabilities as a function of roll and pitch angles as well as critical transition areas between variable roll and pitch angles. We complement this activity with testing in the Arroyo Seco near JPL (a test site frequently used in our planetary rover simulations, providing a diversity of terrain and surface characteristics).

4. SYSTEM IMPLEMENTATION

Our initial experiments on rover reconfiguration are being implemented on a small autonomous rover that has evolved from recent JPL work on planetary surface exploration and sample return. The baseline rover design, the ***Sample Return Rover*** **(SRR)**, is reported in [3, Schenker], wherein it in past incorporated skid steering, independently poseable shoulders, and basic functionalities for stereo-based obstacle detection, continuous motion visually-guided traverse (10+ cm/sec), visually-servoed manipulation, and visual object detection, tracking, and rendezvous with man-made structures (also novel mechanical features of self-deploying collapsible wheels and primary struts, not immediately relevant to the problems being investigated here). More recently, and as summarized in **Figure 9**, SRR has been augmented with four-wheel independent steering, improved computational and sensing resources, and the above-noted behavioral CAMPOUT architecture that is being used both in this research and as well as other work on cooperating rovers [4]. Of particular interest, and complementing the work described in Section 3, is a recent MIT-led collaborative experiment with JPL (Dubowsky et al.), performed on SRR. We next summarize this study and some of its preliminary results. See also [14], for further details and background.

4.1 A simple approach to kinematic reconfigurability

As previously noted, SRR has been developed with the ability to actively modify its kinematic configuration to enhance rough terrain mobility. For example, when traversing an incline, SRR can lower one side of its suspension to increase its stability margin; it accomplishes this using two active shoulder joints (cf. Figure 4). SRR can also redistribute its center of mass by repositioning its manipulator.

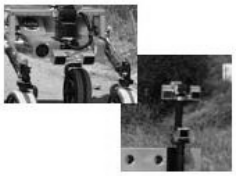


Mobility and Configuration Control:

- 4 wheel independent steering with full instrumentation and capability of up to 3 N-m and 3 rad/sec
- 20 cm dia. wheels with odometry capable of 19 N-m and 21 cm/sec
- Passive, instrumented, rocker-type suspension with active spur-gear differential articulated shoulder joint
- Parallel linkage on suspension enables simultaneous operation of articulated shoulder/passive rocker/steering

Multiple Rover Operations:

- Fully instrumented 4 DOF (pitch, roll, yaw, lateral translation) gimbal
- Compliant gripper for "soft-grip" of payload
- Interchangeable payload support beams to increase the load carrying capacity



Computing, Electronics:

- Pentium 266MHz/32MB, VxWorks 5.4, Solid State Disk (boot-able)
- 2x4-axis mot. ctrl., 2x640x480 color framegrabber, 12bitx16ch D/A
- Ethernet (~1.5 Mb/s) wireless modem; 24v battery pack, 1-1.5 hr.
- stereo b/w pair 120° FOV; arm-mounted stereo color pair 45° FOV; arm-mounted 20° FOV goal camera

Figure 9. Summary of JPL *Sample Return Rover* (***SRR***) features relevant to reconfigurable operations (as well as design elements supporting related cooperating multi-rover research, per Robot Work Crews [4])

Previous researchers have sugested the use of kinematic reconfigurability to enhance rough-terrain performance [15]. However, the practical effectiveness of this concept has not yet been demonstrated on a real rover in rough terrain. Here, one method for stability-based kinematic reconfigurability is presented, as applied to the JPL SRR. The method does not require a detailed map of the terrain, rather only local knowledge of the wheel-terrain contact angles, which are estimated using simple on-board sensors [16]. Computational requirements are relatively light, recognizant of limited on-board resources of planetary rovers. Computer simulation and initial experimental results under field conditions show that such kinematic reconfigurability can greatly improve rover stability in rough terrain.

The kinematic reconfigurability approach presented here relies on constructing a kinematic model of the vehicle in rough terrain, addressing tipover (Slip and frictional degrees of freedom are not addressed in this formulation). The key wheel-terrain contact angles are estimated using simple on-board sensors. Due to the slow speed of the rover as operated here (less than 6 cm/sec) only static forces are considered in calculating rover stability. System stability is expressed in terms of a set of stability angles. A stability angle is the angle formed between a line originating at the center of mass and normal to the tipover axis, and the gravitational (vertical) axis [17]. This angle goes to zero at marginal stability. A performance index, $\boldsymbol{\Phi}$, is defined for the SRR from these stability angles γ_j, $i=\{1,\ldots,4\}$, and the reconfigurable shoulder degrees of freedom, ψ_1 and ψ_2, as **Equation 1**:

$$\Phi = \sum_{j=1}^{4} \frac{K_j}{\gamma_j} + \sum_{i=1}^{2} K_{i+4} \left(\psi_i - \psi_i^{'} \right)^2 \qquad (1)$$

where K_i and K_j are positive constants and the stability angles γ_j are functions of the shoulder and manipulator degrees of freedom (i.e. $\gamma_j = \gamma_j(\psi_1, \psi_2, \theta_1, \theta_2, \theta_3)$). Note that the first term of Φ tends to infinity as the stability at any tipover axis tends to zero. The second term penalizes deviation from a nominal configuration of the shoulder joints, thus maintaining adequate ground clearance, an important consideration in rough terrain. The goal of the stability-based kinematic reconfigurability optimization problem is to *minimize the performance index* Φ *subject to joint-limit and interference constraints.* For rapid computation, and due to the simple nature of Φ, a basic optimization technique such as conjugate-gradient search is employed.

4.2. Experimental Results

Simulation and experimental studies were conducted of the SRR traversing rough terrain in both laboratory and outdoor environments. Results for a representative simulation trial is shown in **Figure 10 (left)**. In this simulation the manipulator and shoulder joints are reconfigured during the traverse. The stability margin of the fixed-configuration system reaches a minimum value of 1.1°, indicating that the system narrowly avoided tipover failure. The minimum stability margin of the reconfigurable system was 12.5°, a comfortable margin.

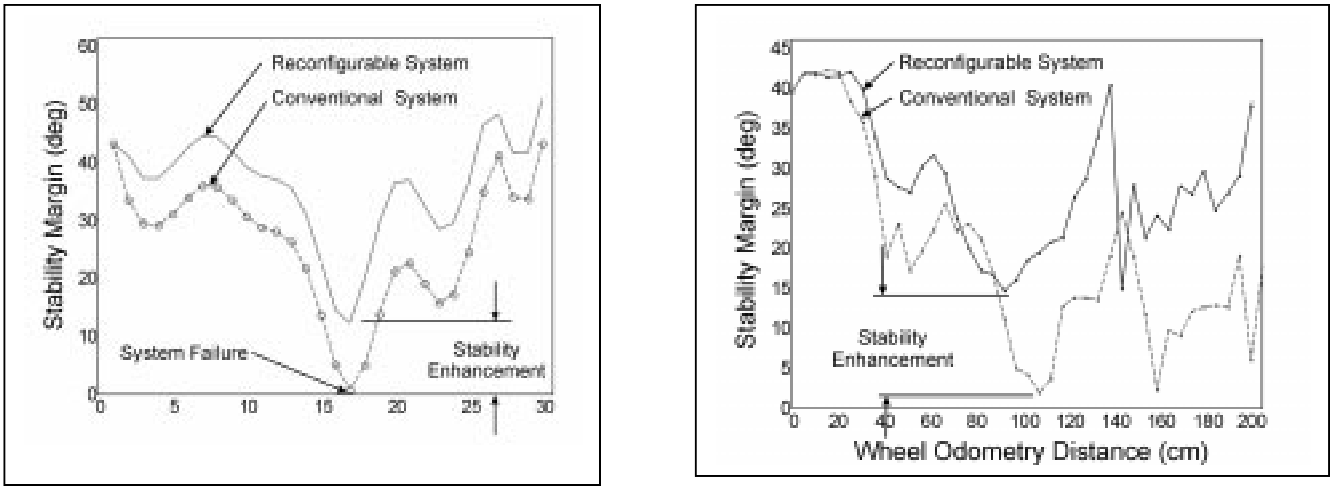


Figure 10. (*left*) Simulation: SRR stability margin for reconfigurable system (solid) and non-reconfigurable (dashed) system; (*right*) experimental data; (*below*) SRR reconfiguring its geometry in response to unstable terrain

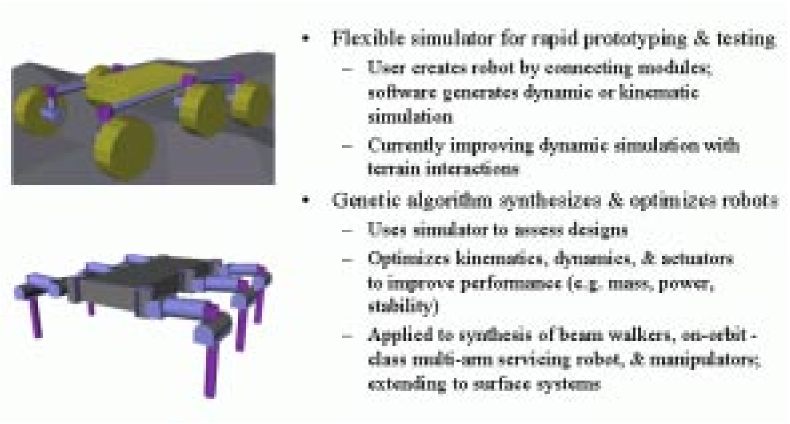
Experiments were then performed with SRR in the JPL Planetary Robotics Laboratory and the adjacent Arroyo Seco in Altadena, California by a joint MIT/JPL team of researchers. SRR was commanded to traverse a challenging rough-terrain path that threatened vehicle stability. For each trial the path was traversed first with the shoulder joints fixed, and then with the kinematic reconfigurability algorithm activated. Results of a representative trial are shown in **Figure 10 (right)**. The stability margin of the fixed-configuration system reaches dangerous minimum values of 2.1° and 2.5°. The minimum stability margin of the reconfigurable system was 15.0°. Clearly, kinematic reconfigurability can result in greatly improved vehicle stability in rough terrain, as illustrated at left.

5. FURTHER DEVELOPMENTS

What we have reported here is preliminary, particularly as regards the potential for a richer regime of behavior-based configuration controls that improve vehicular stability and cross-terrain performance. The techniques demonstrated are discrete-state, and do not at this time reflect any significant effort toward such issues as state-transition smoothing (e.g., exiting one regime and entering another). Dynamics (including beam flexion) are not considered, nor in the regimes (speed, terrain interactions) we are working do dynamics yet appear a driving issue, though certainly of interest. Nor do we address terrain characteristics in any depth, e.g., modeling of soil-tire interactions, including frangibility-sinkage, etc. (known hard problems, about which we are gathering experimental data via test apparata developed at JPL/MIT for that explicit purpose). Finally, what we report here—in these particular experimental implementations—reflects neither detailed physical planning or task-level deliberative planning. These are issues for future development, in line with CAMPOUT implementation and prior work of the investigators involved. In concluding, we touch on two related streams of ongoing investigation, both of which will foster our thrust toward higher-level modular reconfigurable design, as was discussed in Section 2.

5.1 Automated Synthesis of Complex Modular Robotic Systems

The design of future modular, reconfigurable rovers presents many challenges. Before pursuing a detailed design, it will be necessary to investigate many configuration-level issues: Should the individual system components (modules) be capable of totally independent operation? Should the system be composed of heterogeneous or homogeneous sub-robots and modules? Is walking, rolling, or a combination best suited for a reconfigurable mobile robot? How many wheels or legs are best, considering that the system may be able to share resources between sub-robots when necessary? If multiple independent robots are used, what sort of linkages should be use for connections? All of these issues must be decided, preferably in a way that maximizes systemic performance. We have made some significant progress in this direction, summarized in **Figure 11**.



- Flexible simulator for rapid prototyping & testing
  - User creates robot by connecting modules; software generates dynamic or kinematic simulation
  - Currently improving dynamic simulation with terrain interactions
- Genetic algorithm synthesizes & optimizes robots
  - Uses simulator to assess designs
  - Optimizes kinematics, dynamics, & actuators to improve performance (e.g. mass, power, stability)
  - Applied to synthesis of beam walkers, on-orbit - class multi-arm servicing robot, & manipulators; extending to surface systems

Figure 11. Summary of ongoing developments in automated mobility synthesis

The enormous resulting robot design spaces, coupled with limited human experience in relevant designs, makes automated design techniques very attractive. We propose to use ***Darwin2K***, a recently developed automated synthesis and optimization toolkit for robotics [18], to investigate the relative merits of different system concepts and potentially to synthesize a set of feasible, well-optimized configuration-level modular designs. Darwin2K uses a distributed evolutionary algorithm to synthesize robot designs from a set of parameterized modules and optimize robot performance with respect to multiple user-defined metrics and constraints. By altering the set of robot components and constraints supplied to the synthesis program, we can force it to focus on a particular conceptual design area (e.g. multiple identical wheeled rovers, or a modularly reconfigurable walker) and thereby create well-optimized configuration designs which quantify the strengths and weaknesses of the conceptual design. In this way, we can arrive at optimized designs for multiple design concepts and thus make an informed decision on conceptual design.

Applying Darwin2K to this class of design problem will require some improvements in the system's simulation capabilities. Presently, the system can automatically derive a dynamic model of a robot and perform forward and inverse dynamic simulations. The forward dynamic simulation can include bilateral constraints (such as closed kinematic chains) and unilateral constraints (such as contact and joint limits), but the simulation of frictional contact forces is a recent addition and is not robust enough for synthesis purposes. The current algorithm is based on a Linear Complementarity Problem (LCP) method [19], which unfortunately is not guaranteed to terminate when applied to systems modeling multiple three-dimensional frictional contacts. We plan to switch to a sequential quadratic programming (SQP) method [20] which should resolve this problem, and which is amenable to extension to non-rigid contact such as that between a rigid tire and compliant soil.

Several pieces of task-specific software will have to be implemented in order to use Darwin2K for synthesis of modular reconfigurable robots. We will first have to decide which additional parameterized modules are needed: while the system currently has a library of several dozen general-purpose modules for joints, links, tools, and so on, there may be special-purpose modules that are necessary for reconfigurable machines. Each new module requires a function that describes the kinematic, geometric, component (e.g. motor and gear-head), and structural properties of the module's physical instantiation based on the module's parameters, which are fine-tuned by the synthesizer since the optimal parameter values will not be known a priori. Relevant planning and control algorithms are also necessary, since the system uses simulation to evaluate the performance of each candidate design. Currently, the synthesis system control algorithms are fairly low-level, such as a PID joint-space controller and several Jacobian-based Cartesian controllers. However, it seems likely that higher-level planning or behavioral control modules will be useful for computing the requisite trajectories for reconfiguration and mobility, which will then be executed by the low-level controllers. It is worth noting that *the synthesizer has the ability to optimize controller parameters (those of both high- and low-level controllers)*, so it will not be necessary to manually select control parameters—a daunting task, were it so—as the controllers will be used in simulation with tens of thousands of diverse robots.

Finally, we must decide on a task description and *set of metrics for quantifying the performance of each robot*. It will be important to capture the desired capabilities (e.g. extreme mobility and reconfiguration) in the task description, since the synthesizer will evolve robots that are optimized for the particulars of the task specification. The simulated task may be a scenario in which the robot performs a sequence of operations such as traversing benign terrain, reconfiguring, and then ascending or descending a steep cliff face; or, it may be a series of short trials which quantify the robot's locomotion performance over and around obstacles and terrain of varying types. We will also have to select a relevant set of metrics that accurately capture the performance requirements for the robot. The metrics can embody both constraints (such as not tipping over or colliding with obstacles) or open-ended objectives such as reducing mass, power, and task time. More complex metrics including sensor visibility and excursion may also be desirable for this task.

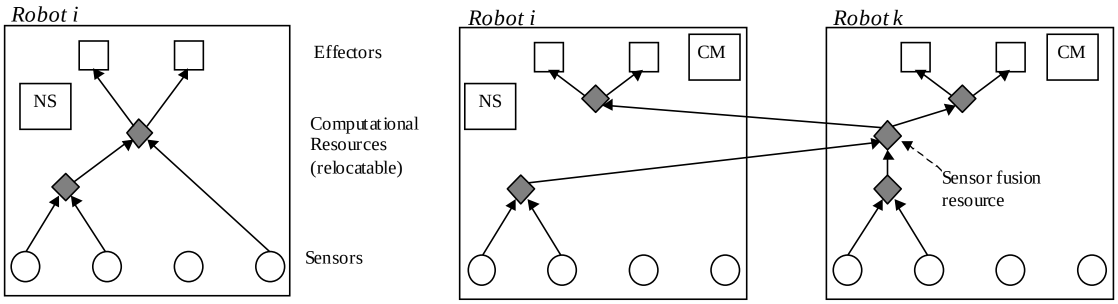
The bulk of the synthesis system is already extant and well-tested; the key challenges in this work will be the improvement of Darwin2K's algorithms for dynamic simulation with frictional contacts, and in developing relevant and flexible control algorithms for reconfigurable robots. Dynamic simulation with frictional contacts is an area of open research, and any robust solution will be broadly applicable. Development of the appropriate control algorithms will provide a baseline capability once the robot hardware has been built, and will serve as a starting point for controllers specialized for the final design.

5.2 Networked Robotics

The essence of networked robotics is the concept of distributed resources that provide one or more interactive services [5, 21, and references therein]. Sensors (vision, range, position), effectors (manipulators, mechatronic modules, grippers, mobile platforms) and computational units (fused state estimation, mapping, planning and navigation-control functions) are three basic categories of resource encountered in robotics. In conventional robot architectures, resources are not distinguished as such; sensors, effectors, and computational units are "hard-wired" components of a fixed, immutable algorithm, the control architecture. Networked robotics recognizes these functional units, enhances them with an interface that makes explicit services they can export, and incorporates scope for a range of local and remote connectivity options. The resources, now encapsulated as independent modules, provide the basis *for flexible re-configurable robot architectures that can span multiple physical robot systems, namely a networked robot*. Higher-level networked modules can in principle autonomously inherit attributes of lower-level resources, with emergent control and sensing properties, and accompanying new descriptors.

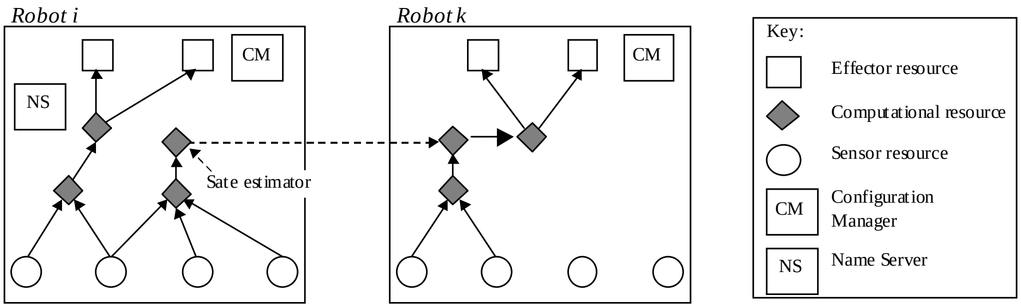
There are three basic elements of a general networked robotics system: a set of resources, a configuration definition specifying the topology and connectivity of the modules, and a name server holding location information for the resources (**Figure 12a**, next page). The latter is essential for connecting the modules together. A typical model for a multiple robot scenario would require individual modules to make themselves known to the name server on initialization. When the configuration is initialized it uses the name server to locate the modules and hence to effect the connections defining the configuration. In more basic, minimalist systems, resource locations may be fixed and therefore a name server may be unnecessary. In more complex systems the name server might also provide access control and resource monitoring functions, performing in this case the additional function of a registry, or repository of a range of information about resources.

Resources can be primitive or derived. Primitive resources, namely sensor and effector resources, generally have a fixed location since they provide direct interfaces to peripheral devices such as sensors and actuators. Computational and derived resources can be relocated. A derived resource corresponds to a sub-configuration comprising primitive and possibly other derived resources; e.g., a state estimator or an independent behavior can draw on a number of primitive resources. Derived resources support abstraction of the global configuration, that is the overall architecture, into functional or behavioral units. A configuration planner can define one or more configurations for a target system comprising one or more robotic platforms. A pre-planner can in turn establish a strategy for deployment of the resources and a tactical, *in-situ*, reorganization can be employed to accommodate runtime constraints on the relocation of computational resources. For example, a sensor fusion module could be located on any platform from which it draws data (**Figure 12b**). A state estimator in a multi-robot dynamic scenario, however, is most likely to be located on that platform providing a majority of strategic data (**Figure 12c**).



(a) Simple (flat) intra-platform configuration

(b) Single configuration spanning multiple robot platforms



(c) Cooperating robots – *robot i* assembles and forwards state about *robot k*. Multiple configurations.

Figure 12. Resources and their configuration, spanning multiple robotic platforms, underpins networked robotics

A *configuration manager* supports reconfiguration according to a *configuration state diagram (CSD)* and local, operational rules that trigger state changes. Multiple configuration managers are required for configurations that span multiple physical platforms that may be configured according to a cooperative multiple robot strategy. For a spanning configuration, on the other hand, a single configuration manager can link together the resources across multiple platforms. On a configuration change request the target configuration is identified and a deployment planner allocates sub-configurations to the local configuration managers along with a configuration change plan for partial or complete reconfiguration of the local resources. A global sequencing plan may be instantiated in order to take down the current configuration and bring up the new configuration in some well-defined pattern. A simple default strategy would allow resources to be initialized, ranging from primitive through progressively derived resources, across all the platforms.

The benefits of networked robotics lie in the ability to distribute, relocate and reconfigure set of resources arrayed across multiple physical mobile robot platforms into single or multiple cooperating robot systems, many properties of which are intrinsically realized in CAMPOUT (Section 3.1, Figure 5, [4, 9]). In a dual-platform scenario, for example, intra-platform and inter-platform sensor fusion can generate a composite state estimate of the robot system (**Figure 12b**). On difficult terrain the configuration could change to a cooperative dual robot/platform scenario where one robot provides state information for the second, and vice versa as the platform encountering the hostile terrain changes (**Figure 12c**). Tools to support definition, modeling & simulation of configurations, and planners for reconfiguration management are essential components of the networked robotics development environment. Networking models, multicast technology, and interfaces between real-time and non real-time networks need to be better explored in order to achieve lightweight re-configurable implementations with optimal or near-optimal real-time performance. Resource models, service interfaces, modularization and granularity are all key research issues in networked robotics [5].

ACKNOWLEDGEMENTS

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

1. C. R. Weisbin, G. Rodriguez, P. S. Schenker, E. Baumgartner, R. Volpe, S. Hayati, and H. Das, “Autonomous rover technology for Mars sample return,” Proc. i-SAIRAS '99 (5th International Symposium on Artificial Intelligence, Robotics and Automation in Space), 1-3 June 1999, Noordwijk, The Netherlands; R. Volpe, E. Baumgartner, P. Schenker, and S. Hayati, “Technology development and testing for enhanced Mars rover sample return operations,” Proceedings of the IEEE Aerospace Conference, March, 2000.

2. P. S. Schenker and G. T. McKee, “Man-machine interaction in telerobotic systems and issues in the architecture of intelligent and cooperative control,” in Proc. IEEE/Intl. Symp. Intelligent Control Workshop (Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems, Orgs.: J. Albus, A. Meystel, D. Pospelov, T. Reader), Monterey, CA, August, 1995.

3. P. S. Schenker, E. T. Baumgartner, R. A. Lindemann, H. Aghazarian, D. Q. Zhu, A. J. Ganino, L. F. Sword, M. S. Garrett, B. A. Kennedy, G. S. Hickey, A. S. Lai, and L. H. Matthies; Jet Propulsion Lab.; B. D. Hoffman, Massachusetts Inst. Technology; T. L. Huntsberger, Univ. So. Carolina, “New planetary rovers for long range Mars science and sample return,” Proc. SPIE Vol. 3522, Intelligent Robotics and Computer Vision XVII, Boston, MA, Nov. 1-5, 1998 (Invited, 14 pages), and references within; see also, T. L. Huntsberger, E. T. Baumgartner, H. Aghazarian, Y. Cheng, P. S. Schenker, P. C. Leger, K. D. Iagnemma, and S. Dubowsky, “Sensor-fused autonomous guidance of a mobile robot and applications to Mars sample return operations,” Proc. SPIE Vol. 3839, Sensor Fusion and Decentralized Control in Robotic Systems II, Boston, MA, pp. 2-8, Sep. 19-22, 1999.

4. P. S. Schenker, T. L. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H. Das, S. Joshi, H. Aghazarian, A. J. Ganino, B. A. Kennedy, and M. S. Garrett, “Robot work crews for planetary outposts: close cooperation and coordination of multiple mobile robots,” Proc. SPIE Vol. 4196, Sensor Fusion and Decentralized Control in Robotic Systems III, Boston, MA, Nov. 5-8, 2000 (*these proceedings*).

5. G. T. McKee and P. S. Schenker, “Networked robotics,” SPIE Vol. 4196 (*these proceedings*).

6. T. Fukuda, T. Ueyama, Y. Kawauchi, and F. Arai, “Concept of Cellular Robotics System (CEBOT) and basic strategies for its realization,” Computers in Electrical Engineering, Vol. 18, No. 1, pp. 11-39, 1992; see also, T. Fukuda et al., “Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator,” 1990 IEEE Intl. Conf. on Robotics and Automation (ICRA).

7. A. C. Sanderson, *TETROBOT: A Modular Approach to Reconfigurable Parallel Robotics*, Kluwer Academic Press, 1997; W. H. Lee and A. C. Sanderson, “Dynamic simulation of tetrahedron-based TETROBOT,” IROS’98; M. Yim, D. G. Duff, and K. D. Roufas, “PolyBot: a modular reconfigurable robot,” Proc. 2000 IEEE Intl. Conf. On Robotics and Automation (ICRA), and, A. Casal and M. Yim "Self-reconfiguration planning for a class of modular robots,” Proc. SPIE Vol. 3839, Sensor Fusion and Decentralized Control in Robotic Systems II, Boston, MA, pp. 2-8, Sep. 19-22, 1999; A. C. Sanderson, “Modular robotics: design and examples,” Proc. IEEE Conference on Emerging Technologies and Factory Automation, 1996; S. Farritor, S. Dubowsky, N. Rutman, and J. Cole, “A systems-level modular design approach to field robotics,” Proc. 1996 IEEE Intl. Conf. on Robotics and Automation (ICRA), also, Shane Farritor et al., “On the design of rapidly deployable field robotic systems,” ASME Conference on Design Methodology, Irvine, CA, 1996; C. Unsal, H. Kiliccote and P. K. Khosla, “I(CES)-Cubes: A modular self-reconfigurable bipartite robotic system; Proc. SPIE Vol. 3839, Sensor Fusion and Decentralized Control in Robotic Systems II, Boston, MA, pp. 2-8, Sep. 19-22, 1999.

8. Ronald C. Arkin, *Behavior-Based Robotics*, The MIT Press (Intelligent Robotics and Autonomous Agents series), 1998.

9. P. Pirjanian, T. L. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, H. Das, S. Joshi, and P.S. Schenker, “CAMPOUT: a control architecture for multi-robot planetary outposts,” SPIE Proceedings Vol. 4196 (*these proceedings*).

10. P. Pirjanian, “Behavior Coordination Mechanisms - State-of-the-art,” Tech-report IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, October, 1999; and, P. Pirjanian, “Multiple objective behavior-based control,” Journal of Robotics and Autonomous Systems, Vol. 31, Issues 1-2, pp. 53-60, April, 2000.

11. J. Balaram, “Kinematic state estimation for a Mars rover,” Robotica, Vol. 18, 251-262, 2000.

12. B. D. Hoffman, E. T. Baumgartner, T. Huntsberger, and P. S. Schenker, "Improved rover state estimation in challenging terrain," Autonomous Robots, Vol. 6, No. 2, pp. 113-130, 1999, and references therein; also, E. T. Baumgartner, H. Aghazarian, A. Trebi-Ollennu, T. L. Huntsberger, and M. S. Garrett, "State estimation and vehicle localization for the FIDO Rover," SPIE Vol. 4196 (*these proceedings*).
13. J. Yen, A. Jain and J. Balaram, "ROAMS: Rover analysis, modeling and simulation software," Proc. i-SAIRAS'99 (Fifth Intl. Symposium Artificial Intelligence and Automation in Space), Noordwijk, The Netherlands, 1-3 June 1999.
14. K. Iagnemma, A. Rzepniewski, S. Dubowsky, T. Huntsberger, and P. Schenker, "Mobile robot kinematic reconfigurability for rough-terrain," SPIE Proceedings Vol. 4196 (*these proceedings*).
15. S. Sreenivasan and B. Wilcox, "Stability and traction control of an actively actuated micro-rover," Journal of Robotic Systems, Vol. 11, no. 6, pp. 487-502, 1994; S. Sreenivasan and K. Waldron, "Displacement analysis of an actively articulated wheeled vehicle configuration with extensions to motion planning on uneven terrain," ASME Journal of Mechanical Design, Vol. 118, No. 2, pp. 312-317, 1996; S. Farritor, H. Hacot, and S. Dubowsky, "Physics-based planning for planetary exploration," Proc. 1998 IEEE Intl. Conf. Robotics and Automation (ICRA), Belgium, May 1998.
16. K. Iagnemma and S. Dubowsky, "Vehicle wheel-ground contact angle estimation: with application to mobile robot traction Control," Proceedings of the 7th International Symposium on Advances in Robot Kinematics, ARK '00, 2000.
17. E. Papadopoulos and D. Rey, "A new measure of tipover stability margin for mobile manipulators," Proc. 1996 IEEE Intl. Conf. Robotics and Automation (ICRA).
18. C. Leger. "Automated synthesis and optimization of robot configurations: an evolutionary approach,", Ph. D. Thesis, Carnegie Mellon University, December, 1999.
19. D. Baraff, "Fast contact force computation for nonpenetrating rigid bodies," Proceedings of SIGGRAPH 94, July, 1994.
20. P. Spellucci, "An SQP method for general nonlinear programs using only equality constrained subproblems," Math. Prog. 92, 1998, pp. 413-448.
21. G. T. McKee and B. Brooks, "Resource management for networked robotics systems, Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS '97), Vol. 3, pp. 1363-1368, 1997.