Mars Surveyor '98 Lander MVACS Robotic Arm Control System Design Concepts

Robert G. Bonitz
Telerobotics Research and Applications Group
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109-8099

Abstract

This paper describes the control system design concepts for the Mars Volatiles and Climate Surveyor (MVACS) Robotic Arm which supports the scientific investigations to be conducted as part of the Mars Surveyor '98 Lander project. Novel solutions are presented to some of the unique problems encountered in this demanding space application with its tight constraints on mass, power, volume, and computing resources. Problems addressed include the 4-DOF forward and inverse kinematics, trajectory planning to minimize potential impact damage, joint drive train protection, Lander tilt prevention, hardware fault monitoring, and collision avoidance.

Keywords

control, manipulator, Mars, robot, trajectory, 4-DOF

1 Introduction

In January of 1999 NASA will launch the Mars Surveyor '98 Lander spacecraft with the Mars Volatiles and Climate Surveyor (MVACS) integrated science payload as part of the Mars Exploration Program [1]. The target landing site is at 71° S latitude late during the Martian southern spring season. The goal of the MVACS mission is to conduct scientific investigations to:

1. Characterize the surface environment;
2. Search for subsurface ground ice;
3. Determine the quantity of CO_2 and H_2O in the soil;
4. Determine the abundance of volatile-bearing minerals in the soil;
5. Search for climate records in the form of fine-scale layering in near-surface materials.

The major elements of the MVACS[†] science payload consist of (see Figure 1):

1. A Surface Stereo Imager (SSI) supplied by the University of Arizona to characterize the general environment;
2. A Robotic Arm Camera (RAC) supplied by the University of Arizona and the Max-Planck-Institute for Aeronomy to provide close-range surface and subsurface images;
3. A Thermal and Evolved Gas Analyzer (TEGA) supplied by the University of Arizona to determine concentrations of ices, adsorbed volatiles, and volatile-bearing minerals in soil samples;
4. A Meteorology Package (MET) and Soil Temperature Probe (STP) supplied by the Jet Propulsion Laboratory (JPL) to measure pressure, wind speed, and temperature;
5. A two-meter 4-DOF Robotic Arm (RA) supplied by JPL to dig trenches, acquire soil samples, position the RAC, position the Soil Temperature Probe, position the soil sample for viewing by the SSI, and deposit the acquired soil sample in the TEGA for analysis.

The Lander spacecraft is being built by Lockheed Martin Astronautics. The Principal Investigator is Professor David Paige of UCLA with project management provided by JPL.

Since this is a space application, there are tight constraints on mass, power, volume, and computing resources. The Robotic Arm is allocated a mass of 3.5kg and a 10 watt average (20W peak) power budget. The

[†] http://thesun.ess.ucla.edu/

Robotic Arm control software will run on a RAD6000 running at 5MHz in a multitasking environment under the VxWorks[†] operating system. Additionally, the integrated instrument payload makes maximum use of subsystem designs that have been previously qualified on the Mars Pathfinder program to reduce cost and mission risk. The purpose of this paper is to present the design concepts of the Robotic Arm Control System delineating some of the unique problems encountered in this application.

2 Robotic Arm System Design

The Robotic Arm system consists of a lightweight 4-DOF manipulator, Arm Control Electronics (ACE) to drive the motors and read in sensor data, and the control software to provide command expansion, trajectory generation, control compensation, and to monitor the sensor data. Salient features of the manipulator are:

1. 4-DOF back hoe design sufficient to achieve digging, sample acquisition, and RAC and STP positioning;

2. Low-mass graphite-epoxy tubular links;

3. Joint actuators consisting of low-power (3.2W) motors with two-stage speed reduction (planetary gear plus harmonic drive) to achieve low-mass with high-torque output;

4. Capability of exerting up to 170N force at the end-effector for digging with up to 800N for ripping with the tines in selected configurations;

5. Non-backdriveable joints through the use of motor detents which can be selectively shut off to conserve energy during selected manipulator motions;

6. Actuator heaters to maintain adequate temperature in the cold Martian environment;

7. Joint encoders and potentiometers to achieve 1cm absolute (0.5cm relative) positioning accuracy.

8. End-effector tools to perform the various mission functions including the RAC, STP, scoop for digging, and tines for ripping;

9. An offset elbow joint so that the end effector can swing through and reach behind to facilitate showing an acquired sample to the SSI;

[†] Wind River Systems, Inc., 1010 Atlantic Ave., Alameda, CA

Table 1: MVACS Robotic Arm Task Commands

Name	Description
ra_acquire_sample (x,y,z,sample_type)	Acquire a sample at location x,y,z.
ra_dig_trench (x,y,z,trench_depth, trench_length, trench_width, trench_angle, dig_angle)	Dig a trench starting at x,y,z with the specified parameters.
ra_ssi_show()	Move the sample into position for viewing by SSI camera.
ra_tega_dump()	Dump the sample into the currently designated TEGA port

10. A payload capacity of 5Kg at full-arm extension.

The Arm Control Electronics consists of circuitry to read in data from the encoders and pots, provide current drive to the motors, and set motor voltage levels, and an 8032 microprocessor to execute low-level joint set-point commands received from, and to pass sensor readings to, the control software.

The control software receives high-level task commands from the ground via the Lander and expands them into Cartesian and joint-motion commands. Arm joint trajectories are computed for each motion command and each via point is sent to the ACE for execution. The control software also monitors the sensor data for collision avoidance and hardware fault detection. An example of a high-level task command is to dig a trench with a specified depth, length and starting location. The control software expands the dig trench command received from the ground into a sequence of Cartesian motion commands which are executed in turn. This provides ease of use of the Robotic Arm by the ground operator and conserves uplink bandwidth. To provide maximum flexibility, however, the ground operator may also send both Cartesian and joint-motion commands. A sample of some of the available task-level and low-level commands are given in Tables 1 and 2, respectively. The MVACS Robotic Arm control system is depicted in Figure 2.

3 Kinematics

Since the MVACS Robotic Arm has only four degrees of freedom, complete specification of the Cartesian position and orientation of the current tool frame

Table 2: MVACS Robotic Arm Move Commands

Name	Description
ra_move_cartesian $(x, y, z, \theta \mid d, \text{ref})$	Move the tool to the designated position. If tool=RAC, x, y, z is the target location and d= distance from the RAC to the target. ref= world absolute | world relative | tool
ra_move_joint (jt1,jt2,jt3,jt4,ref)	Move the joints the designated number of radians or seconds. ref = absolute | relative | time

is not possible. One must choose which four degrees of freedom out of the six degrees that completely specify the position and orientation of the tool frame. For ease of operation by the user, the 4-DOF representation was chosen as $X = [x\ y\ z\ \theta]^T$ where $[x\ y\ z]^T$ is the Cartesian position of the origin of the tool frame in the world-coordinate system and θ is the angle which the tool approach vector makes with the world x-y plane (parallel to the Lander deck). For example, if one desires to insert the STP into the ground perpendicular to it, one would set the tool equal to the STP and specify $X = [x\ y\ z\ \pi/2]^T$. Frame assignments for the manipulator are given in Figure 3. Each of the tool frame approach (z) vectors is orthogonal to the rotation axis of the wrist joint so that the desired orientation is achievable. One exception is the RAC which is attached to the forearm and, thus, has only three degrees of freedom. For the RAC, the user only specifies the desired position of the tool frame and not orientation.

The MVACS 4-DOF manipulator presents an unusual inverse kinematics problem. A typical method for solving the inverse kinematics for a 6-DOF manipulator is to compute the desired location of the wrist frame from the specified position and orientation of the tool frame and then use geometric methods to solve for the first three joints followed by the last three joints [2]. For a 4-DOF manipulator with only one orientation angle specified, the above method fails since one does not have the full rotation matrix of the tool frame and cannot compute the wrist frame location.

In the MVACS case, however, it is possible to solve directly for joint 1 from the specified tool-frame position using geometric techniques since the last three joint axes are parallel and orthogonal to the axis of joint 1. One can then solve for the wrist position from

$$^{w}p_4 = {}^{w}p_{tool} - {}^{w}R_4\,{}^{4}p_{tool} \tag{1}$$

where $^{w}p_4$ is the wrist location, $^{w}R_4$ is the rotation matrix of the wrist, $^{4}p_{tool}$ is the vector from the wrist-frame origin to the tool-frame origin expressed in the wrist frame, and $^{w}p_{tool}$ is the desired position of the tool in world-frame coordinates. From the forward kinematics, we get

$$^{w}R_4 = \begin{bmatrix} c_1 c_{234} & -s_1 & c_1 s_{234} \\ s_1 c_{234} & c_1 & s_1 s_{234} \\ -s_{234} & 0 & c_{234} \end{bmatrix} \tag{2}$$

where $s_i = \sin(\sum q_i)$, $c_i = \cos(\sum q_i)$, q_i is the ith joint angle, and $q_2 + q_3 + q_4 = {}^{w}\theta_{tool} - {}^{4}\theta_{tool} + \pi/2$. $^{w}\theta_{tool}$ is the specified tool-frame orientation angle and $^{4}\theta_{tool}$ is the angle from the wrist-frame z axis to the tool-frame z axis about the wrist-frame y axis. Now that the wrist position is know, one can use standard geometric techniques to solve for remaining joint angles.

The RAC presents a unique inverse kinematics problem as well since we cannot specify both its location and orientation. Furthermore, it would difficult for the operator to specify the location of the RAC given that the location of the target to be imaged is known. To make it easy on the operator when the tool is set to RAC, Cartesian motion commands are interpreted as $[x\ y\ z]^T$ being the location of the target to be imaged and the fourth element, d, as the distance from the RAC to the target. To solve the inverse kinematics, the following method is used:

1. Redefine the forearm to be the line from the elbow joint to the target when the RAC approach vector points at the target and compute its length from d and the arm geometry;
2. Set $^{w}p_4$ to be the target location;
3. Solve for the three joint angles using standard geometric techniques;
4. Compute an adjustment to q_3 from d and the arm geometry necessary because of step 1.

4 Trajectory Planning

The Robotic Arm has the capability to execute both Cartesian and joint motion commands. Cartesian motion can be expressed as absolute or relative motion in the world frame or as tool-frame motion. Furthermore, straight-line or joint-interpolated motion can be specified. Joint-space motion commands can be given as absolute joint angles, as relative joint angles with respect to the current joint positions, or as timed motion. In order to prevent the arm from moving too far and potentially damaging Lander hardware in the event of loss of communication between the ACE and the control software, a sequence of via points is computed for any motion command and only after the arm has nearly reached the current via point is the next via point sent to the ACE. Thus, trajectories are generated as a sequence of points in space only and not in time. The one exception is the timed joint-motion commands in which the operator can command each joint to move for a specified amount of time in which case the generated trajectory is a sequence of via points in time only and not in space.

The manipulator joints have high gear ratios and joint velocities are slow (on the order of tenths of a degree per second). During spatial arm motion at least one joint is moving at its maximum velocity. The remaining joint velocities are appropriately scaled to achieve coordinated motion. For timed joint motions, the operator can set the motor voltages as desired which will determine the joint velocities.

For Cartesian via sequence generation, the bounded-deviation joint-path method in [3] is used. This algorithm is basically a recursive bisector method for finding the via points. The straight-line Cartesian position at the midpoint is compared to the Cartesian position calculated at the midpoint of the corresponding joint space. If the error exceeds a specified amount (set to smaller than the required accuracy), the path is divided into two segments and the algorithm executed for each segment. The process continues until the error limit is satisfied.

Since we require via sequencing for both spatial and timed joint-motion commands as well, a similar approach is taken to generate the via sequence. The algorithm is as follows:

1. Store the current position as the first via point (for timed moves the starting via is set to zero);
2. Compute the joint-motion midpoint;
3. If the amount of the joint motion exceeds the specified limit, execute steps 2-4 for the two segments (start point, midpoint) and (midpoint, end point);
4. Else store the end point as the next via point.

The limit and the amount of joint motion is specified as a single metric and is represented by $||q||$ where q is the joint vector in radians or seconds.

5 Control System

In addition to task expansion and trajectory generation, the Robotic Arm control software also monitors progress of the arm towards its commanded via point as well as joint torque and lander tilt-moment limits. Compensation is a basic proportional control [4] scheme with sufficient gain such that at least one joint motor voltage is saturated so that maximum velocity is achieved until the arm approaches the set point at which time the arm velocity decreases. The remaining motor voltages are scaled accordingly using the scale factor computed from the saturated joint motor (scale factor = max motor voltage/computed voltage).

Due to the high gear ratios, it is possible for the motors to exert enough torque to exceed the specified torque limits of the harmonic drives. During free-space arm motion, the load on the joints will not exceed the torque limits. However, during digging it is possible to exceed the limits if the motion of the end effector is impeded by rock, ice, or hard soil. Furthermore, under the preceding condition in certain arm configurations, it is also possible for the arm to exert enough force and torque on the lander to tilt it; viz., high enough to lift one or more footpads off the ground. It is critical that joint torques be monitored to assure that damage or tilting not occur.

The MVACS Robotic Arm is not equipped with a force-torque sensor, but the motor currents are sensed so it is possible to estimate joint torques whenever the motors are on using the motor torque constant which is nearly linear. However, individual motors are turned off to conserve energy whenever they are not needed to execute the specified motion command. E.g., during digging the azimuth motor (joint 1) is normally off. It is possible, however, to compute the full set of joint torques from the motor currents of the actuated joints.

Define:

n		the number of actuated joints
J	=	4-DOF manipulator Jacobian $\in \mathbb{R}^{4\times 4}$
$\hat{J}$	=	the columns of J associated with the actuated joints $\in \mathbb{R}^{4\times n}$
τ		full set joint torques $\in \mathbb{R}^4$
$\hat{\tau}$	=	actuated joint torques $\in \mathbb{R}^n$
f	=	exerted endpoint force $\in \mathbb{R}^4$

We can get the joint torques for the actuated joints from the sensed motor currents. Since the arm has low mass, high gear ratios, and moves slowly, we can safely ignore the acceleration, Coriolis, centripetal, and gravity dynamics as they represent only a small portion of the joint torques when the limits are being approached. The relationship between the actuated joint torques and the force at the endpoint is [5]

$$\hat{\tau} = \hat{J}^T f. \quad (3)$$

The solution to (3) is [6]

$$f = \hat{J}^{T\#} \hat{\tau} + (I_4 - \hat{J}^{T\#} \hat{J}^T)\epsilon \quad (4)$$

where $\tilde{J}^{T\#}$ is a right pseudo-inverse of $\tilde{J}^T$, I_4 is the 4x4 identity matrix, and ϵ is an element in the null space of $\tilde{J}^T$. Note that the arm cannot exert forces in the null space of $\tilde{J}^T$ ($\epsilon = 0$) and (4) reduces to

$$f = \tilde{J}^{T\#}\tilde{\tau}. \qquad 5)$$

Using (5), we get the full set of joint torques from the actuated joint torques:

$$\tau = J^T f = J^T \tilde{J}^{T\#}\tilde{\tau}. \qquad (6)$$

The algorithm does require a pseudo-inverse which can be computed using a singular value decomposition or, alternately, closed-form solutions to the pseudo-inverse can be determined. There are 14 solutions corresponding to the combinations of actuated joints - 4 with one joint actuated, 6 with 2 joints actuated, and 4 with 3 joints actuated. Each solution requires the inversion of a $n \times n$ matrix. There is one caveat - singularities must be avoided. Otherwise, $\tilde{J}^T$ would lose full-row rank and the pseudo-inverses could not be computed.

Once the full set of joint torques is computed, the reaction force and torque at the Lander deck can be computed and used along with the Lander geometry to compute the tilt moments about the lines between the footpads. The joint torques can be compared to the harmonic drive torque limits and the tilt moments can be compared to the tilt-moment limits to assure that damage or tilting does not occur. If either of the limits are being approached, the planned trajectory is modified by computing a delta to the current via point that moves the arm in a direction orthogonal to and in the plane of the current direction of motion. This has the effect of shallowing the digging trajectory. If the limits are exceeded (with allowance for a safety margin) the arm is stopped and the situation assessed by imaging the scene with the SSI and evaluating the downlinked images along with the engineering data.

6 Fault Monitoring and Collision Avoidance

It is critical that the Robotic Arm operate safely during the execution of its assigned tasks so as not to damage itself or any of the Lander hardware. Each time through the control loop, sensor data is analyzed and an assessment made as to whether any hardware failures have occurred. Available sensor data include joint positions from both encoders pot voltages, motor currents, joint temperatures, power supply status, and A/D reference voltages. Potential hardware faults include failures of the sensors, motors, power supply, or voltage reference. The position of the joints as determined from both the encoders and pot voltages is also assessed and if the difference exceeds a specified limit, the arm is recalibrated. The calibration process consists of driving the arm against the joint limits and reloading the encoder counters. During normal operation the encoders are used as the primary joint-position sensor. In the event of an encoder failure, the arm can be commanded to use the pot voltage as the primary position sensor with some degradation of positioning accuracy.

To prevent collisions of the arm with the Lander hardware, all command sequences will be verified on the ground by simulation to assure that kinematic stay out zones are not violated. The Telegrip[§] robotic system simulation tool is used to simulate the arm motion. In addition, on board collision detection algorithms are executed both during path planning and while the arm is in motion. The MVACS Robotic Arm uses the obstacle detection scheme developed for the NASA Ranger Telerobotic Flight Experiment [7] adapted for the MVACS environment. This scheme is model based and uses simple object models and distance computation algorithms making it suitable for the MVACS real-time application. In the event of detection of an imminent collision, the arm is stopped and the situation assessed by imaging the scene with the SSI and evaluating the downlinked images.

7 Conclusion

Design concepts of the Mars Surveyor '98 Lander MVACS Robotic Arm Control System were presented describing some of the unique problems encountered in this demanding space application with its tight constraints on mass, power, volume, and computing resources. Novel solutions to the inverse kinematics problem, trajectory generation, damage and tilt prevention, hardware fault monitoring, and collision avoidance were described.

References

[1 D.A. Paige, et al, "Mars Volatiles and Climate Surveyor (MVACS), Integrated Payload Proposal for the Mars Surveyor Program '98 Lander," 1995. NASA AO No. 95-OSS-3.

[2] K. Fu, R. Gonzales, and C. Lee, *Robotics : Control, Sensing, Vision, and Intelligence.* CAD/CAM, Robotics, and Computer Vision, McGraw-Hill, 1987.

[§] Deneb Robotics, 3285 Lapeer Road West, Auburn Hills, MI

[3] R. Taylor, "Planning and execution of straight line manipulator trajectories," *IBM Journal of Research and Development*, vol. 23, no. 4, pp. 424-436, 1979.

[4] R.C. Dorf, *Modern Control Systems.* Addison-Wesley, 6th ed., 1992.

[5] M. Spong and M. Vidyasagar, *Robot Dynamics and Control.* New York, NY: John Wiley and Sons, Inc., 1989.

[6] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization.* Addison-Wesley, 1991.

[7] B. Bon and H. Seraji, "Obstacle detection for the Ranger Telerobotic Flight Experiment," in *6th International Symposium on Robotics and Manufacturing, 2nd World Automation Congress,* (Montpellier, France), May 1996.

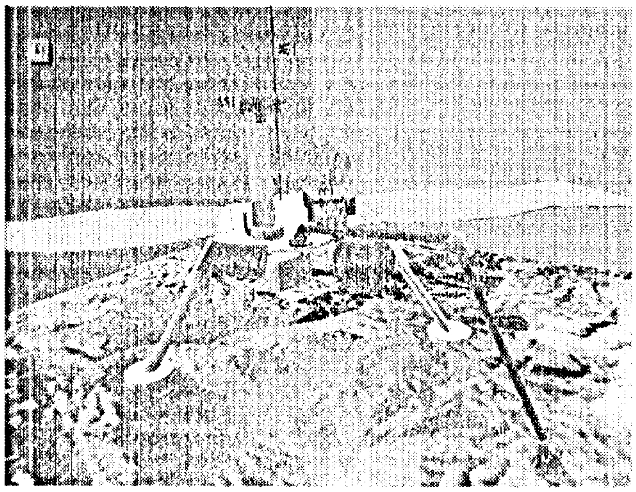

Figure 1: Mars Surveyor '98 Lander with MVACS

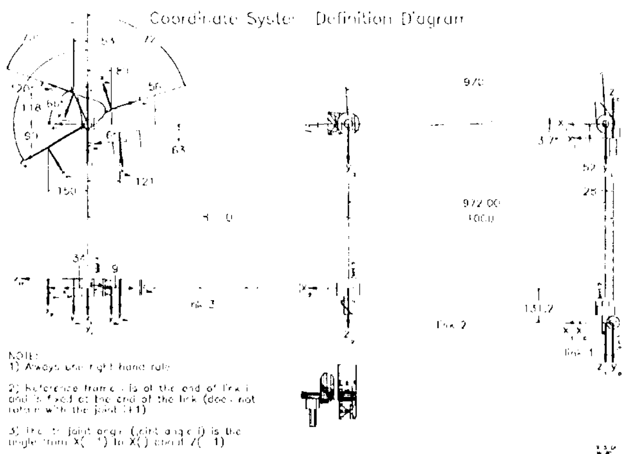

Figure 3: Manipulator Frame Assignments

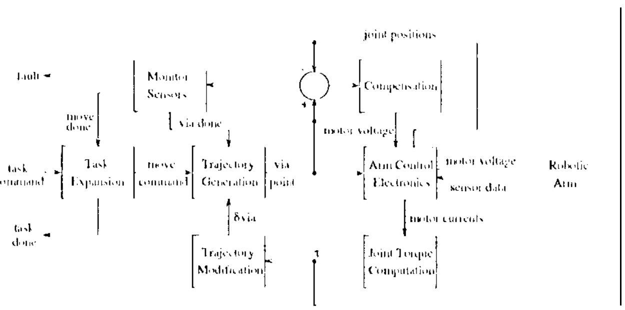

Figure 2: MVACS Robotic Arm Controller