Motion Control for a Novel Legged Robot

P.V. Nagy W.L. Whittaker

Field Robotics Center, Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

ABSTRACT

Legged robots exhibit certain locomotion advantages over wheeled and tracked vehicles: i.e. selective terrain contacts, enhanced agility, and platform stability. On the other hand, walking robots are more complex, and pose greater control challenges than do other classes of robotic locomotors. A motion control scheme has been developed to control all axes of a novel six-legged robot for planetary exploration. The robot's configuration and control scheme lends itself well to ensuring predictable and reliable walking. A single leg has been constructed, and the portion of the overall motion control scheme for it has been implemented.

INTRODUCTION

The control hierarchy for an intelligent walking machine has a number of levels that parallel the control hierarchy of industrial robots. At a high level, navigation planning occurs. At this level, routes are chosen to reach a goal, and obstacles are detected and mapped. Below this is the gait control level. At this level, the gait of the vehicle and footfall locations are determined. For example, if a hexapod is in very rough terrain, the appropriate gait might be to recover one leg at a time, as in the crawl gait. For flat, uneventful terrains, a wave gait facilitates higher speeds. Having planned routes, gaits, and footfall selections, the next level of the hierarchy executes control of the motion. The basic motion regulation level performs the inverse kinematics and controls the leg actuators in a manner analogous to control of an industrial robot. However, the forces on the linkage, desired system output, and specific control algorithms are differerent for a walking robot's leg.

The goal of the control scheme for a walking robot is to achieve sustained stable motion by maintaining body attitude, controlling velocity, avoiding vertical jolts, and surmounting or avoiding obstacles. Furthermore, the robot must be able to do all of this on rough ground, exhibiting terrain adaptability.

Walking robots differ from industrial robots in that they usually form multiple closed-loop kinematic chains that have poorly-defined and variable boundary conditions. Robots that hop, run, or bounce [1] don't face this problem; rather, they have difficulty achieving and maintaining dynamic stability. These robots require algorithmic-type control, such as Zero Moment Point control for a biped [2]. Statically stable walking machines have four or more legs, of which at least three are in contact with the ground at all times. If more than three legs support the weight of the robot, some scheme for the robot to actively interact with the ground is needed. For a robot that ventures outdoors, the ground compliance and resulting foot/terrain interaction cannot be known *a priori*. Furthermore, the existence of multiple closed loop chains may give rise to vertical control instabilities, due to force interactions between legs [3].

The physical configuration of a legged robot may reduce its motion complexity and enable simplifying advantages to its control scheme. For example, some robot configurations decouple the motions of suspension from those for propulsion. This simplifies the control schemes by separating control of quantities that do not interact significantly. This approach was adopted in the design of our robot.

THE AMBLER

We have configured a walking robot to accomplish reliable locomotion with unparalleled terrainability in the conditions speculated for planetary surfaces. A walker configuration is challenged to achieve very efficient power expenditure, traversability, and stable platform motion; which are of prime importance for planetary exploration. The proposed walking vehicle, AMBLER, is depicted in Figure 1. the original concept [4] was a hexapod with a single stack from which all legs emanated. The final design utilizes a twin leg stack configuration.

A feature that simplifies control and makes low-energy walking possible is the decoupling of the horizontal and vertical actuations. Each leg resembles a cylindrical type of robot arm. There are two prismatic joints (elbow and foot extensions) and one rotary (shoulder) joint. Body attitude and altitude control are accomplished by the six vertical actuators. The body is pulled forward by the horizontal actuators. The horizontal actuators have no gravity load on them, if the body is kept level.

The ability to move the rover forward with only three powered joints lends itself well to having low power requirements. The remaining joints can freewheel or be backdriven, precluding actuator conflicts.

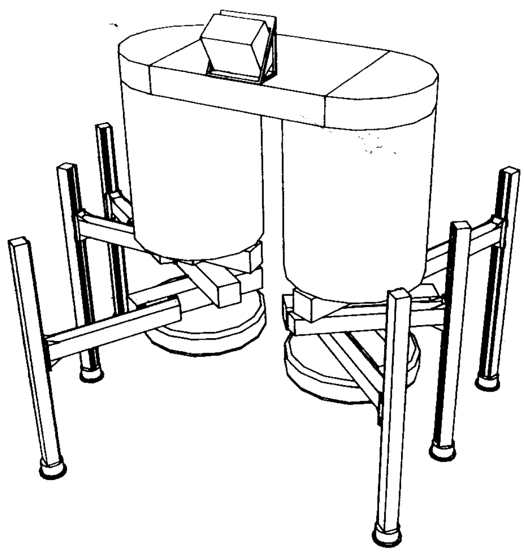

Figure 1. Configuration of AMBLER

THE WALKING MOTION MANAGER

The walking_motion_manager (WMM) is a body of code which executes motion control [5]. It accepts motion sequences generated by the walk_planner in the form of a series of Cartesian knot-points for leg and body positions. These motion sequences are represented as knot-points to be interpolated and executed by the WMM. In addition, control of actuator power and modes of operation that correlate to desired motion sequences are prescribed through these commands.

The structure of the walking_motion_manager is shown in Figure 2. The WMM consists of seven sub-objects:

* Trajectory Command Queue
* Trajectory Generator
* Propulsion Control
* Attitude/Altitude Control
* Actuators
* Sensing
* Contingency Executor

All sensing critical to motion control is contained within the WMM. This ensures fast response to ensure the survivability of the rover. Consequently, motor control signal processing, sensing, and reacting to body tilt occurs within the WMM.

Trajectory Command Queue

This module of the walking_motion_manager queues command strings will be of a standard form:

Body move, Leg position for legs 1 to 6, Actuation mode for legs 1 to 6, Commanded speed

In addition, elemental commands for body and individual leg moves are supported.

Body movement is relative to a commanded goal position (knot-point). This consists of increments along all three linear dimensions and a change in the orientation of the body about its vertical axis. The Cartesian leg positions are specified relative to the body position for a newly commanded knot-point.

Information in addition to a knotpoint is necessary to execute motion. The mode of operation for both horizontal and vertical actuators is commanded. For the horizontal actuators, there are linear and joint interpolated motions. The vertical motion allows position control, force control, and transition control. Transition control is the control scheme used to establish a footfall - changing from position to force control, thereby achieving a desired foothold. Other commands are used to enable or disable amplifiers and brakes.

The speed command is a percentage rating of maximum joint velocity. This will be dictated by higher levels of the hierarchy and also by the walk_planner, based on the type of terrain the robot is traversing.

Trajectory Generator

The trajectory generator generates fine-grained joint-space trajectories. It forms these from the locomotion goals which are stored in the trajectory command queue. The locomotion goals are interpolated to give servo setpoints. If the leg is operating in linear mode, the knot-points are linearly interpolated, as shown in Figure 3, for the body and one recovering leg. In this diagram, the leg is raised, weaves around obstacles, and is then placed.

Motion planning is carried out asynchronously, while motion execution is a synchronous event. Therefore, the planned trajectories must be assigned a time base. The commanded speed usually specifies the speed of the body. The recovering leg position(s) are given simultaneously with commanded body locations. Consequently, the recovering leg trajectories are coupled to the body trajectory. In this manner, the time base for the leg trajectories are defined implicitly. In the cases where the body does not move, the speed command is interpreted as a commanded leg speed. The setpoints are pre-computed, queued, and provided to the interpolators in real-time.

The setpoints are given in joint space for the horizontal actuators and for the vertical actuators of recovering legs. For linear interpolation, the inverse kinematic model is used for this purpose. The remaining setpoint is the commanded height of the body.

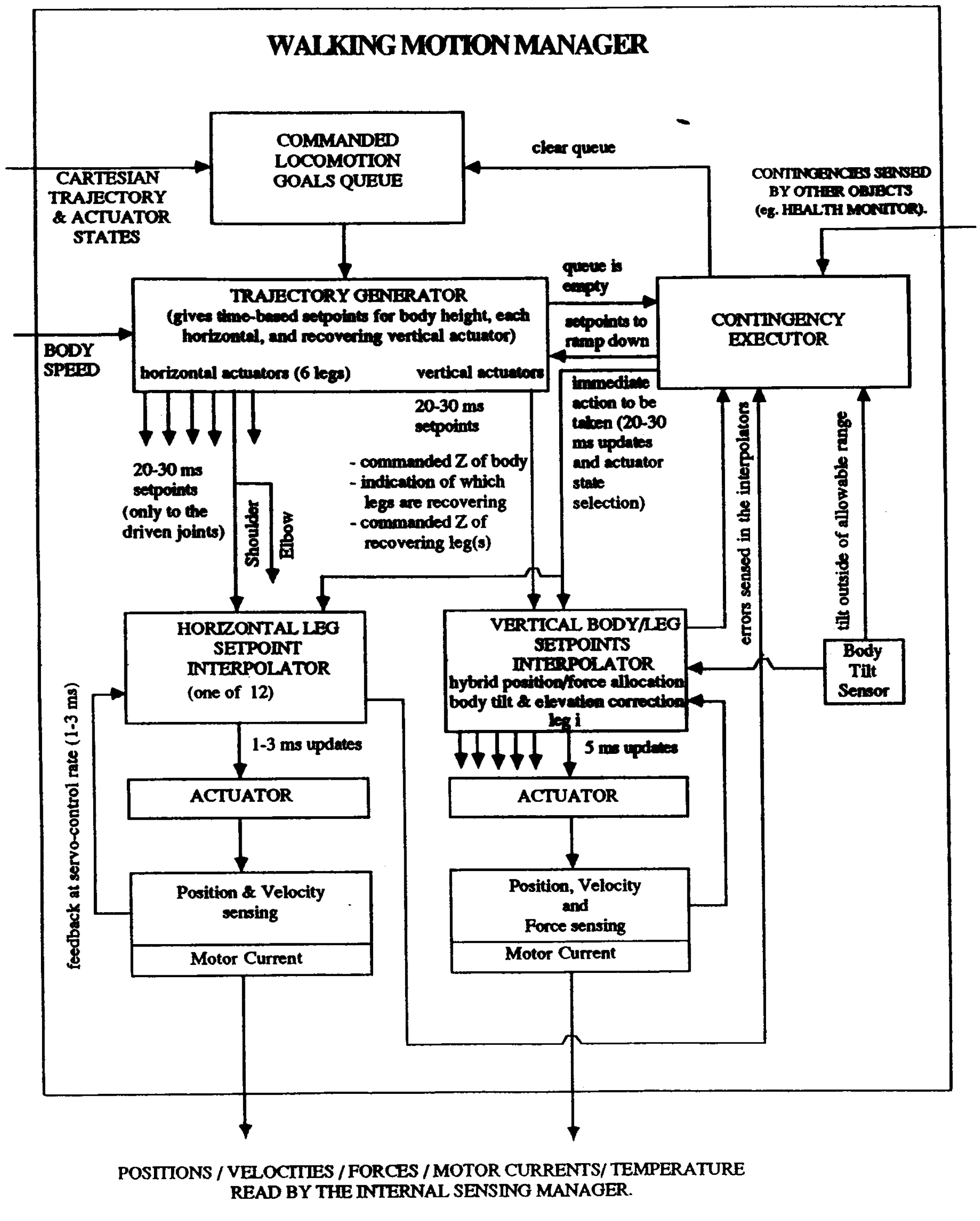

Figure 2. Structure of the Walking_Motion_Manager

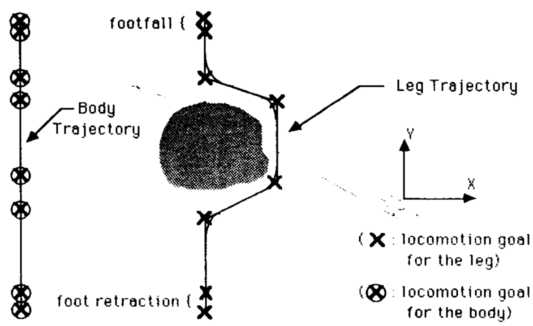

Figure 3. Linear Interpolation During a Leg Recovery

Propulsion Control

The horizontal setpoint interpolators control the propulsion actuations of the AMBLER. They are analagous to the form of an industrial robot controller [6]-[8]. Each joint's servoloop acts independently of one another, with the exception of being coordinated by a host processor at relatively slower update rates.

With the separate servosystem approach there is no global Cartesian-based feedback; each joint servos to its commanded goal, ignoring the dynamic effects of the other joints and linkages. The speed requirement for a planetary rover is of the order of 1 km per day [9]. Thus the robot moves slowly, lessening effects such as coriolis, centrifugal, and inertial coupling. Furthermore, to be able to develop the torques required to move such a massive machine on low power, the actuators are heavily geared. The shoulder and elbow motors have gear ratios of 240:1. As a consequence, motor inertia and friction effects dominate, and each axis may be effectively controlled as separate entities. Advantages of the separate servosystem approach include simplicity and modularity in design.

Attitude / Altitude Control

The vertical setpoints interpolator controls the attitude and altitude of the AMBLER. There is only one setpoint interpolator for the vertical actuators, as the vertical actuations require coordinated real-time control in reaction to sensor data. To ensure energy-efficient walking and stability, the body should be kept level at all times. This is accomplished by attitude sensors, supporting legs, and the interpolator algorithm. One servoloop that controls the supporting legs is always active when the leg brakes are removed. In addition, up to three legs may be controlled in a leg recovery mode.

The recovering leg(s) are controlled positionally, in the same manner as the horizontal actuators. Their servo update time will be slightly longer than for the horizontal actuators. While the processor for the vertical interpolator will be more powerful than the horizontal interpolators, it has a larger computational burden. However, the setpoints from the trajectory generator will be received at the same rate as with the horizontal interpolators.

The usual gait of our robot will consist of picking up and recovering one leg at a time. This leads to lower power consumption and extreme conservatism with regards to robot safety, which is a primary concern for a planetary rover. Thus the robot, AMBLER, will usually be supported by five or six legs. Only three of these legs are necessary to control the tilt and elevation of the rover's body. Thus, the body support system is over-constrained. There are several options for controlling the supporting legs. Three options are under current consideration are pure positional control, pure compliant (force) control, and a hybrid scheme that combines these two methods.

Pure positional control has the advantage of simplicity. In this method, the rover could bring a recovering foot into ground contact (at some force threshold) and hold that leg extension. It could do this by continuously applied position control or by applying the brakes to that vertical actuator and cease servoing. The latter approach would conserve energy. One disadvantage of either method is that one may not bias the weight supported by each leg in some (quasi)optimal manner. One may wish to allocate the forces carried by different legs to maximize stability. It is also possible that as the rover moves along, only three legs might be supporting the robot due to the other feet slipping during the body motion. If the projection of the rover's center of gravity is near the edge of the remaining ground support polygon, it could lead to an unacceptable tilt of the rover. Another problem occurs when driving the rover forward with only three actuators, e.g. forming a four-bar linkage through two legs. If one of these legs slips, the trajectory of the body will be affected.

Alternatively, one could compliantly control all supporting legs. For example, one choice of a reference vector of leg forces is to distribute the weight of the machine equally among the supporting legs. A better choice would be to distribute the leg forces based on the rover's current geometry for optimal stability. One way of controlling each actuator under such a scheme is hybrid position/force control. The commanded leg extensions and the gains of both position and force servoloops could be adjusted according to estimates of soil compliance and distance of the feet from the body. However, force control of all supporting legs may lead to control instabilities due to modeling and sensing inaccuracies and time delays in the control system. When this approach was tried on the OSU hexapod, the actuator systems became unstable [3]. In that work, it was noted that the instabilities become a problem when the discrete control frequency (servoloop update) is insufficient or the system is excessively stiff. Compliantly servoing all legs still holds promise for our robot, as we have adopted reasonably fast update times, and our robot will have large intrinsic compliance.

In reference [3], the authors report that controlling three legs purely by position and the remainder in compliant control, led to greater control stability. This method is well suited for explicitly controlling the tilt and elevation of the rover by using the three positionally-controlled legs. Additionally, in our implemetation, we may bias the force on these legs, so that they carry larger loads and will then be less likely to slip. If the rover is being pulled forward by use of the minimal number of horizontal actuators, they could be chosen such that they pull through the positionally-controlled legs as well. This will then help increase the accuracy of executing desired body trajectories, as these legs are less likely to slip. A disadvantage of this method is that we rely heavily on three legs for both body positioning and tilt control. Should one of these slip into a hole or suddenly fail, the robot could tipover. This may be countered by choosing the positionally controlled legs based on rover geometry and the constraints imposed by the local terrain. For example, if a foot rests at the edge of a rock, it is unsuitable for leveling control. Contingency actions could also instantly switch all legs into positional control (or lock them in their current position) in case of such an emergency.

Ongoing work in leveling control includes a vertical control testbed and simulation studies. The control testbed consists of six vertically-oriented actuators. These are connected by a reconfigurable frame, allowing changes in the geometry of the mechanism. Simulation studies of various leveling control algorithms are being carried out by utilizing the full dynamic model of the AMBLER [10].

Contingency Executor

The contingency executor is the code which represents the last line of defence for survivability of the rover. When an event arises that is immediately hazardous to the survival of the rover, the contingency executor will react. This module safely brings the rover to a complete stop, flushes the trajectory setpoint queue, and informs higher levels of the architecture that a failure has occured. At this point, diagnostics, replanning, and/or repair are done by other parts of the rover's system. The control actions to bring the rover to a halt are conveniently decoupled, as in regular locomotion.

Control of the horizontal actuators in case of contingency is straightforward. In reacting to contingencies, the rover should try to maintain current horizontal link positions and stop forward motion quickly and smoothly. One good way of doing this is to use all of the horizontal actuators. While this will lead to some actuator conflict, the time duration of this is not very long. Using all actuators will cause a peak energy load; however, this may be handled by storage batteries and by shutting down components not critical to blind locomotion (e.g. cameras, scanners) if necessary.

In normal operation, it is envisioned that a minimal number of horizontal actuators will pull the rover forward. If one or more of these fail, then the robot has less than the required number of actuators to effect a controlled deceleration. Using all operational joints during a contingency will ensure that this problem will not occur. Powering all horizontal actuators also helps ensure that the rover stays within its conservative support polygon [11] which guarantees rover stability if any two of the supporting legs fail.

The most appropriate method for controlling the vertical actuators in case of a contingency would be to control them as one normally would, to help ensure that the attitude of the rover is maintained. However, an alternate scheme is required in the case of a failed vertical actuator or if excessive tilt is sensed.

For the failure of a vertical actuator, the power to its amplifier will be disabled, and its brake applied. The rest of the actuators control the rover's tilt and height until the rover is safely brought to a halt.

With excessive tilt, all six legs could be used for attitude control. Having all of the horizontal actuators powered during this event will help stiffen up the rover and prevent further foot slippages than those which may have occured to initiate the large tilt.

The contingency executor will take action on a number of triggering events. Aside from excessive body tilt, it will trigger if there is a hardware fault or a large servo error. If the active joint servos are not tracking their trajectories closely, there could be a hardware problem, or the amplifiers are saturated, signifying that the task currently being executed is too difficult to be accomplished with only the chosen actuators. Any sensed collision (other than during footfall) will be the result of an intolerable error in the planned path, path tracking error and/or the local terrain model. The rover should be brought to a halt and a path replanned after updating local terrain maps. The contingency executor may also be triggered by other modules AMBLER's software system. For example, in a higher-level module, foot slippage could be estimated from position and force data. If it is excessive, the rover could temporarily employ more horizontal actuators for propulsion while situated on the type of terrain currently being traversed.

THE SINGLE LEG TESTBED

A full-scale single leg has been built to check the leg's mechanical integrity and resolve physical control issues [12]. We also use it to implement a pared down version of our integrated computer architecture, which consists of planning, perception, and control. The testbed is shown in Figure 4. The leg used in this testbed conforms to the original single stack design [12], however, the experimental program, and the software developed is directly applicable to the new leg design. The leg is mounted on a trolly, which is used to simulate body motion. The trolly is carrying an ERIM laser rangefinder for terrain data acquisition.

The walking_motion_manager for the single leg testbed has been implemented. The code is written in "C" and is downloaded from a SUN workstation to a 68020 running under vxWorks. VxWorks is a real-time, multitasking operating system. Motion control is effected by Creonics motion control cards that are residing on the same VME bus as the 68020. These cards provide programmable P,I,D and feed-forward gains and a trajectory tracking command to follow joint trajectories, as specified by the host controller. However, we are concurrently developing our own servoloop hardware, should we find it necessary or desirable to have complete flexibility.

With the Creonics motion control cards controlled by the walking_motion_manager, we have achieved coordinated motion and vertical-axis force control. An interface has also been built which accepts commands from the gait control level.

CONCLUSIONS

Walking robots are complex to design and control. However, with careful thought, this complexity may be substantially reduced. Computational requirements for control purposes once

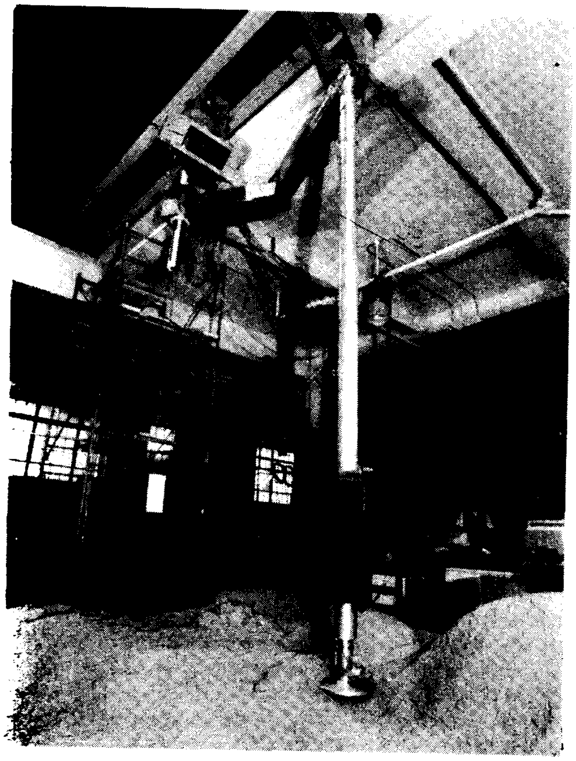

Figure 4. The Single Leg Testbed

posed a problem; however, this requirement is now tractable. In fact, the requirements related to perception (e.g. using vision or laser range finding) pose a much greater computational burden than computing for locomoting for any autonomous vehicle operating in an unstructured environment.

A first-level implementation of a basic motion regulation scheme for an energy-efficient gravity-decoupled walking robot has been implemented for a prototype leg. The proposed robot and corresponding motion management scheme should be in place during the fall of 1989.

REFERENCES

[1] Raibert M.H., Chepponis M. and Brown H.B., "Running on Four Legs as Though They Were One", IEEE Journal of Robotics and Automation, vol. RA-2, no. 2, June 1986, pp. 70-82.

[2] Todd D.J., Walking Machines: An Introduction to Legged Robots, Kogan Page, London, 1985.

[3] Klein C.A. and Chung T.S., "Force Interaction and Allocation for the Legs of a Walking Vehicle", IEEE Journal of Robotics and Automation, vol. RA-3, no. 6, Dec 1987, pp. 546-555.

[4] Bares J., et. al., "AMBLER: An Autonomous Rover for Planetary Exploration", IEEE Computer Magazine, June 1989, pp. 18-26.

[5] Wettergreen D., et. al., "A Software Architecture for Planetary Exploration Vehicles", internally circulated document, Robotics Institute, Carnegie Mellon University, April 15, 1988.

[6] Nagy P.V., "Tracking Control of an Industrial Robot", M.Eng.Thesis, Department of Mechanical and Production Engineering, National University of Singapore, June 1987.

[7] Luh J. Y. S., "Conventional Controller Design for Industrial Robots - A Tutorial", IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 3, May/June 1983, pp. 298-316.

[8] Valavanis K., "Unimate PUMA Manipulator Manual", Rensselaer Polytechnic Institute Technical Report RAL-3, March 1982.

[9] Spiessbach A., et. al., "Issues and Options for a Mars Rover", Proceedings of SPIE, Mobile Robots II, vol. 852, Nov. 1987, pp. 164-171.

[10] Manko D.J., "Models of Legged Locomotion on Natural Terrain", PhD thesis proposal, submitted to the Department of Civil Engineering, Carnegie Mellon University, June 1988.

[11] Mahalingam S., "Terrain Adaptive Gaits for the Ambler", M.Sc. Thesis, Department of Mechanical Engineering, University of North Carolina at Charlotte, June 1988.

[12] Nagy P.V. and Whittaker W.L., "Experimental Program for the CMU Mars Rover Single Leg Testbed", in print, 20th annual Pittsburgh conference on Modeling and Simulation, May 4-5, 1989.