NASA Conference Publication 10131

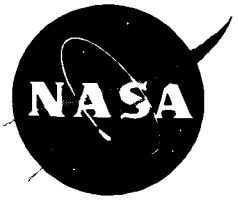
Selected Topics in Robotics for Space Exploration

Edited by
Raymond C. Montgomery
NASA Langley Research Center
Hampton, Virginia

Howard Kaufman
Rensselaer Polytechnic Institute
Troy, New York

Proceedings of a workshop sponsored by the NASA Langley Research Center and the Center for Intelligent Robotic Systems for Space Exploration and held at Langley Research Center Hampton, Virginia March 17–18, 1993

DECEMBER 1993



National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

Center for Intelligent Robotic Systems for Space Exploration
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Table of Contents

Comments by the Editors

Raymond C. Montgomery
NASA Langley Research Center
Hampton, Virginia

and

Howard Kaufman
Rensselaer Polytechnic Institute
Troy, New York

The papers and abstracts contained in this report represent both formal presentations and experimental demonstrations at the Workshop on Selected Topics in Robotics for Space Exploration which took place at NASA Langley Research Center, 17-18 March 1993. This workshop grew from discussions between Dr. Ray Montgomery from NASA Langley Research Center (LaRC) and Dr. Howard Kaufman from Rensselaer Polytechnic Institute (RPI). Because both the Guidance, Navigation, and Control Technical Committee (GNCTC) of the LaRC and the Center for Intelligent Robotic Systems for Space Exploration (CIRSSE) at RPI, shared common research directions in robotics for space exploration, it was evident that a forum for technical exchange would be very valuable.

Thus, with approval from both CIRSSE and the GNCTC, Drs. Montgomery and Kaufman solicited papers and/or demonstrations from LaRC, CIRSSE, and from persons from industry, government, and other universities with close ties to either LaRC or to CIRSSE.

The presentations were very broad in scope with attention given to space assembly, space exploration, flexible structure control, and telerobotics.

The organizers would like to thank all those who contributed to the success of the workshop. Special thanks are due Betty Lawson of CIRSSE, who retyped many of the abstracts and who helped with the initial solicitation of papers and Mr. Jack Pennington and Dr. Robert Williams who organized and participated in the LaRC Automation Technology Research Branch tours.

Introduction to CIRSSE

Alan A. Desrochers, CIRSSE Director
Rensselaer Polytechnic Institute
Troy, New York

Robotic systems combine mechanical devices such as robot manipulators and hands; sensors, such as computer vision, optical, tactile, and force with computer hardware and software to form an integrated system capable of reliably executing a variety of different tasks. Much of the research in robotics has focused on the theory, algorithms, and development of these individual component technologies. In the CIRSSE research program, we emphasize the integration of these technologies through the definition of architectural principles and the development of an integrated testbed for experimental studies and demonstration tasks. We have demonstrated assembly of space-truss structures, and have worked closely with NASA laboratories in the definition of space mission scenarios and applications.

The CIRSSE integrated testbed includes two 9-degree-of-freedom robotic manipulators mounted on a 12 foot long track. The resulting useful work volume of the 18-degree-of-freedom robotic system is more than 400 cubic feet. The effective control of cooperating manipulators for coordinated motion and force control tasks is an important research area that utilizes both simulation and experimental studies to verify new concepts.

The CIRSSE testbed incorporates several different types of sensors. There are currently five cameras integrated into the system. These TV cameras are supported by a special purpose high-speed computer for image processing and automated interpretation of images from a single camera or from two stereo pairs of cameras: one fixed and one mobile. A laser scanning range sensor is mounted from the ceiling of the testbed in order to provide 3-D depth information from any point in the robot work space. Tactile and force sensors are mounted on each of the robot hands, and they provide sensory feedback for the dexterous manipulation of objects and devices. The interpretation of sensory information acquired from multiple sensors is an important research topic in integration.

The CIRSSE hardware computing environment is based on a distributed multiprocessor system. Two VME-bus systems support the distributed multiprocessor environment and partition the two most critical real time functions: motion control and vision processing. The host and development environment is based on SUN workstations.

The CIRSSE Testbed Operating System (CTOS) is built on the commercial VXworks software, and provides a versatile multiprocessor real-time operating system capability. CTOS supports a versatile message passing protocol that adapts to different hardware configurations and available communication speeds. The communication protocols for intraboard, intrabus, and network communications among processors provides for an efficient implementation and debugging of systems. The message passing protocol that underlies all the CIRSSE testbed implementations supports an event driven, object-oriented approach to the overall architecture.

The CIRSSE Testbed integrated architecture is structured hierarchically in intelligence and integrates planning, coordination, and execution functions. Within this broad hierarchical framework the event driven, distributed nature of the system is maintained. Based on the task decomposition, the subplanning modules are available both on-line and off-line. The task decomposition itself is represented as a Petri net, and this Petri net structure is used as an embedded model for the coordination of individual execution modules. The Petri net itself may be thought of as a distributed event driven system, and the coordination level is implemented using a message passing protocol which parallels the lower level execution. The architecture at all levels remains modular and therefore, provides flexibility, reliability, and ease of implementation.

The CIRSSE multi-manipulator, multi-sensor integrated testbed, the CTOS operating system, and the CIRSSE integrated testbed architecture are unique developments of the CIRSSE Center. They provide a broad resource for the experimental study of robotic concepts and the demonstration of application tasks meeting the special needs of space exploration.

PART I
Technical Papers

A Global Approach to Kinematic Path Planning to Robots with Holonomic and Nonholonomic Constraints

Adam Divelbiss, Sanjeev Seereeram, John T. Wen
Rensselaer Polytechnic Institute
Troy, New York

A Global Approach to Kinematic Path Planning to Robots with Holonomic and Nonholonomic Constraints

Adam Divelbiss Sanjeev Seereeram John T. Wen
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
divelbis@cat.rpi.edu seereeram@ral.rpi.edu wen@ral.rpi.edu

Abstract

Robots in applications may be subject to holonomic or nonholonomic constraints. Examples of holonomic constraints include a manipulator constrained through the contact with the environment, e.g., inserting a part, turning a crank, etc., and multiple manipulators constrained through a common payload. Examples of nonholonomic constraints include no-slip constraints on mobile robot wheels, local normal rotation constraints for soft finger and rolling contacts in grasping, and conservation of angular momentum of in-orbit space robots. The above examples all involve equality constraints; in applications, there are usually additional inequality constraints such as robot joint limits, self collision and environment collision avoidance constraints, steering angle constraints in mobile robots, etc.

This paper addresses the problem of finding a kinematically feasible path that satisfies a given set of holonomic and nonholonomic constraints, of both equality and inequality types. The path planning problem is first posed as a finite time nonlinear control problem. This problem is subsequently transformed to a static root finding problem in an augmented space which can then be iteratively solved. The algorithm has shown promising results in planning feasible paths for redundant arms satisfying Cartesian path following and goal endpoint specifications, and mobile vehicles with multiple trailers. In contrast to local approaches, this algorithm is less prone to problems such as singularities and local minima.

1 Introduction

Controlling robot motion, including both manipulators and mobile vehicles, usually involves the following steps:

1. Kinematic path planning: find a path that satisfies all the geometric specifications and constraints of a given task.
2. Trajectory generation: index the path with time to generate a dynamic trajectory.
3. Dynamic trajectory following: design a servo controller (possibly incorporating dynamic information of the overall system) to follow the trajectory.

This paper focuses on the kinematic path planning problem. In contrast to most of the existing algorithms which are local and reactive in nature, our approach is a global one which warps an entire path to satisfy all the constraints. A common classification of constraints involves holonomic versus nonholonomic. If a constraint can be expressed in terms of the generalized coordinate, it is holonomic; if a constraint involves the generalized velocity and it is not integrable, then the constraint is nonholonomic. Examples of holonomic constraints include a manipulator constrained through its contact with the environment, e.g., inserting a part, turning a crank, etc., and multiple manipulators constrained through a common payload. Examples of nonholonomic constraints include no-slip constraints on mobile vehicle wheels, local normal rotation constraints for soft finger contacts in grasping, and conservation of angular momentum of space robots. There may be other design constraints imposed by the task, for example, equality constraints such as the desired terminal configuration, specified end effector path, etc., and inequality constraints such as manipulator joint limits, self collision and environment collision avoidance constraints.

There is abundant literature on path planning for redundant robots, which are examples of systems with holonomic constraints, and mobile robots which are examples of systems with nonholonomic constraints, but seldom on both. The principal reason is that in the local approach, the two problems are fundamentally different in that redundant robots can in general move in all directions locally in the configuration space, but nonholonomic systems can only move in certain directions. Consequently, the issues related to redundant robots are singularity, redundancy resolution, joint cyclicity for cyclic end effector paths, etc., and, for nonholonomic systems, the main issue is in finding a path whose tangent lies within the admissible directions. There are also commonalities in the two classes of problems:

1. Kinematic models are linear in control (i.e., admissible velocities).
2. Collision avoidance and joint limits are represented as a set of inequality constraints.

In our approach, the entire path is iterated toward a solution. Therefore, in this framework, whether the constraint is holonomic or nonholonomic does not make a fundamental difference; a more important distinction is between equality and inequality constraints. Indeed, path planning for redundant robots and mobile robots are treated in exactly the same way in our proposed approach.

The manipulator path planning problem is traditionally based on geometric approaches. (Comprehensive surveys can be found in [1, 2, 3].) The majority of these use the configuration representation of the manipulator and the obstacles and joint limits (as originally proposed in [4]). In this formulation, the path planning problem is reduced to finding a feasible path for a single point. Various tools from geometry, topology and algebra have been utilized to develop methods such as roadmap, cell decomposition, and potential field methods. The drawback of the configuration space approach is that the configuration space can become high dimensional for manipulators with several joints and mapping out the free region for a redundant arm in a cluttered environment is a time consuming process and produces large graphs to be searched. Recently, some work has emerged which focuses on minimizing this growth in computation by incorporating heuristics [5, 6].

The specific issue of redundancy resolution in redundant manipulators has traditionally been addressed by using the Jacobian pseudo-inverse [7, 8, 9, 10], which is a local, or point-wise on the path, approach. This approach is simple to use, has low computational overhead, and can be combined with the local potential field for collision avoidance. The drawbacks include local minima, breakdown near or at singularities, and non-cyclic or even unstable joint motion for cyclic end effector paths [11, 12, 13, 14]. Some global techniques based on optimization have also been proposed [15, 16, 17, 18, 19], but the resulting two-point boundary value problem (TPBVP) is difficult to solve for complex problems.

In the rapidly accumulating literature on nonholonomic motion planning, there appears to be a similar dichotomy of approaches: graph search based methods as in [20, 21, 22, 23, 24, 25, 26] and analytic methods [27, 28, 29, 30, 31, 32, 33, 34, 35]. The former class tends to be computationally inefficient but can handle general constraints, the latter class gives elegant insights into the structure of the solution but may generate impractical paths. Recently, some important insights have been gained by linearizing the kinematics equation about a non-stationary trajectory instead of a fixed equilibrium. It is shown in [36] that the linearized time varying system is frequently controllable, and a locally stable feedback controller can be designed. In [37], a globally stable feedback controller is also found. Based on this approach, a general procedure for designing globally stable time varying feedback controller has been obtained [38].

A common starting point for the analytic approaches to the path planning problem is to pose it as a general nonlinear control problem. Using this framework, we present a new method of solving the path planning problem in this paper. The formulation is based on converting the nonlinear control problem into a nonlinear algebraic equation. The problem then becomes a nonlinear root finding problem in which the dimension of the search space is very high (infinite if $\{u(t) : t \in [0,1]\}$ is taken from an infinite dimensional functional space) compared to the number of equality constraints (for the end point constraint). The nonlinear root finding problem is further converted to an initial value problem (IVP) which is much easier to solve than the usual TPBVP typically arising in the optimization approach. Under some additional assumptions, the IVP can be shown to be wellposed [39] and a variable step size ODE solver is used to propagate the solution. Examples ranging from a front-wheel driven car to triple trailers to nine degree-of-freedom (DOF) manipulators have been successfully tackled [40, 41, 42]. A similar approach has also been proposed independently in [43, 44] for kinematic path planning with only the end point constraint. For this case, the wellposedness property of the IVP is shown to be generic.

Since the dimension of the search space is very high, there are many possible solutions to the root finding problem which results from the path planning problem. For any practical applications, additional constraints must be placed. We have adopted an approach similar to the global exterior penalty function method [45, 46] which converts inequality constraints into a zero finding problem. This differs from the familiar artificial potential field method which is an interior penalty function (or barrier function) method in that the initial guess may be infeasible.

The inequality constrained case can then be combined with the equality constraints into an augmented zero finding problem. Non-configuration space constraints, for example, constraints involving corners of a vehicle. body of the robot, etc., can also be

incorporated in this formulation.

The global path planning approach that we have proposed has been applied to examples involving redundant manipulators and nonholonomic vehicles. We note the following attractive features of this algorithm:

1. Both equality and inequality constraints can be included in this formulation. The constraints can be nonlinear in the configuration variable, so task space constraints (which involve nonlinear kinematic function of the configuration variables) are also allowed.
2. The initial guess does not have to be feasible. The planner iteratively warps the path until all constraints are satisfied.
3. This approach emphasizes feasibility over optimality in contrast to other global approaches. Once a feasible solution is found, optimality can be incorporated as a secondary constraint.
4. The IVP formulation is computationally easier to solve than the TPBVP.
5. There are additional points related specifically to redundant manipulators:
   - Goal task variability: This approach can be used for finding a feasible joint sequence from a fixed initial configuration to a specified Cartesian or joint space goal – the ***path planning*** problem – as well as for global ***redundancy resolution*** along a specified Cartesian path.
   - Singularity robustness: The global nature of the planner avoids the Jacobian singularity problem inherent in local methods. While the controllability about a configuration is lost at the singularity, the algorithm can proceed as long as controllability about the planned path, which is a much looser condition to satisfy, is retained.
   - Incorporation of additional equality constraints: As an application of this capability, the planner can generate cyclic joint space motion for a specified cyclic task space motion.

   There are also additional points related specifically to nonholonomic systems:
   - The planner only requires local controllability about a path in every iteration but does not require controllability about a configuration. This is important since the latter is not satisfied for nonholonomic systems.

The rest of this paper is organized as follows: Section 2 describes the theory behind the proposed algorithm. Section 3 describes the global exterior penalty function method to handle inequality constraints. Section 4 shows a number of simulation examples involving path planning for redundant manipulators and mobile vehicles with trailers.

2 Kinematic Path Planning Subject to Equality Constraints

In this paper, we consider the problem of finding a continuous path that links the specified initial and final configurations while satisfying a set of specified equality and

inequality constraints (which may be either holonomic or nonholonomic). To state this problem precisely, we first represent a kinematic model as a control system:

$$\dot{q}(t) = u(t) \quad ; \quad q(0) = q_0 \tag{1}$$

where $q \in \mathbf{R}^n$ is the configuration variable and $u \in \mathbf{R}^n$ is a pseudo-velocity considered as the control input. Note that since this is a path planning problem, the path variable, t, is arbitrarily normalized to be within the interval $[0, 1]$. There may also be additional holonomic and nonholonomic constraints imposed along the path. If the constraints along the path are directly incorporated in the kinematics equation, these constraints are treated as ***hard constraints.*** If the planner iteratively updates the path until the constraints are within a certain tolerance, these constraints are considered as ***soft constraints.*** We first state the general path planning problem with soft constraints:

> *Define* $\underline{u} = \{u(t) : t \in [0,1]\}$ *and* $\underline{q} = \{q(t) : t \in [0,1]\}$. *Given (1), find* $\underline{u} \in L_2([0,1]; \mathbf{R}^n)$ *such that* $q(t)$ *satisfies (1) and* $\underline{c}(\underline{q}, \underline{u}) = 0$ *where* $\underline{c} : L_2([0,1]; \mathbf{R}^n) \times L_2([0,1]; \mathbf{R}^n) \rightarrow \mathbf{Y}$ *is a given equality constraint function for a specified normed linear space,* $\mathbf{Y}$.

The constraint function $\underline{c}$ may include physical constraints, such as nonintegrable velocity constraints on wheels (nonholonomic) and contact constraints of manipulators (holonomic), or artificial constraint such as the desired end effector path of a redundant manipulator. Pathwise holonomic constraints can be represented as

$$\underline{c}(\underline{q}, \underline{u})(t) = k_1(t, q(t)) \qquad , t \in [0,1] \tag{2}$$

and pathwise nonholonomic constraints are of the form

$$\underline{c}(\underline{q}, \underline{u})(t) = k_2(t, q(t), \dot{q}(t)) \qquad , t \in [0,1] \tag{3}$$

and can not be integrated to a constraint only involving the configuration variable. In many applications, k_2 is linear in $\dot{q}$ and invariant in t, i.e., $k_2(t, q(t), \dot{q}(t)) = K_2(q(t))\dot{q}(t)$. A system may also be subject to what is known as second order nonholonomic constraints (for example, for a manipulator with unactuated joints [47]).

For the path planning problem with hard constraints, the path constraints in (2) and (3) are explicitly removed. In the holonomic case, write (2) in the differential form:

$$\frac{\partial k_1(t,q)}{\partial t} + \frac{\partial k_1(t,q)}{\partial q} u = 0. \tag{4}$$

Denote the Jacobian matrix $\frac{\partial k_1(t,q(t))}{\partial q}$ by $J(t,q)$. If J is a fat matrix (as in the case of redundant manipulators) and is nonsingular, then the constraint can be explicitly incorporated in the kinematics, resulting in

$$\dot{q} = -J^+(t,q)\frac{\partial k_1(t,q)}{\partial t} + \tilde{J}(t,q)v \tag{5}$$

where $J^+(t,q)$ denotes the Moore–Penrose pseudo–inverse of $J(t,q)$, $\tilde{J}(t,q)$ is an arbitrary full rank matrix whose range coincides with the null space of $J(t,q)$, and v is the new effective control variable.

In the nonholonomic case, if the constraint is linear in u, then the constraint can be eliminated, resulting in a new kinematic equation:

$$\dot{q} = \widetilde{K}_2(q)v \tag{6}$$

where $v \in \mathbf{R}^m$ is the new effective control variable (m is the dimension of the null space of $K_2(q)$, assuming it is a constant).

The general planning problem can now be restated as before except the kinematic equation is modified as below to incorporate the path constraints:

$$\dot{q}(t) = h(t, q(t)) + f(t, q(t))u(t) \quad ; \quad q(0) = q_0. \tag{7}$$

The drift term, $h(t, q)$, is zero in the case of nonholonomic constraints linear in $\dot{q}$. The equality constraint function may involve only the end configuration:

$$\underline{c}(\underline{q}, \underline{u}) = q(1) - q_d \tag{8}$$

where q_d is the desired final configuration. In the case of redundant manipulators, joint cyclicity implies $q_d = q_0$.

For the path planning problem as formulated above with either soft or hard constraints, our approach is based on converting the differential description of the problem into an algebraic form. First consider the soft constraint formulation. Recall this case involves a very simple kinematic equation given by (1). For a given initial configuration q_0, $\underline{q}$ can be related to $\underline{u}$ via a linear causal map, $\underline{D}$, and an initial condition vector, $\underline{q}_0$:

$$\underline{q} = \underline{D}\,\underline{u} + \underline{q}_0. \tag{9}$$

Define the constraint error as

$$\underline{y} = \underline{c}(\underline{D}\,\underline{u} + \underline{q}_0, \underline{u}). \tag{10}$$

The path planning problem is now reformulated as a nonlinear root finding problem for $\underline{c}$ as a function of $\underline{u}$ Nonlinear root finding has the reputation being numerically challenging [48, §9.6]. However, the situation here is different than the general case in that the dimension of the search variable, $\underline{u}$, is typically much larger than the dimension of the constraint equation. Consequently, there are a very large number of roots and finding one of these roots is less difficult than the general root finding problem. However, equality constraints alone may not produce physically realizable solution; we shall see how this formulation can be extended to inequality constraints in the next section.

To find a $\underline{u}$ that solves $\underline{y} = 0$, our basic strategy is to lift a path in $\mathbf{Y}$ that connects an initial guess $y(0)$ to zero to a path in $L_2([0,1]; \mathbf{R}^m)$. Then the end point of the path in $L_2([0,1]; \mathbf{R}^m)$ is a solution that satisfies the stated equality constraint $c(\underline{q}, \underline{u}) = 0$. To achieve this, we set up the path iteration equation:

$$\frac{d\underline{y}}{d\tau} = G\frac{d\underline{u}}{d\tau} \tag{11}$$

where

$$G = \nabla_{\underline{q}}\underline{c}\ \underline{D} + \nabla_{\underline{u}}\underline{c}.$$

If G is full rank, or, equivalently, the null space of G^* (the adjoint map of G) is zero, then any one of the following algorithms can be applied to update $\underline{u}$:

$$\frac{d\underline{u}}{d\tau} = G^+ \frac{d\underline{y}_d}{d\tau} + \tilde{G}\xi \tag{12}$$

where G^+ is the Moore–Penrose pseudo–inverse of G, and $\underline{y}_d$ is any desired convergence profile of $\underline{y}$ in τ and $\underline{y}_d(0) = \underline{y}(0)$, for example, we can choose $\underline{y}_d(\tau) = e^{-\alpha\tau}\underline{y}(0)$, for exponential convergence. However, this rate may not correspond to the physical convergence rate since (12) may take much longer to solve. In practice, Eq. (12) may be solved by an ODE solver or discretized with the time step in τ found by line search.

In the hard constraint formulation and with only the end configuration constraint (i.e., $\underline{c}(\underline{q}, \underline{u}) = q(1) - q_d$), the equality constraint error can be written as

$$\underline{y} = F(\underline{u}) - q_d \tag{13}$$

where F maps the control, $\underline{u}$, (for a given q_0) to the final configuration. The analytic form of F is in general difficult to find, and will not be explicitly required. The system (7) is globally controllable if and only if the nonlinear map F is onto, and the system is locally controllable around $\underline{u}$ if and only if $\nabla_{\underline{u}} F(\underline{u})$ is a linear onto map. Local controllability around $\underline{u}$ is equivalent to the controllability of the linear time varying system obtained after linearizing (7) around $\underline{q}$ and $\underline{u}$. The path planning problem can be iteratively solved as before using (12) but with

$$G = \nabla_{\underline{u}} F(\underline{u}).$$

A sufficient condition for convergence is that G is full rank for all τ, or, equivalently, the time varying linearized system with $\underline{u}(\tau)$ is controllable. For systems without drift, such as nonholonomic systems, it has been shown in [43] that this condition is generic.

Equation (12) can be solved by an ordinary differential equation (ODE) solver with an initial guess of $\underline{u}(0)$ such that the full rank condition is satisfied. In addition, $\underline{y}_d(\tau)$ can be used to check the accuracy of the solution, $\underline{y}(\tau)$, at each τ. It can be shown that if $\underline{u}(0)$ is sufficiently close to a solution, then the wellposedness of the IVP (and its convergence) is assured. As stated before, the wellposedness of the IVP in the hard nonholonomic constraint case has been shown to be generic and there has been a surge of recent interest of this technical issue [39]. For the general problem, the wellposedness condition remains an important research topic. In all of our simulation experience involving equality constraints, this has never been a problem. In the next section, when constraints are included through the global exterior penalty function, the issue of wellposedness becomes more severe.

An interesting aspect of this approach is that in (12), ξ does not affect the guaranteed convergence rate (specified by α), though it does affect the way $\underline{u}$ converges. Since the dimension of $\underline{u}$ is much larger than $\underline{y}$, there is much freedom in ξ to affect the eventual convergent solution. For example, ξ may be chosen so that additional constraints in $\underline{u}$ may be satisfied. We have explored choosing ξ via quadratic programming with some success, but more research needs to be done in exploiting this degree of freedom.

There are other possible choices of $\underline{u}$, for example, in the soft constraint case,

$$\frac{d\underline{u}}{d\tau} = G^* \frac{d\underline{y}_d}{d\tau} + \tilde{G}\xi \tag{14}$$

$$\frac{d\underline{u}}{d\tau} = \frac{G^*\underline{y}}{\left\|G^*\underline{y}\right\|^2}\frac{dw_d}{d\tau} + \tilde{G}\xi \qquad w_d = \frac{1}{2}\left\|y_d\right\|^2. \tag{15}$$

The trade–offs between computational load and convergence speeds between these schemes are yet to be fully explored.

The initial guess $\underline{u}(0)$ will clearly affect the convergence. We have not extensively explored an intelligent procedure (perhaps based on past experience) for this selection. In all of our examples, the initial conditions are simply chosen to avoid the rank deficiency of G.

In the algorithm described above, it is critical to be able to calculate G efficiently. Since G relates an infinitesimal variation $\delta\underline{u}$ to a corresponding infinitesimal variation $\delta\underline{c}$, it can be computed based on the linearized kinematic equation (and the constraint equation in the soft constraint case). This procedure is described in detail in [49], and has been used in all of our examples.

3 Kinematic Path Planning Subject to Inequality Constraints

In the previous section, only equality constraints are considered. However, for realistic problems, there are also many inequality constraints which need to be enforced. For problems involving equality constraints only, the obtained path is frequently undesirable. For example, in the case of a tractor with twin–trailers, the front wheel angles of the tractor may go through an unrealistically large range motion, the trailers can become jack–knifed, and the entire vehicle may go through several complete revolutions; in the redundant arm case, the joints can violate joint limits, and there may be self collision or collision with other objects in the workspace. Clearly, if this algorithm is to be used in practical applications, additional constraints must be incorporated. In this section, we present an approach similar to the exterior penalty function method of [45] and which has so far been shown to be very effective in addressing the inequality constraint issue.

There are three basic approaches to address the inequality constraints:

- Convert the inequality constraints into equality constraints by defining a function that is zero when the constraint is satisfied and non–zero when the constraint is not satisfied. Once this is done, the zero finding approach of the previous section can be applied to find a zero solution which corresponds to a feasible path.
- If the feasible region is convex polyhedral, then linear programming or quadratic programming can be used to select the free variable ξ in (12) without affecting the convergence of the equality constraints.
- Pre–warp the vector with a multiplicative potential function $\phi(q)$ which is zero on the boundary of the inequality constraint and positive inside the feasible region. It has recently been pointed out in [44] that as long as the initial path $\underline{q}(0)$ lies strictly within the feasible region, then the same algorithm (12) can be applied to iteratively move $\underline{y}$ to zero.

We have extensive experience with the first two approaches. The first approach has proven to be very effective, though there are currently no (genericity) results on the

full rank gradient condition as in the equality constraint case. The second approach is computationally intensive due to the need to repeatedly solve a constrained optimization problem. It is perhaps best used as a local optimization refinement in conjunction with the first approach. The third approach has just recently been suggested and is computationally untried; we intend to pursue this direction in the proposed research. For the remainder of this section, we will elaborate on the exterior penalty function approach.

Suppose the feasible region is defined by a set of p inequalities:

$$g_1(\underline{q}, \underline{u}) \leq 0 \tag{16}$$

where g_1 is nonlinear and assumed to be as smooth as needed, and $\leq$ is interpreted as a component-wise relationship. These constraints may be directly placed on the configuration variables such as joint limits or on other variables such as the end effector of a manipulator, vehicle boundaries, etc. Without loss of generality, we shall consider the hard constraint case for the discussion below. Let the relationship between $\underline{q}$ and $\underline{u}$ be denoted by $\underline{q} = \underline{F}(\underline{u})$. By substituting this relationship, the constraint inequalities are transformed to

$$g(\underline{F}(\underline{u}), \underline{u}) \leq 0 \tag{17}$$

or more simply expressed as

$$g(\underline{u}) \leq 0. \tag{18}$$

Suppose now we wish to constrain the states of the system to stay within the feasible region defined by the task space. We then define a penalty function corresponding to the i^{th} system state as:

$$z_i = \sum_{j=1}^{N} \begin{cases} \gamma_{ji} g_{ji}^{\lambda}(\underline{u}) & g_{ji}(\underline{u}) > 0 \\ 0 & g_{ji}(\underline{u}) \leq 0 \end{cases} \tag{19}$$

where $\lambda > 1$, g_{ji}'s correspond to the constraint applied to state i at discrete time j and γ_{ji}'s are constant weights. This function is nonzero when state i violates the constraints at any time along the path, and is identically zero only when all constraints are satisfied. For this penalty function, the penalty imposed depends upon the constraint violation raised to the λ power. Therefore, the penalty function approaches infinity as the system makes ever increasing incursion out of the feasible region. Of course many other choices for the constraint function are also possible. For example, another penalty function which we use frequently is the following:

$$z_i = \sum_{j=1}^{N} \begin{cases} \gamma_{ji}(1 - e^{-\lambda g_{ji}(\underline{u})})^r & g_{ji}(\underline{u}) > 0 \\ 0 & g_{ji}(\underline{u}) \leq 0 \end{cases} \tag{20}$$

where $r, > 1$ and $\lambda > 0$. This penalty function is bounded by the sum of the γ_{ji}'s, and is therefore in some sense less harsh than the previous penalty function. In practice we have seen that the second penalty function gives faster convergence than the first, in cases where the vehicle makes large excursions outside of the feasible region. This advantage in convergence stems from the fact that $\nabla_{\underline{u}} z_i(\underline{u})$ is smaller for the second constraint function than for the first when large constraint violations occur, causing

the IVP to be more wellposed for the second constraint function. For either penalty function, the composite constraint vector is defined as $z = \left[\begin{array}{ccc} z_1 & \ldots & z_n \end{array} \right]^T$.

We now have a situation similar to the unconstrained case: ***Find*** $\underline{u}$ ***so that*** $\underline{y} = 0$ ***and*** $z = 0$. The same approaches described in Section 2 can now be applied. Using the hard constraint formulation, the new differential equation in $\underline{u}$ now becomes:

$$\frac{d\underline{u}(\tau)}{d\tau} = G_1^+(\underline{u}(\tau)) \left[\begin{array}{c} \underline{y}_d \\ z_d \end{array} \right] + \widetilde{G_1}(\underline{u}(\tau))\xi(\tau) \tag{21}$$

where,

$$G_1(\underline{u}) \triangleq \left[\begin{array}{c} G \\ \nabla_{\underline{u}} z(\underline{u}) \end{array} \right] \tag{22}$$

and $\underline{y}_d$ and z_d describe the desired path in τ that links the initial error to zero. The gradient of $z(\underline{u})$ can be easily obtained via chain rule and the linearized kinematic equation [49]. The convergence of this algorithm now requires the full rank condition on G_1 in each iteration. Note that in (21), there is again a free parameter $\xi(\tau)$. The extra degrees of freedom offered by $\xi(\tau)$ may be used to satisfy additional optimality considerations.

The penalty function formulation we have presented allows for a wide range of constraint types. Up to this point we have considered constraints which apply directly to the configuration variable of the system, but the formulation allows constraints to be applied to non–configuration variables as well. In any practical application it is not enough just to constrain the origin of the system inside a feasible region. The boundaries of the system must stay inside as well. For instance, for a single planar body, we relate any point p on the boundary to the configuration variable q by the nonlinear transformation:

$$\left[\begin{array}{c} p_x \\ p_y \end{array} \right] = \left[\begin{array}{c} q_1 \\ q_2 \end{array} \right] + \left[\begin{array}{cc} \cos(q_4) & -\sin(q_4) \\ \sin(q_4) & \cos(q_4) \end{array} \right] \left[\begin{array}{c} r_x \\ r_y \end{array} \right] \tag{23}$$

where r_x and r_y are the x and y positions of the boundary point in the body frame. The same penalty function which applied to q before now applies to p, $g_j(p(q_j(\underline{u})))$. A similar transformation can be obtained to relate boundary points of multi–bodied systems to the configuration variable.

The penalty function formulation also allows for a wide range of feasible region types. For instance, when the feasible region is polyhedral, g takes on particularly simple form:

$$g(\underline{q}) = A\underline{q} - b.$$

We have looked at this type of constraint extensively. In fact, this constraint type is always used when limiting wheel or jackknife angles on wheeled vehicles. This type of constraint is simple to use but the feasible region is necessarily convex.

The penalty function can be used to enforce both convex and non–convex, non–polyhedral constraints. For instance, supposed it is desired to drive the system while keeping boundary points outside of an circular region. The constraint g at each time j can then be expressed as:

$$g_j(p) = -(p_x - x_0)^2 - (p_y - y_0)^2 + r^2$$

where x_0 and y_0 are constants, r is the radius, and p is a boundary point as described in (23). We have had much success in using this formulation with cars and tractor-trailer vehicles and present an example in the next section.

In the preceding discussion, system constraints are expressed as analytic functions of the configuration variable. But again, in practical situations it may not be possible to adequately represent the constraints by analytic functions. In particular, for highly unstructured environments it may require great effort to find an appropriate analytic function, and once found, it may be computationally intensive to apply it. Therefore, we propose a method based upon the contour map built up from the task space obstacle boundaries. First, suppose that the feasible region is defined by a set of p inequalities of the form:

$$g_2(\underline{F}(\underline{u})) \leq 0 \tag{24}$$

where g_2 is nonlinear, smooth as needed, and $\leq$ is component-wise. The gradient of g_2 with respect to $\underline{u}$ is then obtained by applying the chain rule,

$$\nabla_{\underline{u}} g_2 = [\nabla_{\underline{u}} \underline{F}]^T \nabla_{\underline{q}} g_2. \tag{25}$$

Note that in this equation, $\nabla_{\underline{u}} \underline{F}$ is independent of the constraints and that $\nabla_{\underline{q}} g_2$ depends solely upon the constraints. Therefore, rather than compute g_2 and $\nabla_{\underline{q}} g_2$ explicitly using analytic functions, it is possible to use a lookup table to compute these values. For g_2, the lookup table contains the cost for each point in a grid covering the task space. For $\nabla_{\underline{q}} g_2$ the table contains the gradient components for each point on the grid. As before, this formulation will also work for non-configuration variables. Since the exterior penalty function approach is used, the contour map is zero everywhere inside the feasible region and non-zero outside. We have just begun using this method for path planning and have so far had good success. An example of the contour map method applied to a car is given in the next section.

In this, and the previous sections, we have used the notation $\underline{u}$ to denote the set of control inputs applied to the system at each point in time along the path. For the continuous case this set contains an infinite number of elements. Therefore $\underline{u}$ in vector form would be an infinite dimensional vector. However, in order to implement the algorithm on a digital computer $\underline{u}$ needs to have finite length. The first obvious step would be to discretize $u(t)$ using the standard basis. We have used this approach extensively and have obtained good results from it. Another approach we have tried is to use the first N elements of the Fourier basis to approximate $u(t)$. In this formulation, for each discrete time j we represent the control input as:

$$u_j = \Phi(j\Delta t)\underline{\lambda} \tag{26}$$

where Φ is a matrix containing the Fourier basis elements and $\underline{\lambda}$ is the constant vector of Fourier coefficients. In this formulation, $\underline{\lambda}$ uniquely describes the control over all time, since it is independent of time. Now for a given initial configuration $\underline{q}_0$, $\underline{q}$ can be related to $\underline{\lambda}$ by a causal map, $\underline{F}_1$:

$$\underline{q} = \underline{F}_1(\underline{\lambda}). \tag{27}$$

Furthermore, substituting $\underline{F}_1$ for $\underline{F}$ and $\underline{\lambda}$ for $\underline{u}$ in all previous equations, we see that all previous methods still work. The full advantages and disadvantages in using

the Fourier basis formulation rather than the discrete basis is unknown at present. However, in the case of driving a car around a circle, we have observed that the discrete basis formulation has great difficulty arriving at a solution whereas the Fourier basis formulation solves the problem with relative ease.

4 Examples

We have applied the algorithm to a large number of computer simulation examples and experimentally to a mobile robot. For illustration purposes, we include several examples involving three different wheeled vehicles and a point–to–point path planning of a 4R planar arm and a 9DOF arm like the one in our lab (see next section on a fuller description of this arm).

In the first example, we consider the parallel parking of a double tractor–trailer. A true double tractor–trailer, like those seen driving on the highways, actually consists of a tractor with three trailers: two long trailers connected by a comparatively short trailer. The short trailer, or dolly, makes any backing–up situation extremely difficult in that small backward motions can produce large jackknife angles between the dolly and the trailers. In fact, it is known that with a human operator one must, in general, disconnect the end trailer and dolly before any backward motions are attempted.

In the example below, shown in Fig. 1–2 below, the front wheels are constrained to $\pm 35°$ and the jackknife angles (the jackknife angle is defined as the angle between the center line of one trailer and the center line of an adjacent trailer) are constrained to $\pm 50°$. The limits on the jackknife angles are sufficient to ensure that none of the trailers collide. In this example the algorithm required about four hundred fifty evaluations of the right hand side of the differential equation to achieve a tolerance on the potential fields of 0.01. The penalty function method used here is simply the first convex, polyhedral method mentioned in the previous section.

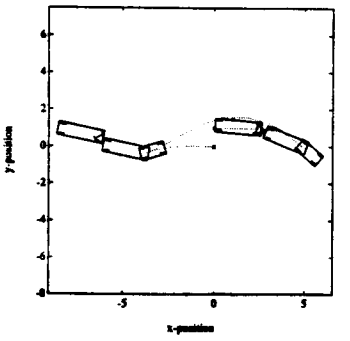


Figure 1: Parallel Parking Path of a Double Tractor–Trailer with Constraints

The next example is presented to demonstrate the versatility of the algorithm with regard to the type of constraint applied. In this example, it is desired to drive a tractor–trailer vehicle around a circular obstacle while remaining within a rectangular region limiting motion along the x–axis. To further complicate the example, two other

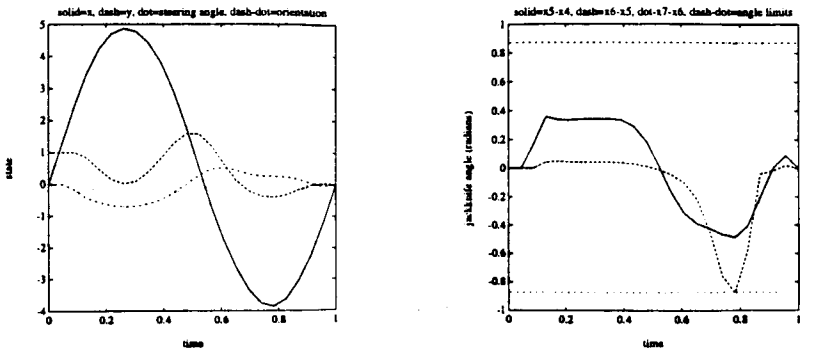


Figure 2: Parallel Parking Path of a Double Tractor–Trailer with Constraints

circular obstacles are included above and below the first one. In addition, these task space constraints are applied to fourteen different non–configuration variable points around both the boundary of the tractor and the boundary of the trailer. The steering angle is constrained to $\pm 20°$, and the jackknife angle is constrained to $\pm 50°$. We therefore have in this one example, all of the different types of constraints mentioned in the previous section: convex polyhedral (steering, jackknife, and x-position), non–convex non–polyhedral (circular obstacles), and non–configuration variable (boundary point) constraints. The Fourier basis representation of the control input, with thirty three basis elements, was also used in this example. Using a steepest descent method with a Golden section search, the algorithm took fifty four evaluations of the right hand side to converge. The initial path in this case, drives straight through the obstacle. The final result is as shown below in Fig. 3.

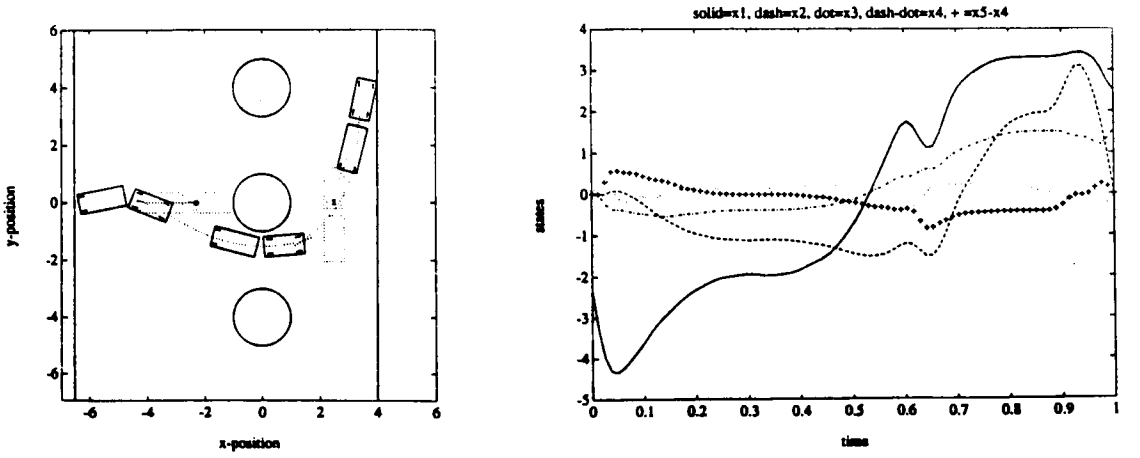


Figure 3: Tractor–Trailer Example with Non–Convex Task Space Constraints

The next simulation example uses the contour map method to calculate the penalty

function. In this example, it is desired to drive a car around a circular obstacle while remaining inside of a circular region. Seven points around the boundary of the car are used for non-configuration variable constraints. The front wheels are constrained to $\pm 15°$. The Fourier basis was once again used to represent the control input along the path. This example took about one hundred evaluations of the right hand side using a discretized version of (21) with Golden section search to find the optimal step size. The initial path drives straight through the obstacle. Below, the first set of plots, in Fig. 4, show the car and path in the context of the task space. The second set of plots, in Fig. 5 show the contour map used in this example.

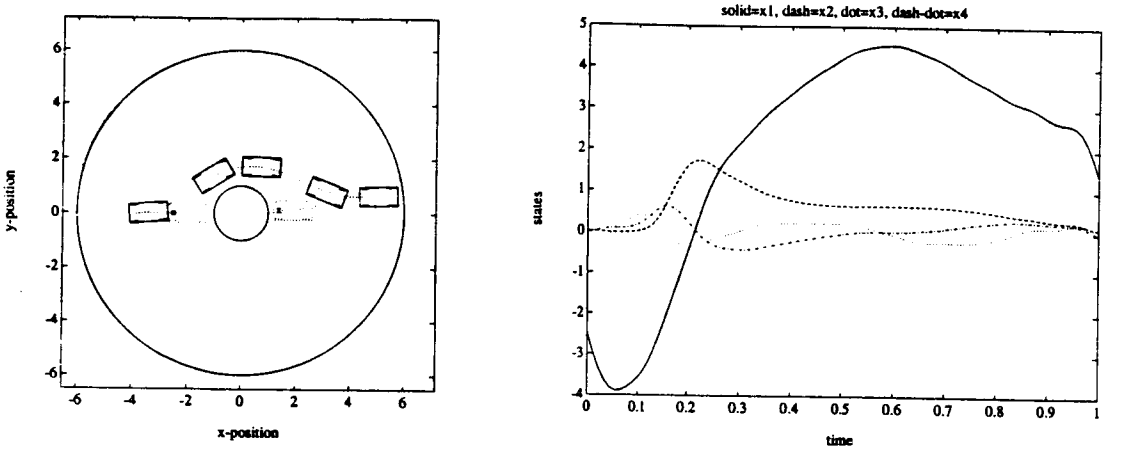


Figure 4: Path Planning with Contour Map Method

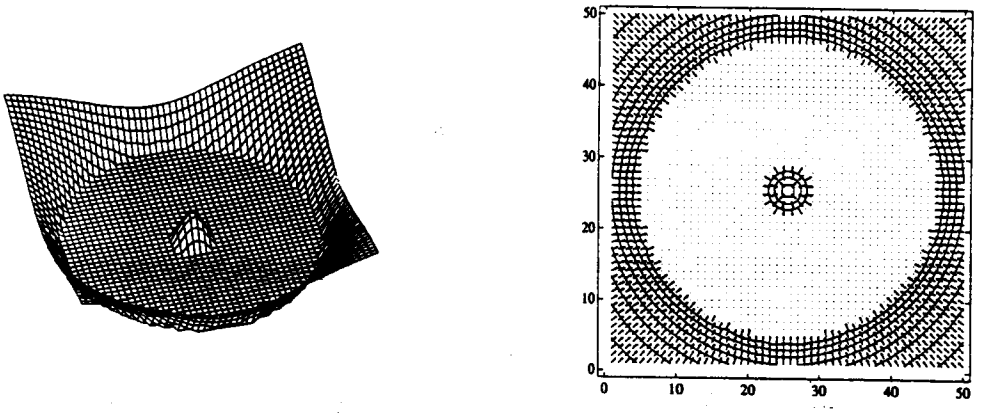


Figure 5: Plots of Exterior Penalty Function and Contour Map

In the final wheeled vehicle example, we present experimental results of applying the

path planning algorithm to an actual mobile robot. The robot is actually a one quarter scale model car, complete with a passive suspension system, on–board IBM compatible 486 33–MHz computer, and wheel encoders for use in dead–reckoning feedback. The desired path was a simple parallel parking path with constraints only on the steering angle of $\pm 15°$ which is the physical limit of the car. The purpose of this experiment was to determine if a real robot can actually follow a path generated by the planning algorithm. In this example, the path was generated, and then a velocity profile was imposed. A simple, Lyapunov function based, feedback controller was used to try and track the path. Although there is some error in tracking the path, it is obvious from the plots below, in Fig. 6–7 that the path is such that the real car can follow it. The tracking errors are due in part to error in the dead–reckoning feedback and to the controller used.

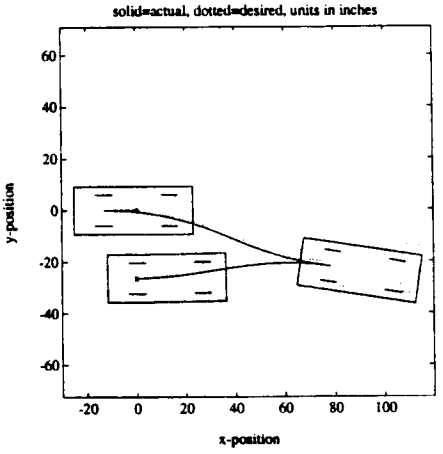


Figure 6: Parallel Parking of Catmobile: Experimental vs. Simulation Path

Various examples of redundant manipulators are presented which incorporate both joint and task space inequality constraints. Key features of this approach include various possible goal task specifications (from end-point only to entire path specification), joint path cyclicity, and robustness to manipulator singularities. Inequality constraints are handled by the global penalty functions and a linear inequality set description as shown in Sections 3. Simulations were performed using Matlab with computationally intensive routines coded as Matlab-callable C routines (cmex files). Note that the end effector constraint is treated as a soft constraint here as compared to the nonholonomic constraint considered above which is treated as a hard constraint.

The first example illustrates the joint sequence for a 3R planar arm required to trace out the tip path shown while remaining within a set of task space boundaries (Fig. 8). The path shown in Fig. 8 includes an intermediate joint vector in which links 2 and 3 become aligned – equivalent to a pose switch for spatial arms. Local planning methods typically encounter difficulty in handling pose changes since they correspond to arm singularities, and the arm Jacobian losing rank. The present algorithm executes the pose switch smoothly, while tracking the desired tip path. In the next situation (Fig. 9) a 4R arm is required to traverse to specified coordinates X_f. The obstacles are chosen

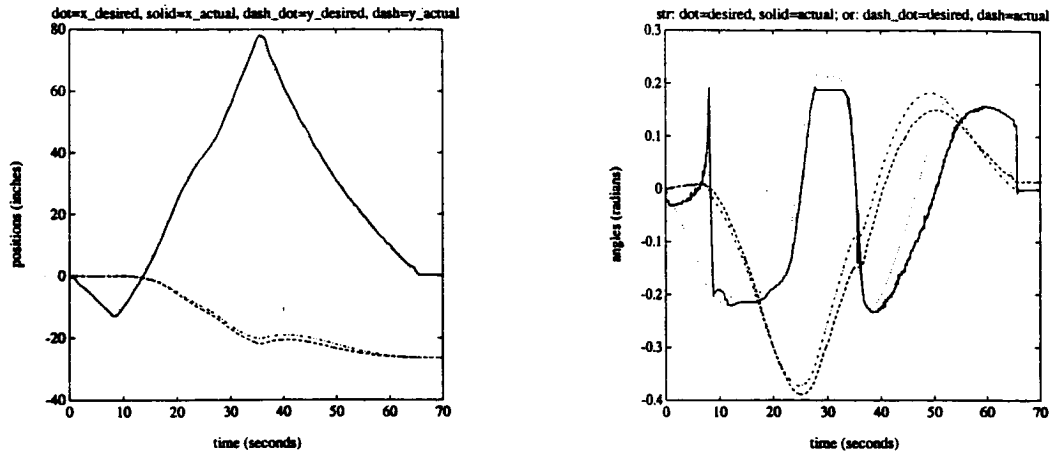


Figure 7: Parallel Parking of Catmobile: Experimental vs. Simulation State Trajectories

to provide a relatively narrow opening through which the arm must pass. Typically, this type of scenario is very challenging for planners based on purely local potential field formulations due to a local minimum formed in the space between Obstacles 1 and 2. By iterating from an intermediate path sequence generated for the goal endpoint without consideration of the obstacles, it is possible to warp the intermediate path to meet all the constraints. It should also be noted that the final path cannot be accomplished without switching the pose along the way.

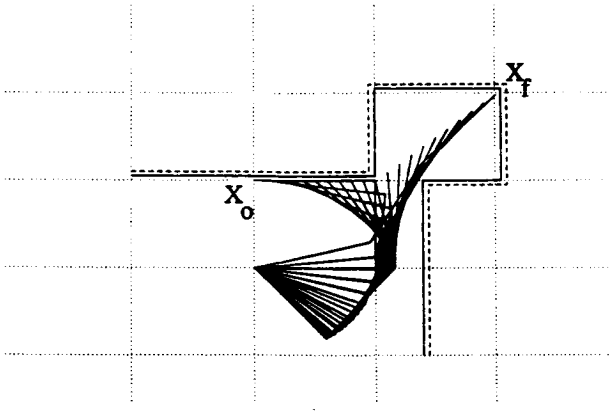


Figure 8: Cartesian Tip Path

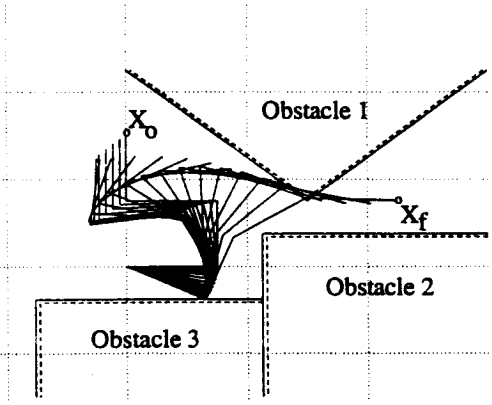


Figure 9: Obstacle Avoidance Problem

The ability to incorporate joint path cyclicity as an equality constraint is illustrated by an example where a 3R arm must trace out the boundaries of a Cartesian square path, within a restricted workspace (Fig. 10). Path planning for cooperating arms manipulating a common payload imposes an additional kinematic closure constraint on the arm tips. Fig. 11 shows the path sequence generated for a pair of 3R planar manipulators required to move the connecting linkage from its initial position of $(-1, 2, 0°)$ to

a goal position of $(2, 2, 45°)$. In the planar examples shown, the apparent collisions of the links with the obstacle boundaries arise out of the fact that checking is performed for the tips of each link only. A more complete collision detection procedure is being developed for spatial arm planning.

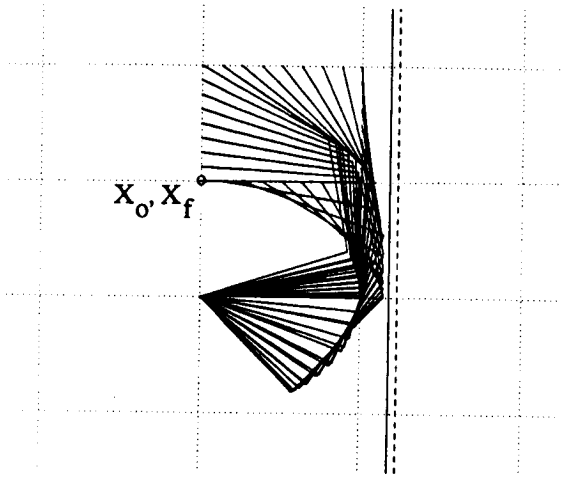


Figure 10: 3R Cyclic Path

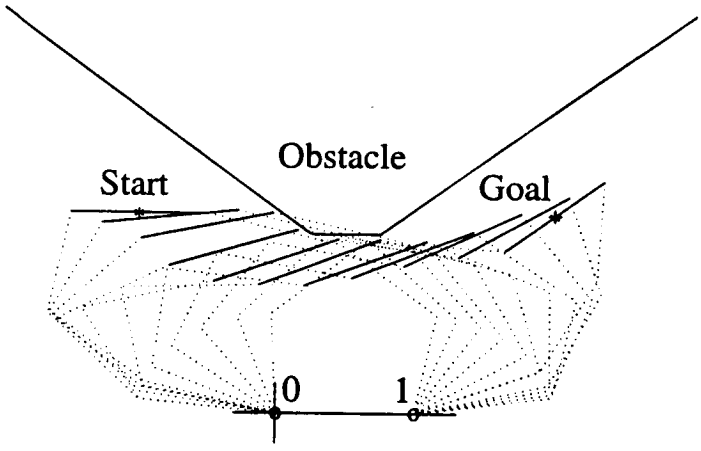


Figure 11: Cooperating Arm Problem

The remaining examples illustrate applications to a spatial redundant 9 DOF arm. This consists of a PUMA 600 manipulator mounted on a platform providing additional linear, rotate and tilt joints. (See [50] for details of the CIRSSE dual-arm robotic testbed.) Fig. 12 shows the output sequence for joint path end-point planning to a specified task space position/orientation in the presence of an obstacle. Fig. 13 shows the path sequence generated for a path following task incorporating a straight line translation coupled with a rotation of 180° about the tip Y axis. Because of the manipulator joint limits, this can only be accomplished by switching the PUMA's pose from ***elbow-up*** to ***elbow-down*** at some intermediate point. As in the planar examples, the present algorithm accomplishes a smooth transition between these arm configurations. The final example (Fig. 14) illustrates a cyclic joint path sequence computed for the arm to follow a circular task space path while maintaining a fixed tool orientation (perpendicular to the plane of the circle).

For the cases shown, the planar examples required from seven to ten iterations to reduce the task error and meet all the constraints, while the spatial examples ranged from ten to twenty-five iterations. Discretization levels ranged from $N = 10$ to 40 in size. For the 9 DOF redundant arm planning scenario, this represents a small computational load for the Sun SPARC-station used. workstations. For problems of larger size, this technique can be applied recursively to the desired discretization level, thereby keeping the individual iteration array operations small. The ability to apply the algorithm to the global problem (even at the coarsest resolution) is essential for

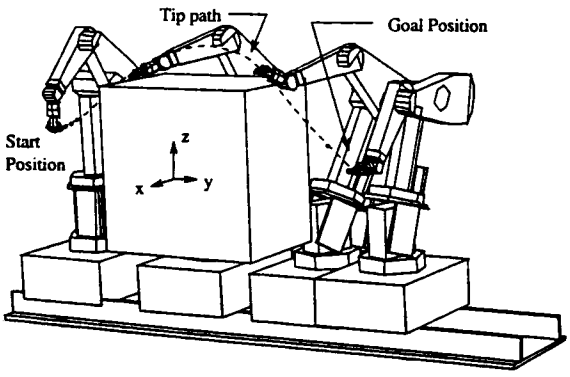


Figure 12: 9 DOF Path Planning

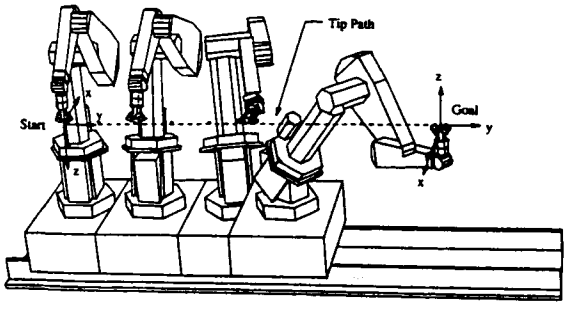


Figure 13: 9 DOF Path Following

developing globally feasible solutions to the constrained path planning problem.

5 Conclusions

Kinematic path planning for robots is a key step in their effective utilization on earth and in space. The various types of constraints such as holonomic and nonholonomic, and equality and inequality constraints pose particular challenges. In this paper, we consider a novel and promising method which warps the entire path until all constraints are satisfied. The main approach is to convert the differential (local) kinematic relationship to an algebraic (global) equation. With emphasis placed on feasibility rather than optimality, we obtain an initial value problem rather than two–point boundary value problem typically arisen in a global optimization approach. This formulation is general enough to include both redundant manipulators and nonholonomic systems, and combination of the two; these topics are traditionally treated separately due to their unique properties when local algorithms are applied. The inequality constraints are handled through a global exterior penalty function method, which allows for non–polyhedral constraints as well as constraints on non–configuration variables. For the future research, we will address the following fundamental issues related to this promising algorithm:

1. Develop conditions of convergence of the algorithm based on the wellposedness of the initial value problem. Also develop strategy to proceed when the algorithm fails to converge due to the rank deficiency of G in (12).
2. Develop an algorithm to adaptively adjust the path discretization step size based on the local discretization error.
3. Incorporate optimality as a secondary criterion.
4. Improve the sensitivity and robustness of the algorithm with respect to imperfection in the kinematic model.
5. Implement the planner on various experimental platforms.

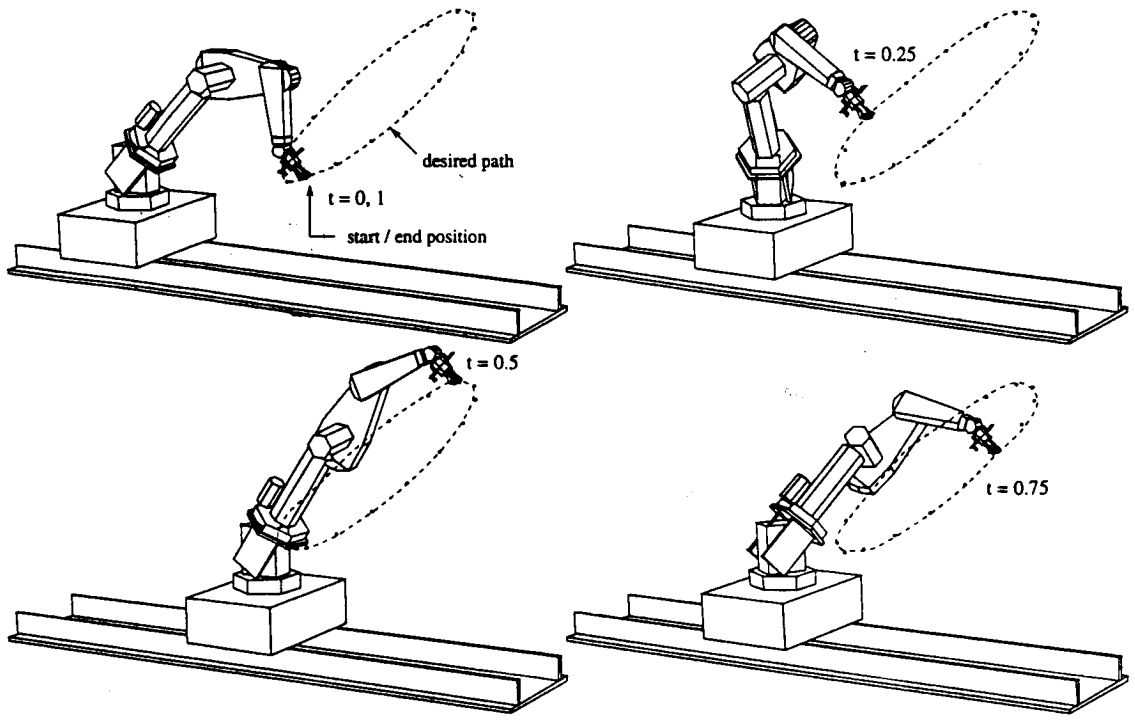


Figure 14: Spatial 9 DOF Cyclic Cartesian Path

Acknowledgment

This work is supported in part by the Center for Intelligent Robotic Systems for Space Exploration and New York State Center for Advanced Technology for Automation and Robotics at Rensselaer Polytechnic Institute.

References

[1] Lozano-Pérez T. and Taylor R.H. Geometric issues in planning robot tasks. In ***Problems of Robotics***. MIT Press, Cambridge, Mass., 1989.

[2] J.C. Latombe. ***Robot Motion Planning***. Kluwer International Series in Engineering and Computer Science: Robotics: Vision, Manipulation and Sensors. Kluwer Academic Publishers, 1991.

[3] Hwang Y.K. and Ahuja N. Gross motion planning – a survey. ***ACM Computing Surveys***, 24(3):219–291, 1992.

[4] Lozano-Pérez T. and Wesley M.A. An algorithm for planning collision free paths among polyhedral obstacles. ***Communications of the ACM***, 22(10):560–570, October 1979.

[5] Mazer E. Lozano-Pérez T., Jones J.L and O'Donnell P.A. Task-level planning of pick–and–place robot motions. ***Computer***, 22(3):21–29, March 1989.

[6] Weaver J.M. and Derby S.J. A divide-and-conquer method for planning collision-free paths for cooperating robots. In *AIAA Space Programs and Technologies Conference, Paper AIAA-92-1722*, Huntsville, AL, 1992.

[7] Whitney D.E. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, MMS-10(2):47–53, 1969.

[8] Khatib O. Dynamic control of manipulators in operational space. In *6th IFToMM Congress on Machines and Mechanisms*, New Delhi, 1983.

[9] Vukobratovic M. and Kircanski M. A dynamic approach to nominal trajectory synthesis for redundant manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-14, 1984.

[10] Yoshikawa T. Manipulability of robot mechanisms. *International Journal of Robotics Research*, 4(2):3–9, March 1985.

[11] Hollerbach J. Baillieul J. and Brockett R. Programming and control of kinematically redundant manipulators. In *Proc. 24th IEEE Conf. Decision and Control*, pages 768–774, Las Vegas, NV, December 1984.

[12] Hollerbach J.M. and Suh K.C. Redundancy resolution of manipulators through torque optimization. In *Proc. 1985 IEEE Robotics and Automation Conference*, pages 308–315, St. Louis, MO, March 1985.

[13] Baillieul J. Martin D.P. and Hollerbach J.M. Resolution of kinematic redundancy using optimization techniques. *IEEE Transactions on Automatic Control*, 5(4):529–533, August 1989.

[14] A. De Luca. Redundant Robots: Introduction to Chapter IX. In M. Spong and F. Lewis, editors, *Robotics*. IEEE Press, 1992.

[15] D.P. Martin, J. Baillieul, and J.M. Hollerbach. Resolution of kinematic redundancy using optimization techniques. *IEEE Transactions on Robotics and Automation*, 5(4), August 1989.

[16] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison–Wesley Series in Electrical and Computer Engineering: Control Engineering. Addison–Wesley Publishing Company, 1991.

[17] Suh K.C. and Hollerbach J.M. Local versus global torque optimization of redundant manipulators. In *Proc. 1987 IEEE Robotics and Automation Conference*, pages 619–624, Raleigh, NC, March 1987.

[18] Kazerounian K. and Nedungadi A. Redundancy resolution of robotic manipulators at the acceleration level. In *The 7th World Congress on the Theory of Machines and Mechanisms*, pages 1207–1211, Sevilla, Spain, September 1987.

[19] Wang Z. and Kazerounian K. A general formulation for redundancy resolution of serial manipulators. In *Proc. 1990 ASME Design Technical Conference*, pages 373–379, Proc. 1990 ASME Design Technical Conference, September 1990.

[20] P. Jacobs and J. Canny. Robust motion planning for mobile robots. In *1991 IEEE R&A Workshop on Nonholonomic Motion Planning*, Sacramento, CA, April 1991.

[21] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. In *Proc. IEEE 5th Int. Conf. on Advanced Robotics*, pages 1012–1027, Pisa, Italy, June 1991.

[22] P. Jacob, J.-P. Laumond, and M. Taix. A complete iterative motion planner for a car–like robot. *Journees Geometrie Algorithmique*, 1990.

[23] L. Dorst, I. Mandhyan, and K. Trovato. The geometrical representation of path planning problems. Technical report, Philips Laboratories North American Philips Corp., Briarcliff Manor, New York, 1991.

[24] B. Mirtich and J. Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proc. 1992 IEEE Robotics and Automation Conference*, pages 2533–2540, Nice, France, May 1992.

[25] T. Hague and S. Cameron. Motion planning for nonholonomic industrial robot vehicles. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1275–1280, Osaka, Japan, November 1991.

[26] T. Fraichard and C. Laugier. On line reactive planning for nonholonomic mobile in a dynamic world. In *Proc. 1991 IEEE Robotics and Automation Conference*, pages 432–437, Sacramento, CA, April 1991.

[27] R.W. Brockett. Control theory and singular Riemannian geometry. In *New Directions in Applied Mathematics*, pages 11–27. Springer–Verlag, New York, 1981.

[28] Z.X. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation*, 6(1):62–72, February 1990.

[29] G. Lafferriere and H.J. Sussmann. Motion planning for controllable systems without drift: A preliminary report. Technical Report SYCON–90–04, Rutgers Center for Systems and Control, June 1990.

[30] R.M. Murray and S.S. Sastry. Steering nonholonomic systems using sinusoids. In *Proc. 29th IEEE Conference on Decision and Control*, pages 2097–2101, Honolulu, HI, 1990.

[31] Z. Li and L. Gurvits. Theory and applications of nonholonomic motion planning. Technical report, New York University, Courant Institute of Mathematical Sciences, July 1990.

[32] Y. Nakamura and R. Mukherjee. Nonholonomic motion planning of space robots. In *Proc. 1990 IEEE Robotics and Automation Conference*, Cincinnati, OH, 1990.

[33] Z. Vafa and S. Dubowsky. On the dynamics of manipulators in space using the virtual manipulator approach. In *Proc. 1987 IEEE Robotics and Automation Conference*, Raleigh, NC, March 1987.

[34] A.M. Bloch, N.H. McClamroch, and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. In *Proc. 29th IEEE Conference on Decision and Control*, pages 1312–1314, Honolulu, HI, 1990.

[35] B. d'Andrea Novel, G. Bastin, and G. Campion. Modelling and control of nonholonomic wheeled mobile robots. In *Proc. 1991 IEEE Robotics and Automation Conference*, pages 1130–1135, Proc. 1991 IEEE Robotics and Automation Conference, April 1991.

[36] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proc. 1990 IEEE Robotics and Automation Conference*, pages 384–389, Cincinnati, OH, May 1990.

[37] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proc. 1991 IEEE Robotics and Automation Conference*, Sacramento, CA, April 1991.

[38] C. Samson and K. Ait-Abderrahim. Feedback stabilization of a nonholonomic wheeled mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1242–1247, Osaka, Japan, November 1991.

[39] H.J. Sussmann and Y. Chitour. A continuation method for nonholonomic path finding problem. IMA Workshop on Robotics, January 1993.

[40] A. Divelbiss and J.T. Wen. A perturbation refinement method for nonholonomic motion planning. In *Proc. 1992 American Control Conference*, Chicago, IL, June 1992.

[41] A. Divelbiss and J.T. Wen. Nonholonomic motion planning with constraint handling: Application to multiple–trailer vehicles. In *Proc. 31th IEEE Conference on Decision and Control*, Tucson, AZ, December 1992.

[42] S. Seereeram and J.T. Wen. A global approach to path planning for redundant manipulators. Accepted for publication in IEEE Trans. on Robotics and Automation, 1993.

[43] E.D. Sontag and Y. Lin. Gradient techniques for systems with no drift. In *Proceedings of Conference in Signals and Systems*, 1992.

[44] E.D. Sontag. Non–singular trajectories, path planning, and time–varying feedback for analytic systems without drift. IMA Workshop on Robotics, January 1993.

[45] Dorny C.N. *A Vector Space Approach to Models and Optimization.* R. E. Krieger Publishing Co., Florida, 1986.

[46] D.G. Luenberger. *Linear and Nonlinear Programming.* Addison–Wesley, 2 edition, 1984.

[47] G. Oriolo and Y. Nakamura. Free–joint manipulators: Motion control under second–order nonholonomic constraints. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Osaka, Japan, November 1991.

[48] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge, U.K., 1986.

[49] A. Divelbiss and J.T. Wen. A perturbation refinement method for nonholonomic motion planning. CAT report #10, Rensselaer Polytechnic Institute, March 1992.

[50] J.F. III Watson, D.R. Lefebvre, S.H. Murphy A.A. Desrochers, and K.R. Fieldhouse. Testbed for cooperative robotic manipulators. In A.A. Desrochers, editor, *Intelligent Robotic Systems for Space Exploration.* Kluwer Academic Publishers, 1992.

Application of the CIRSSE Cooperating Robot Path Planner to the NASA Langley Truss Assembly Problem

Jonathan M. Weaver and Stephen J. Derby
Rensselaer Polytechnic Institute
Troy, New York

Application of the CIRSSE Cooperating Robot Path Planner to the NASA Langley Truss Assembly Problem

Jonathan M. Weaver
Research Assistant

Stephen J. Derby
Associate Professor

Center for Intelligent Robotic Systems for Space Exploration
Department of Mechanical Engineering, Aeronautical Engineering, and Mechanics
Rensselaer Polytechnic Institute, Troy, New York 12180-3590

Abstract

A method for autonomously planning collision free paths for two cooperating robots in a static environment has been developed at CIRSSE. The method utilizes a divide-and-conquer type of heuristic and involves non-exhaustive mapping of configuration space. While there is no guarantee of finding a solution, the planner has been successfully applied to a variety of problems including two cooperating 9 dof robots.

Although developed primarily for cooperating robots, the method is also applicable to single robot path planning problems. A single 6 dof version of the planner has been implemented for the truss assembly task at NASA Langley's Automated Structural Assembly Lab (ASAL). The results indicate that the planner could be very useful in addressing the ASAL path planning problem and that further work along these lines is warranted.

1 Introduction

The robot path planning problem involves determining if a continuous and obstacle avoiding path exists between start and goal positions, and, if so, to find such a path. The complexity of the path planning problem has been shown to be exponential in the number of dof [1, 2]. A review of the many path planning techniques is beyond the scope of this paper. Reference [3] presents a recent survey paper on the subject.

A method has been developed at Rensselaer's Center for Intelligent Robotic Systems for Space Exploration (CIRSSE) to autonomously plan collision free paths for two robots working cooperatively in a known, static environment [4, 5, 6]. Cooperation refers to the case whereby both robots simultaneously grasp and manipulate a common, rigid, payload. The planner is based around a divide-and-conquer heuristic aimed at traversing c-space while performing selective mapping on an as-needed basis. This path planner has been applied to the CIRSSE testbed. The testbed consists of two 9 dof robots, each of which consists of a 3 dof platform and a 6 dof Puma. A sample path found by the cooperating 9 dof planner is shown in Figure 1. This example required approximately 10 minutes solution time on a SparcStation 1.

Although developed primarily for the cooperating robot case, the c-space traversal heuristic around which the planner is based may also be applied to single robot path planning problems. This paper discusses a single arm version of the planner which was implemented for the truss assembly task at NASA Langley's Automated Structural Assembly Lab (ASAL). The purpose of the implementation was to assess the potential usefulness of the planner for the ASAL path planning problem.

The ASAL path planning problem is described in Section 2. Our path planning strategy is discussed in Section 3. Sections 4 and 5 present some implementation details and results for application of the planner to the ASAL path planning problem. Section 6 presents some conclusions and areas for future work.

2 Problem Statement

A CimStation model of NASA Langley's ASAL is shown in Figure 2 (CimStation model provided by NASA Langley). The system consists of a 6 dof Merlin robot, shown in Figure 3, mounted to a xy-positioning table (referred to as the carriage), and a turntable. The turntable includes a triangular platform which can rotate around a vertical axis through its center. The Merlin robot is kinematically similar to a Puma. The objective of the ASAL is to assemble truss structures consisting of 102 2 meter long struts. Such a truss is illustrated in Figure 4. The truss is assembled upon the turntable of the ASAL by positioning the carriage and the turntable such that the Merlin may take each strut from a canister near the base of the Merlin and install it in its final position in the assembly.

The ASAL path planning problem as addressed herein is defined as follows: Given a carriage and turntable position for each strut, determine a suitable path for the Merlin to safely move the robot and its payload from a start position to a prescribed goal position. The start position is above the canister holding the as-yet unassembled struts. The goal position for each strut is taken as 10 cm from the final position in the negative of the approach direction. The assembly sequence is as specified by NASA Langley.

It is assumed that feasible and collision free start and goal joint configurations of the robot are known.

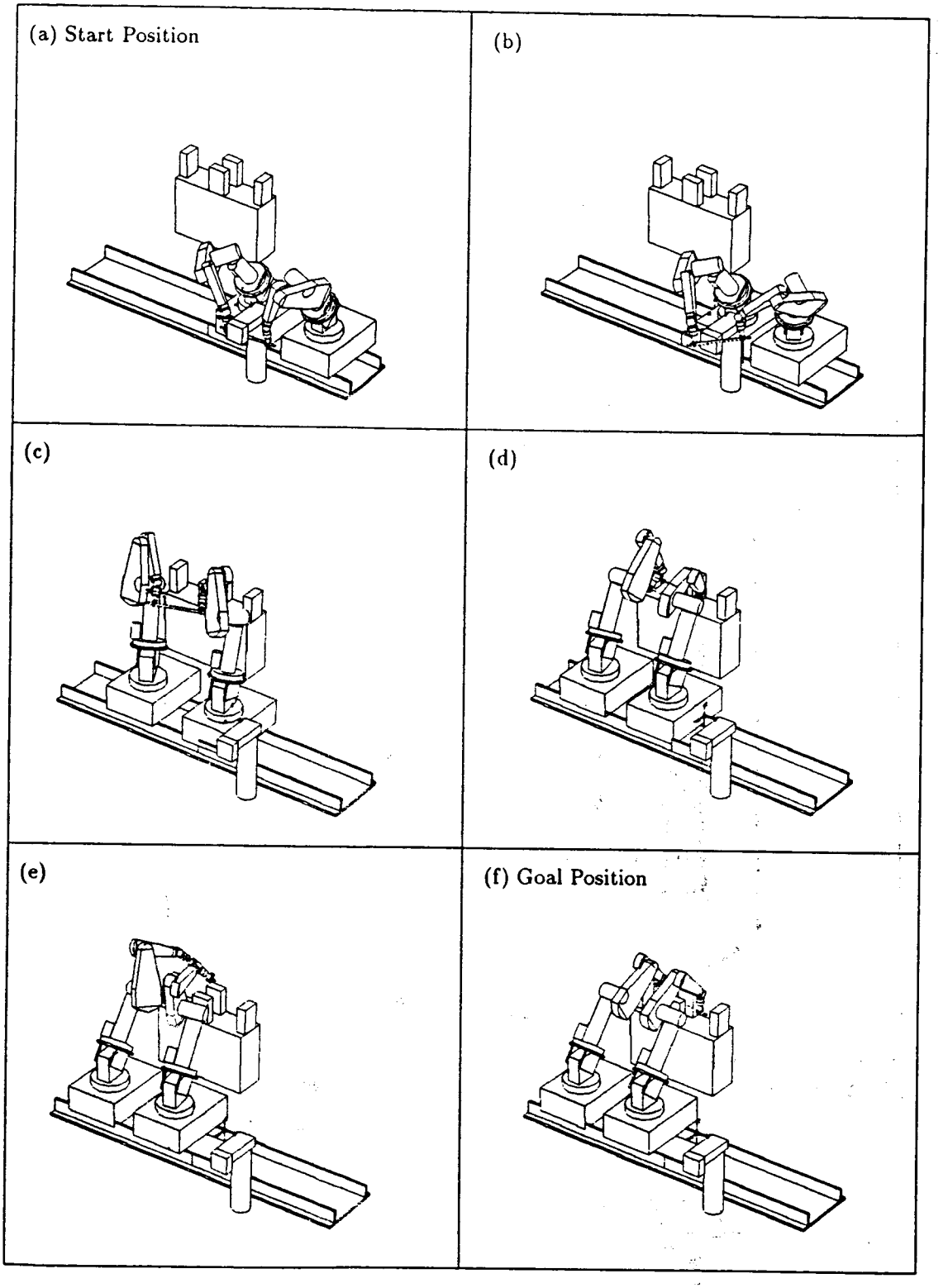
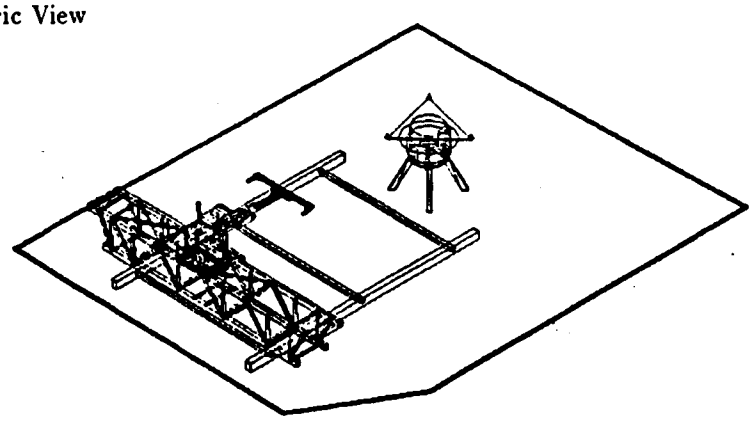
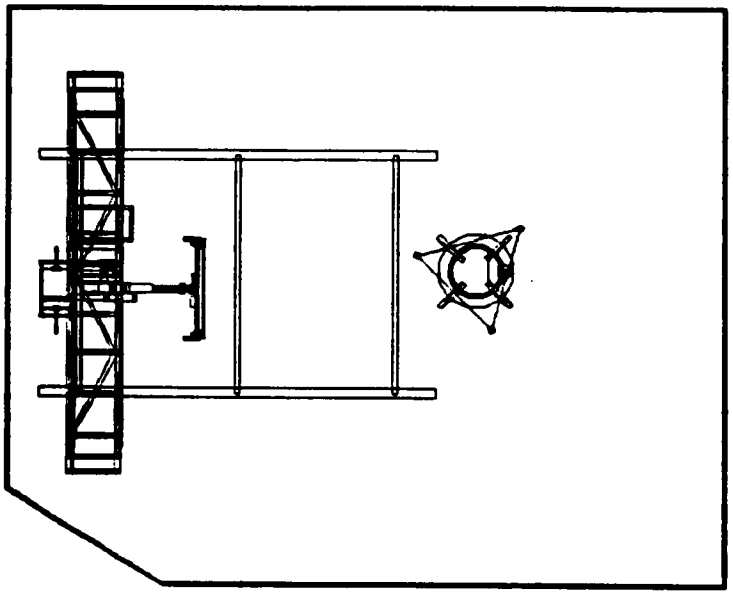


Figure 1: Sample Results for Cooperating 9 DOF Robots

(a) Isometric View



(b) Top View



(c) Side View

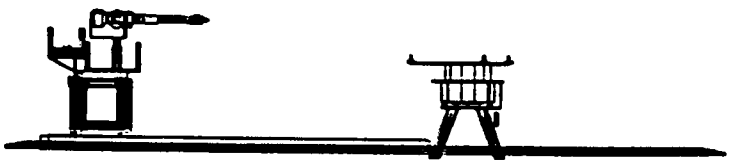


Figure 2: NASA Langley's Automated Structural Assembly Lab

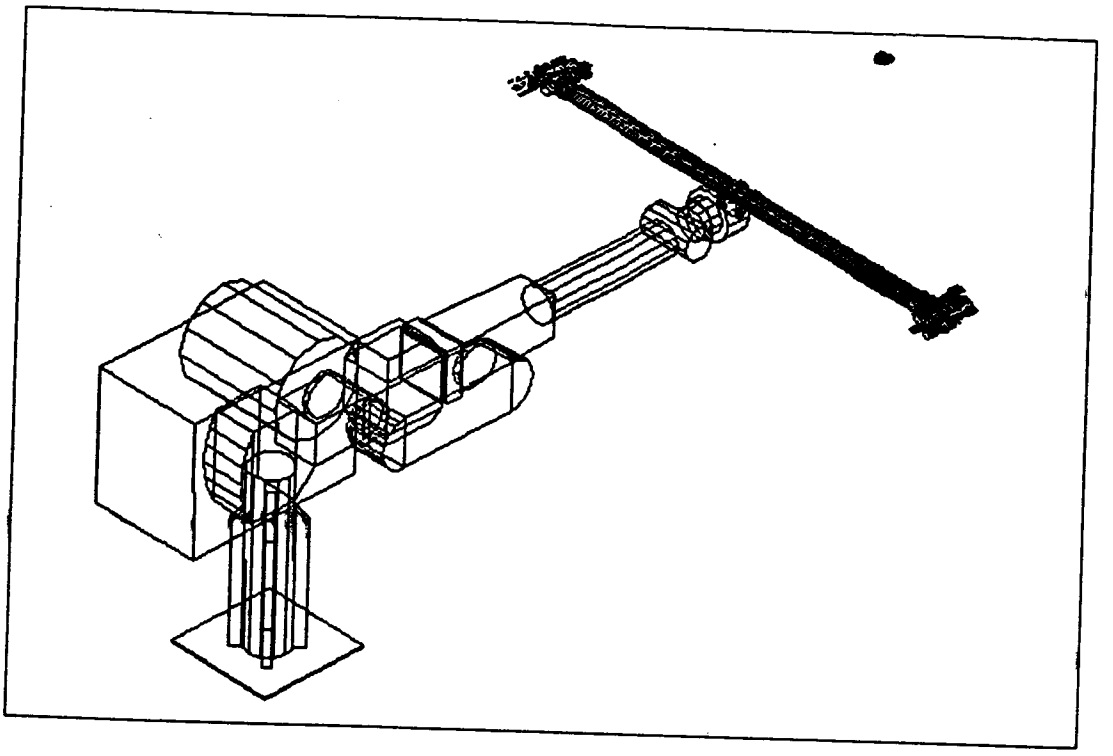


Figure 3: 6 DOF Merlin Robot with End Effector for Truss Assembly

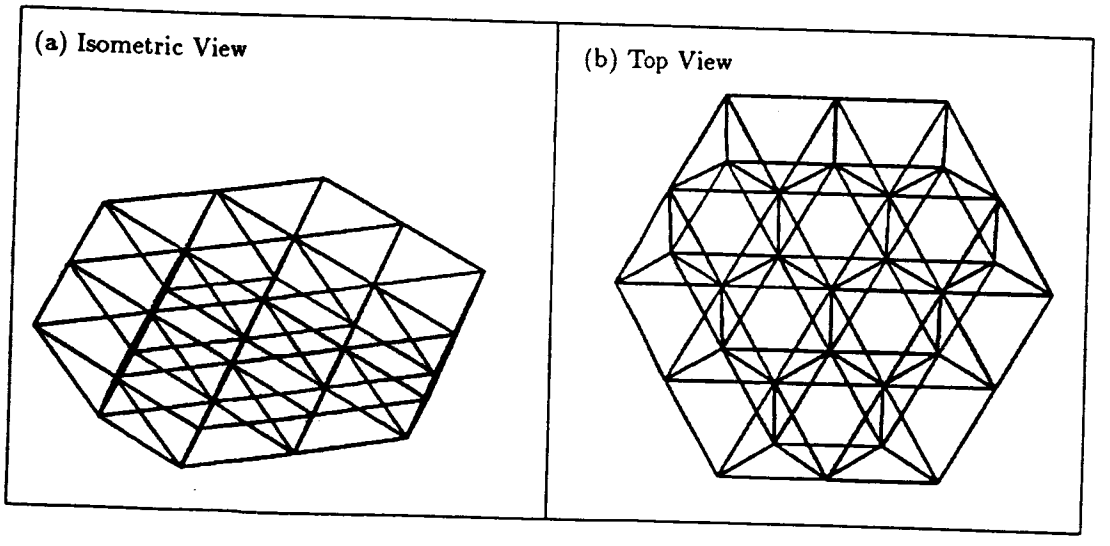


Figure 4: 102 Strut Truss Structure

3 Strategy

Like the single robot planner presented by Dupont [7], the principle strategy of our planner is to minimize the computationally expensive mapping of configuration space by performing mapping on an as required basis. The approach is based around a divide-and-conquer style heuristic for traversing through c-space. Computationally expensive precomputations and exhaustive c-space mappings are avoided. The approach is applicable regardless of the number and type of joints in the robot and for any number of obstacles in the workspace. A *string tightening* algorithm may be applied to modify any safe path found by the planner into a more efficient one, where efficiency is measured by joint space trajectory length.

The path planning method involves first attempting to traverse a c-space vector from the start to the goal of one of the robots. If this vector passes through unsafe space, the hyperspace orthogonal to and bisecting the unsafe segment of the vector is systematically searched to identify an intermediate goal point for consideration as a via point. An attempt is made to traverse from the last safe point to the intermediate goal point. This process is repeated as necessary until the attempted traversal to the newest intermediate goal point is entirely safe. At that point, progression is attempted toward all previous guide points in the opposite order in which they were found, where guide points include not only previous intermediate goal points but also the safe points found on the goal end of each unsafe region which invoked a search. When progression to a particular guide point is not entirely safe, that point is permanently dismissed and progression is attempted toward the next guide point in the specified sequence. The progression continues until an attempt has been made to progress to the global goal point. If that attempted progression is not entirely successful the overall process is repeated until the global goal point has been safely traversed to.

In 2D, the hyperspace orthogonal to an unsafe vector (the space which the heuristic searches) is simply a line. For 2D problems, the initial search is performed equally in both directions until a safe point is found. Subsequent searches will first exhaustively search in the direction which has a component in the previously successful search direction. Only when no safe point can be found in that direction will the other direction be considered. A 2D example of the c-space traversal heuristic is shown in Figure 5. This example involves non-disjoint safe space and requires multiple searches. More 2D examples and a vector description of the heuristic may be found in [6].

In the general nD case, the search space will be $n-1$ dimensional. In this case, several approaches were considered for computing search directions. The most effective method found involves considering all combinations of ± 1 and 0 (except all zeros) times a set of orthogonal basis vectors for the subspace. This yields $3^{n-1}-1$ search directions for an n dof problem. The following vectors may be calculated in the sequence shown and then normalized to yield one such orthogonal basis:

$$\begin{aligned}
\mathbf{B}_1 &= (1, h_1, 0, \ldots, 0) \\
\mathbf{B}_2 &= (b_{1_1}, p_2, h_2, 0 \ldots, 0) \\
\mathbf{B}_3 &= (b_{2_1}, b_{2_2}, p_3, h_3, 0 \ldots, 0) \\
&\vdots \qquad\quad \vdots \qquad\qquad \vdots \\
\mathbf{B}_{n-1} &= (b_{n-2_1}, b_{n-2_2}, \ldots, b_{n-2_{n-2}}, \\
&\qquad p_{n-1}, h_{n-1})
\end{aligned} \tag{1}$$

where the p_i are chosen so that the $\mathbf{B}_i$ and $\mathbf{B}_{i-1}$ are orthogonal, then the h_i are chosen so that the $\mathbf{B}_i$ lie in the search hyperplane.

Initial searches favor all directions equally, whereas subsequent searches sort the i directions $\mathbf{S}_i$ into g equal breadth bins by the following rule:

$$\mathbf{S}_i \in bin(j) \;\; if \;\; \frac{j-1}{g} \leq \frac{dp_i - dp_{min}}{dpmax - dp_{min}} \leq \frac{j}{g} \tag{2}$$

where dp_i is the dot product of $\mathbf{S}_i$ with the previously successful search direction, and dp_{min} and $dpmax$ are the minimum and maximum dp_i, respectively.

Searches then exhaust $bin(i)$ before considering $bin(i+1)$.

3.1 Completeness

Unfortunately, this path planning method is not complete, i.e., it cannot guarantee finding a solution even if one exists. Though certainly undesirable, this lack of completeness does not seem unreasonable since researchers have thus far been unable to develop algorithms which achieve both completeness and practicality for reasonably difficult yet practical path planning problems for more than a few degrees of freedom. We sacrificed completeness in exchange for the possibility of solving some practical yet potentially difficult problems as quickly as possible.

3.2 String Tightening

Once a safe path is found, it may be modified to reduce the joint space trajectory length of the path. This process is referred to as string tightening [7]. Since the path planner produces discretized paths, the objective during string tightening is to reduce the following cost function:

$$L_1^N = \sum_{i=1}^{N-1} \sqrt{\sum_{j=1}^{n} (\theta_j(i+1) - \theta_j(i))^2} \tag{3}$$

where:

L_1^N = the joint space trajectory length
N = number of knot points in path
n = number of dof
$\theta_j(i)$ = i^{th} knot point for joint j

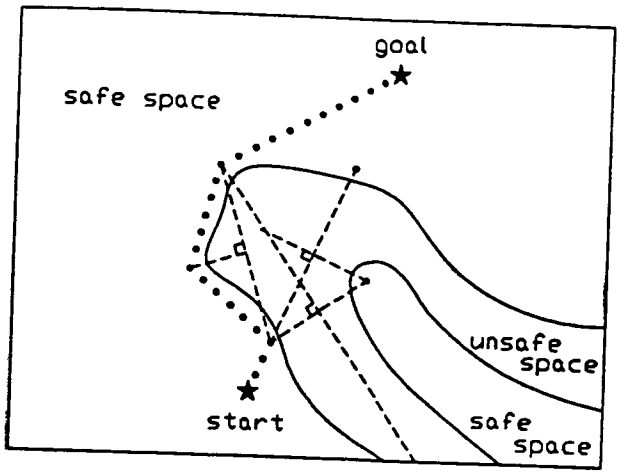


Figure 5: 2D Example of C-Space Traversal Heuristic

The tightening algorithm involves examining each sequence of three adjacent knot points and performing whichever of the two options below produces the most desirable effect on L_1^N:

1. Make no changes to the knot points.

2. Modify the second knot point so that the three knot points are straight in the robot's joint space (if not already so).

The feasibility of option 2 must be determined by checking the configuration for interference. These local adjustments are continued along the length of the path until no significant improvement can be obtained from further adjustments.

4 Implementation

In addition to having been implemented for single and cooperating robot path planning problems for the CIRSSE testbed, the path planning strategy described in this paper has been implemented for the ASAL path planning problem described in Section 2. The programs are written in C and utilize sections of code developed by Schima [8]. They also invoke methods and code developed by Hamlin and Kelley [9, 10]. The polytope representation scheme was chosen because it permits accurate modeling of the robots and typical obstacles in the workcell while enabling relatively fast interference checking. Paths are visually simulated using CimStation [11]. The implementation uses 242 search directions and 5 bins for search direction prioritization.

Since this was a preliminary implementation intended to evaluate the possible usefulness of the path planner for the ASAL path planning problem, some simplifications were made:

- Nodes were not modeled.
- In the ASAL, panels are installed (the first set after the 60^{th} strut). These panels were not modeled (except for one particular strut as a case study).

5 Results

The path planner quickly found paths for the first 21 struts since there is little possible interference at that stage. Due to symmetry, the assembly of the remaining 81 struts can be accomplished using only 21 unique trajectories for the Merlin with the appropriate carriage and turntable positions for each strut. The path planner was able to find feasible paths for all 102 struts with solution times ranging from 1 to 30 minutes on a SparcStation 1, with the vast majority of solution times in the 2 to 5 minute range.

The 61^{st} strut is possibly the most difficult from a path planning perspective due to the confined location of the goal position and due to the presence of an installed panel above the goal position. Although this implementation generally ignored the panels, a panel was modeled as an obstacle for this strut. In spite of the panel, a path was found without requiring any intermediate carriage/turntable positions. The path found for this strut is illustrated in Figure 6.

Some particular comments regarding this implementation follow:

- The path planner has no trouble with goal positions placing the load or robot in very close proximity to obstacles.
- The path planner performs well even with a large number of obstacles. For example, the final few struts of the assembly involve over 100 workspace obstacles. The additional collision checks required near the end of the assembly seem to increase execution time by a factor of approximately two.

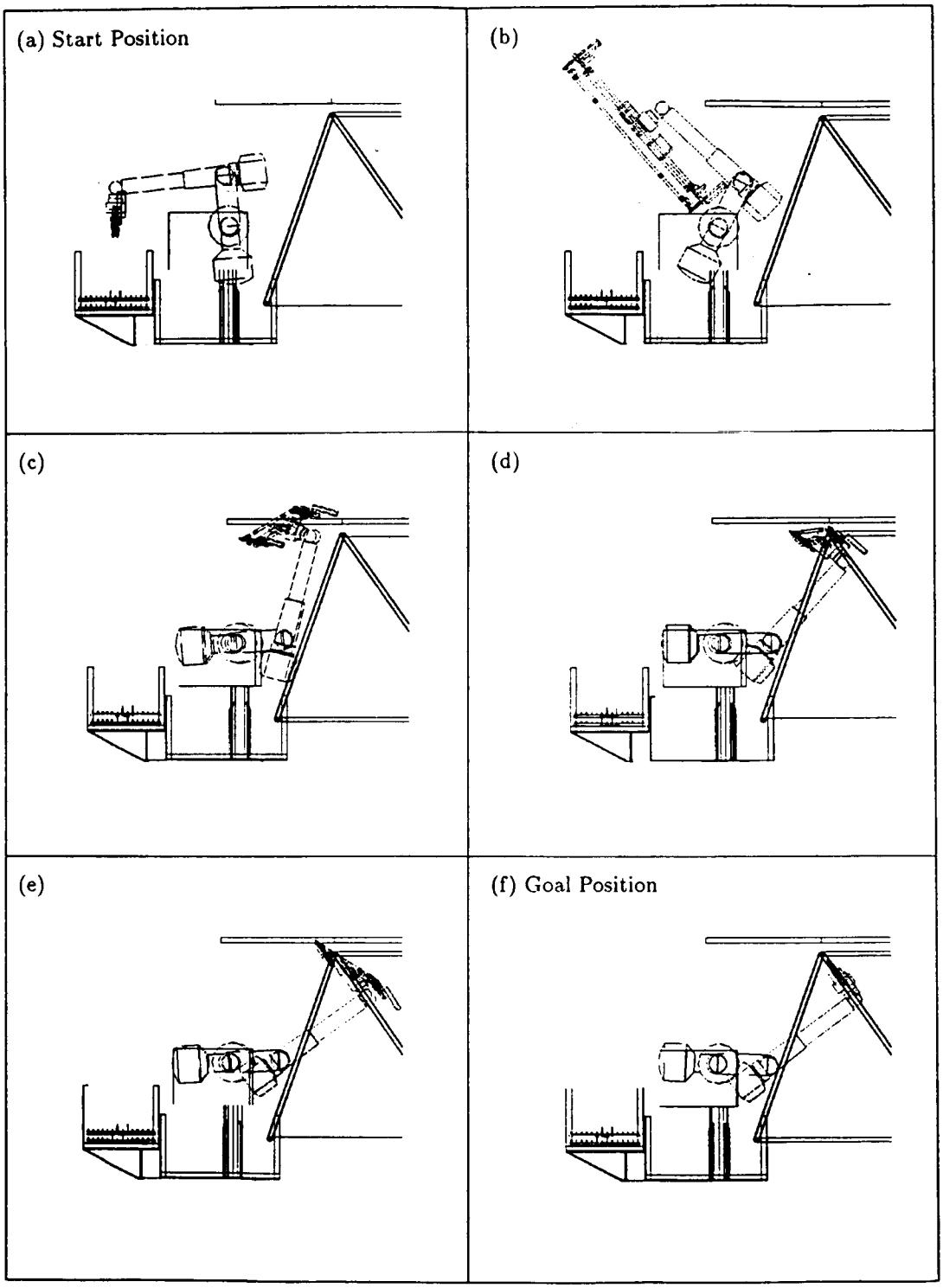


Figure 6: Sample Results for 61^{st} Strut (Side Views)

- The paths found typically include segments which are obstacle boundary tracing. Because of the close tolerances involved, it is not practical to simply model the objects larger than actual size to provide a safety margin since doing so would often result in an unsolvable problem.

- Panels and nodes were not modeled. As a result, some of the paths might collide with the panels or nodes if the paths were used in an actual assembly. This could be remedied simply by modeling the panels and nodes and including them in the collision checking routine. Due to the small size of the nodes it is expected that including them would have little impact on the difficulty of the path planning problems. Although the panels will typically represent significant obstacles to be avoided, a strut for which the panels would seem to interfere the most was solved with the relevant panel modeled.

- In a few cases the path planner was not able to solve the problem quickly in the forward direction but could quickly solve the problem in the opposite direction. Although a very confined goal position makes it likely that solving in reverse may prove easier, trial and error was the only sure way to decide which direction would yield better performance.

- Return paths for the robot after inserting a strut were not planned.

6 Conclusions and Future Work

This implementation of the path planner for the ASAL assembly task illustrates the potential usefulness of the path planning technique developed at CIRSSE for solving the practical and potentially very difficult ASAL path planning problem. Based on the results of this study, additional work appears warranted towards applying this planning technique to the path planning problems at the ASAL. Some particular issues which would need to be addressed before paths created by the planner could be executed on the actual hardware are as follows:

- Nodes and panels need to be modeled.

- An improved string tightening algorithm or an alternate method of path modification is required to provide paths with adequate clearances. This could be done by modifying the current string tightening cost function to include a penalty on clearance or by utilizing the path found by the planner as input to a potential fields based smoothing algorithm.

Acknowledgments

This work was supported by NASA grant NAGW-1333.

References

[1] John H. Reif. Complexity of the mover's problem and generalizations extended abstract. In *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pages 421–427, 1979.

[2] J.T. Schwartz and M. Sharir. On the 'piano movers' problem ii. general techniques for computing topological properties of real algebraic manifolds. *Computer Science Technical Report No. 41*, February 1982. Courant Institute, New York University.

[3] Y.K. Hwang and Narendra Ahuja. Gross motion planning - a survey. *ACM Computing Surveys*, 24(3):219–291, September 1992.

[4] J.M. Weaver and S.J. Derby. A method for planning collision free trajectories for two cooperating robots. AIAA Space Programs and Technologies Conference, 1992. Paper No. AIAA-92-1722.

[5] J.M. Weaver and S.J. Derby. A divide-and-conquer method for planning collisions free paths for cooperating robots. In *Robotics, Spatial Mechanisms, and Mechanical Systems, ASME*, volume DE-45, pages 461–471, 1992.

[6] J.M. Weaver and S.J. Derby. A divide-and-conquer method of path planning for cooperating robots with string tightening. In *Proceedings of the Fourth Annual Conference on Intelligent Robotic Systems for Space Exploration*, pages 30–40, Sponsored by NASA CIRSSE, Troy, NY, 1992.

[7] Pierre E. Dupont. *Planning Collision-Free Paths for Kinematically Redundant Robots by Selectively Mapping Configuration Space.* PhD thesis, Rensselaer Polytechnic Institute, Troy, NY, 1988.

[8] Francis J. Schima. Two arm robot path planning in a static environment using polytopes and string stretching. Master's thesis, Rensselaer Polytechnic Institute, Troy, NY, 1990.

[9] A representation scheme for rapid 3-d collision detection. CIRSSE Document No. 9, 1988.

[10] G.J. Hamlin and R.B. Kelley. Efficient distance calculation using the spherically-extended polytope (s-tope) model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2502–2507, Nice, France, May 1992. Vol. 3.

[11] Cimstation user's manual, cimstation 4.3. Silma Inc., Cupertion, CA, 1992.

Inertial-Space Disturbance Rejection for Space-Based Manipulators

Kevin Holt and Alan A. Desrochers
Rensselaer Polytechnic Institute
Troy, New York

Disturbance Rejection for Space–Based Manipulators

K. Holt
Lockheed Engineering & Sciences Company
2400 NASA Road 1
M/S C33
Houston, TX 77058

A.A. Desrochers
Electrical, Computer, & Systems
Engineering Department
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

ABSTRACT

This paper describes the implementation of a disturbance rejection controller for a 6–DOF PUMA manipulator mounted on a 3–DOF platform. A control algorithm is designed to track the desired position and attitude of the end–effector in inertial space, subject to unknown disturbances in the platform axes. Experimental results are presented for step, sinusoidal, and random disturbances in the platform rotational axis and in the neighborhood of kinematic singularities.

Robotic manipulators have been proposed as a means of reducing the amount of extra vehicular activity time required for space station assembly and maintenance. The proposed scenario involves a robotic manipulator attached to some mobile platform, such as a spacecraft, satellite, or the space station itself.

Disturbances in the platform position and attitude may prevent the manipulator from successfully completing the task. This work explores the possibility of using the *manipulator* to compensate for platform disturbances.

The problem of controlling a robotic manipulator on a mobile platform has received considerable attention in the past few years. Joshi and Desrochers [1] designed a nonlinear feedback control law to carry out tasks (with respect to the robot base frame) in the presence of roll, pitch and yaw disturbances in the platform axes. Dubowsky, Vance, and Torres [2] proposed a time-optimal planning algorithm for a robotic manipulator mounted on a spacecraft, subject to saturation limits in the attitude control reaction jets. Papadopoulos and Dubowsky [3] developed a general framework for analyzing the control of free-floating space manipulator systems. Most recently, Torres and Dubowsky [4] have presented a technique called the enhanced disturbance map to find manipulator trajectories that reduce the effect of disturbances in the spacecraft position and attitude.

One common assumption in the literature is that the disturbance signal is exactly known. If this is the case, then the end-effector location can be calculated without relying on direct end-point sensing. However, this assumption is invalid if there is a significant delay in the platform position and attitude measurements, or if the kinematics of the platform are not well known, or if the platform is a non-rigid structure. In the more likely case that only the *nominal* platform location and *upper bound* on the disturbance signal are known, direct end-point sensing is needed to measure the end-effector location.

1 The Jacobian and Singularities

The inverse Jacobian is ill-defined for certain manipulator configurations. This section presents an alternative mapping, called the approximate pseudoinverse Jacobian, which is defined for all manipulator configurations.

The Jacobian maps differential changes in joint position to differential changes in Cartesian position and orientation according to the following relationship:

$$du = J(q)dq \tag{1}$$

where $du \in \Re^6$ is the differential Cartesian displacement vector (linear and angular), $q \in \Re^n$ is the vector of joint positions, $dq \in \Re^n$ is the vector of differential joint displacements, and $J \in \Re^{6\times n}$ is the Jacobian matrix.

For the PUMA, the Jacobian matrix is simplest when expressed in frame 6:

$$ {}^6J_{3,9} = \begin{bmatrix} --(d_5+d_6)C_{56} & d_7+a_5S_6 & d_7 \\ (d_5+d_6)S_{56} & a_6+a_5C_6 & a_6 \\ a_5C_5+a_6C_{56}+d_7S_{56} & 0 & 0 \\ --S_{56} & 0 & 0 \\ --C_{56} & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{2} $$

$$ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -S_7 & C_7S_8 \\ --1 & 0 & -C_8 \\ 0 & C_7 & S_7S_8 \end{bmatrix} $$

The following compact notation will be used to denote the matrix ${}^6J_{3,9}$:

$$ {}^6J_{3,9} = \begin{bmatrix} B & 0 \\ D & E \end{bmatrix} \tag{3} $$

where B, D, and E are 3×3 submatrices of the Jacobian.

The inverse Jacobian, when it exists, can also be written in block matrix form

$$ {}^6J_{3,9}^{-1} = \begin{bmatrix} B & 0 \\ D & E \end{bmatrix}^{-1} = \begin{bmatrix} B^{-1} & 0 \\ --E^{-1}DB^{-1} & E^{-1} \end{bmatrix} \quad (4) $$

The PUMA has three singularities. The first is referred to as the *Arm Fully Stretched singularity*. This singularity occurs whenever the arm switches between the **flex** and the **noflex** configurations.

The second singularity corresponds to the *Hand Over Head singularity*. The Hand Over Head configuration corresponds to changing between the **right** and **left** configurations.

The third singularity is the *Wrist singularity*, and occurs when the arm switches between the **flip** and **noflip** configurations.

2 Approximate Pseudoinverse Jacobian

The usual method of dealing with singularities of the Jacobian is to avoid them. This approach is not applicable to the disturbance rejection problem since a sufficiently large disturbance could force the manipulator into a singular configuration. Also, the manipulator must avoid not just singular *points*, but singular *regions*, since the norm of J^{-1} becomes very large in the neighborhood of a singularity.

The pseudoinverse Jacobian is often used to overcome the difficulty of J being a nonsquare matrix, and is defined as

$$ J^{\dagger} \triangleq \begin{cases} J^T(JJ^T)^{-1} & m \leq n \\ J^{-1} & m = n \\ (J^TJ)^{-1}J^T & m \geq n \end{cases} \quad (5) $$

Clearly, this method of computing $J^{\dagger}$ does not address the issue of singularities since it still relies on matrix inversion. A more general approach to computing the pseudoinverse uses *singular value decomposition*. This has one serious drawback, which is the high cost of computing the singular value decomposition (SVD). The SVD algorithm uses a series of Householder transformations to reduce the input matrix to diagonal form. Since this is an $\mathcal{O}(N^3)$ operation, finding the SVD for the 6×6 Jacobian matrix can be too costly to implement in real-time. The alternative presented in this section is called the *approximate pseudoinverse Jacobian*, and is denoted by $J^{\ddagger}$.

The basic idea behind the approximate pseudoinverse is to use the partitioned form of J and perform the SVD on the submatrices B and E. This reduces the number of computations by a factor of four, since two 3×3 singular value decompositions is an $\mathcal{O}(2(N/2)^3)$ operation.

The definition of the approximate pseudoinverse Jacobian is

$$ J^{\ddagger} \triangleq \begin{bmatrix} B^{\dagger} & 0 \\ --E^{\dagger}DB^{\dagger} & E^{\dagger} \end{bmatrix} \quad (6) $$

where B, D, and E are defined as in (3).

Several properties of the approximate pseudoinverse are stated below.

1. $J^{\ddagger} = J^{-1}$ when J is nonsingular.
2. $J^{\ddagger}$ does *not* satisfy the Moore-Penrose conditions when J is singular.
3. Let $dp, d\phi \in \Re^3$ be the linear and angular components of du, respectively, and let $dq_1, dq_2 \in \Re^3$ be the components of dq. Then, the approximate pseudoinverse solution is

$$ \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix} = \begin{bmatrix} B^{\dagger} & 0 \\ --E^{\dagger}DB^{\dagger} & E^{\dagger} \end{bmatrix} \begin{bmatrix} dp \\ d\phi \end{bmatrix} \quad (7) $$

If J is singular, the approximate pseudoinverse finds the minimum norm solution *as if dp and $d\phi$ were decoupled*; that is, $dq = J^{\ddagger}du$ minimizes $\|Bdq_1 - dp\|_2$ and $\|Edq_2 - d\phi\|_2$.

3 Behavior Near Singularities

Figure 1 compares the 2-norm, or the maximum singular value, of $J^{\dagger}$ (solid curve), $J^{\ddagger}$ (dashed curve), and J^{-1} (dotted curve) in the vicinity of the Hand Over Head singularity.

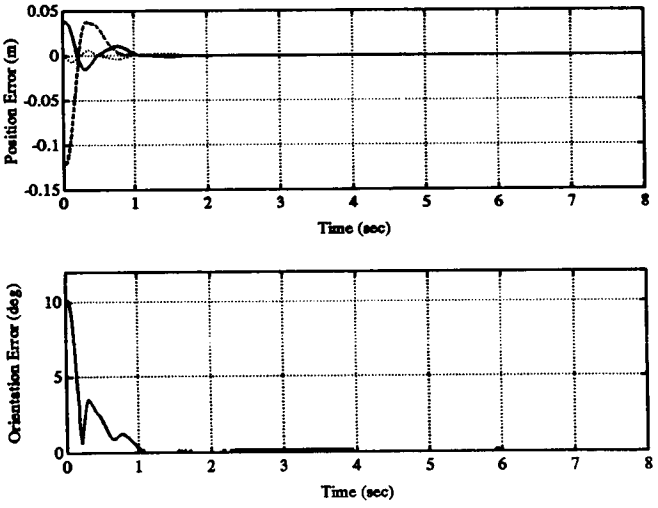


Figure 1: 2-Norms of $J^{\dagger}$ (solid curve), $J^{\ddagger}$ (dashed curve), and J^{-1} (dotted curve) Near Hand Over Head Singularity

The discontinuities in $\|J^{\dagger}\|_2$ and $\|J^{\ddagger}\|_2$ occur when the smallest nonzero singular value, σ_r, falls below the threshold value, σ_{min}. Setting σ_{min} to a relatively small value will shrink the width of the "well" about the singular point, thus extending the range over which $J^{\dagger} = J^{-1}$ and $J^{\ddagger} = J^{-1}$. The side-effect is that the norm will be very large and highly discontinuous near the singularity. By the same token, setting σ_{min} to a relatively large value will reduce the discontinuity in the norm by increasing the width of the singular region. A threshold value of $\sigma_{min} = 0.1$ was used to generate Figure 1.

4 Bound on Approximation Error

The pseudoinverse and approximate pseudoinverse Jacobians are identical only when J is nonsingular. In order to characterize the difference in behavior at a singularity, some measure of the *approximation error* is needed. A reasonable way to measure the approximation error is to see "how close" $J^{\ddagger}$ is to being a true generalized inverse using the following norm:

$$\|JJ^{\ddagger}J - J\|_2$$

Consider the matrix

$$JJ^{\ddagger}J = \begin{bmatrix} BB^{\dagger}B & 0 \\ D-(I-EE^{\dagger})D(I-B^{\dagger}B) & EE^{\dagger}E \end{bmatrix} \quad (8)$$

Subtracting J yields

$$JJ^{\ddagger}J - J = \begin{bmatrix} I & 0 \\ 0 & I-EE^{\dagger} \end{bmatrix} \begin{bmatrix} B & 0 \\ D & E \end{bmatrix} \begin{bmatrix} I-B^{\dagger}B & 0 \\ 0 & I \end{bmatrix} \quad (9)$$

When both B and E are singular, the approximation error is bounded as follows:

$$\begin{aligned} &\|JJ^{\ddagger}J - J\|_2 = \\ &\left\| \begin{bmatrix} I & 0 \\ 0 & I-EE^{\dagger} \end{bmatrix} \begin{bmatrix} B & 0 \\ D & E \end{bmatrix} \begin{bmatrix} I-B^{\dagger}B & 0 \\ 0 & I \end{bmatrix} \right\|_2 \quad (10) \\ &\leq \|J\|_2 \end{aligned}$$

If B is nonsingular, a less conservative upper bound can be found:

$$\begin{aligned} \|JJ^{\ddagger}J - J\|_2 &= \left\| \begin{bmatrix} I & 0 \\ 0 & I-EE^{\dagger} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} \right\|_2 \\ &\leq \|E\|_2 \quad (11) \end{aligned}$$

Likewise, when E is nonsingular the upper bound reduces to

$$\begin{aligned} \|JJ^{\ddagger}J - J\|_2 &= \left\| \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I-B^{\dagger}B & 0 \\ 0 & I \end{bmatrix} \right\|_2 \\ &\leq \|B\|_2 \quad (12) \end{aligned}$$

Finally, if both B and E are nonsingular, the approximate pseudoinverse is identical to the pseudoinverse:

$$\|JJ^{\ddagger}J - J\|_2 = 0 \quad (13)$$

Coordinate Frame k	Computation Time		
	${}^{k}J^{-1}_{3,E}$	${}^{k}J^{\dagger}_{3,E}$	${}^{k}J^{\ddagger}_{3,E}$
0	1.31 ms	25.31 ms	6.38 ms
1	1.31 ms	25.31 ms	6.38 ms
2	1.19 ms	25.31 ms	6.25 ms
3	1.09 ms	24.98 ms	6.11 ms
4	0.97 ms	24.65 ms	5.98 ms
5	0.97 ms	24.65 ms	5.98 ms
6	0.88 ms	24.98 ms	5.98 ms
7	0.82 ms	24.98 ms	5.84 ms
8	0.81 ms	24.98 ms	5.85 ms
9	0.81 ms	24.65 ms	5.85 ms
E	0.95 ms	25.31 ms	6.11 ms

Table 1: Computation Times for ${}^{k}J^{-1}_{3,E}$, ${}^{k}J^{\dagger}_{3,E}$, and ${}^{k}J^{\ddagger}_{3,E}$

5 Computation Time

Table 1 compares the computation times of the the inverse, pseudoinverse, and approximate pseudoinverse Jacobians for each coordinate frame. As predicted, the approximate pseudoinverse is about four times faster to compute than the pseudoinverse.

The inverse, pseudoinverse, and approximate pseudoinverse Jacobian solutions were implemented in the C programming language using the GNU[1] gcc Version 2.2.2 compiler. The data in Table 1 was collected by timing the software on a Motorola MVME 147SA-2 Single Board Computer.

6 A Kinematic Control Law for Disturbance Rejection

Consider a 6-DOF PUMA manipulator mounted on a 3-DOF platform. The goal is to maintain the desired position and attitude of the end-effector with respect to the inertial reference frame (frame 0), subject to arbitrary disturbances in the platform axes. The following information is assumed to be available:

1. $\theta \in \Re^6$, the PUMA joint positions
2. $\eta_o \in \Re^3$, the nominal platform joint positions
3. $\bar{\delta} \in \Re^3$, the maximum deviations from the nominal platform joint positions
4. ${}^{0}u_{0,E} \in \Re^6$, the inertial end-effector location

Two factors contribute to the motion of the end-effector: the differential displacement of the PUMA joints, which can be measured, and the differential displacement of the platform joints, which is unknown. Let δ denote the disturbance signal and let dv be the component of the end-effector motion caused by the

[1]Copyright (C) 1989, 1991 Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA.

differential displacement of the platform joints. Then, the differential end-effector displacement can be written as

$$\begin{aligned} {}^0du_{0,E} &= {}^0J_{3,E}(\eta_o+\delta,\theta)d\theta + dv \\ &= {}^0_3\mathbf{R}(\eta_o+\delta)\,{}^3J_{3,E}(\theta)d\theta + dv \end{aligned} \quad (14)$$

Note that coordinate frame transformations have been applied to isolate the dependence of the PUMA Jacobian on the platform joint positions.

A discrete-time model of the system will now be derived by approximating the differential quantities in (14) with displacements. The underlying assumption here is that the sampling period, ΔT, is sufficiently small (i.e., the sampling rate is much higher than the bandwidth of the system).

Define Δu_k as $\Delta u_k \triangleq u_k - u_{k-1}$, where the subscript k denotes the kth sample step. In the limit as ΔT goes to zero, the displacement Δu_k equals the differential du:

$$\lim_{\Delta T \to 0} \Delta u_k = du \quad (15)$$

Similarly, $\Delta\theta_k \to d\theta$ and $\Delta v_k \to dv$ as $\Delta T \to 0$. Therefore, the discrete-time approximation is

$$\begin{aligned} du &\approx \Delta u_k = u_k - u_{k-1} \\ d\theta &\approx \Delta\theta_k = \theta_k - \theta_{k-1} \\ dv &\approx \Delta v_k = v_k - v_{k-1} \end{aligned} \quad (16)$$

and the discrete version of (14) is

$${}^0u_k - {}^0u_{k-1} = {}^0_3\mathbf{R}(\eta_o+\delta_k)\,{}^3J_{3,E}(\theta_k)\Delta\theta_k + \Delta v_k \quad (17)$$

where the subscripts denoting the reference and velocity frames of du have been dropped to avoid confusion with the time index.

Let ${}^0u_d$ be the desired position and orientation of the end-effector along some specified trajectory. The control objective is to drive the end-effector to this position and orientation:

$${}^0u_k \to {}^0u_d \quad \text{as} \quad k \to \infty \quad (18)$$

Ideally, the control objective could be achieved in minimum time by computing the PUMA joint displacements $\Delta\theta_d$ needed to cancel out the inertial-space error. However, exact cancellation would require complete knowledge of the disturbance signal. The next best solution then is to compute a $\Delta\theta_d$ which *approximately* cancels out the inertial-space error. With this goal in mind, the proposed control law is

$$\Delta\theta_d = {}^3J^{\dagger}_{3,E}(\theta_k)\,{}^3_0\mathbf{R}(\eta_o)K_c({}^0u_d - {}^0u_k) \quad (19)$$

where $K_c \in \Re^{6\times 6}$ is a matrix of control gains. Equation (19) will be referred to as the $J^{\dagger}$ *control law* in the sequel.

A simple expression for the closed-loop system can be derived by assuming that there is a one period delay in the control actuation:

$$\Delta\theta_{k+1} = \Delta\theta_d \quad (20)$$

$$\begin{aligned} {}^0u_k - {}^0u_{k-1} &= {}^0_3\mathbf{R}(\eta_o+\delta_k)\,{}^3J_{3,E}(\theta_k) \\ &\quad {}^3J^{\dagger}_{3,E}(\theta_{k-1})\,{}^3_0\mathbf{R}(\eta_o)K_c({}^0u_d - {}^0u_{k-1}) + \Delta v_k \end{aligned} \quad (21)$$

In order to simplify this expression, define the quantity

$$M_{k,k-1} \triangleq {}^0_3\mathbf{R}(\eta_o+\delta_k)\,{}^3J_{3,E}(\theta_k)\,{}^3J^{\dagger}_{3,E}(\theta_{k-1})\,{}^3_0\mathbf{R}(\eta_o)K_c \quad (22)$$

Rewriting (21) in terms of $M_{k,k-1}$, it is easy to see that the closed-loop system is linear with time-varying coefficients:

$${}^0u_k = (I - M_{k,k-1})\,{}^0u_{k-1} + M_{k,k-1}\,{}^0u_d + \Delta v_k \quad (23)$$

A block diagram of the closed-loop system is shown in Figure 2.

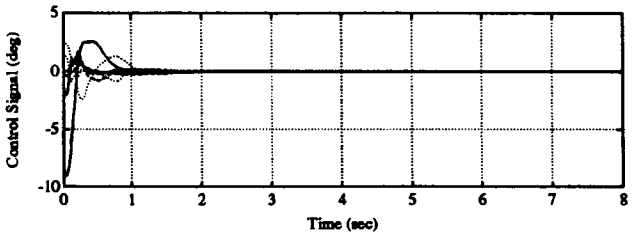


Figure 2: Block Diagram of Closed-Loop System

6.1 Design Parameters

The selection of the control gain is greatly simplified by restricting K_c to be a scalar times the identity matrix:

$$K_c = k_c I, \quad 0 \le k_c \le 2 \quad (24)$$

The parameter k_c controls the spectral radius of M_k. For example, if $k_c = 0.5$, then the eigenvalues of M_k will lie on a circle of radius 0.5 in the Λ-plane (or at zero, if J is singular).

It is straightforward to choose a stable k_c if $\bar{\delta}$ is known *a priori*. (Recall that $\bar{\delta}$ is the vector of maximum deviations in the platform joint positions.) Let $\bar{\lambda}$

denote the spectrum of the matrix ${}^0_3\mathbf{R}(\eta_o + \bar{\delta}){}^3_0\mathbf{R}(\eta_o)$. By invoking the slowly time-varying condition, α can be approximated as follows:

$$\alpha \approx \sup_i \arg(\bar{\lambda}) \tag{25}$$

and k_c is calculated as

$$k_c = \frac{2}{\sqrt{\tan^2 \alpha + 1}} \tag{26}$$

The selection of σ_{min} is essentially a trade-off between tracking accuracy and the norm of the control signal. The selection of σ_{min} should be based on the desired upper bound on the norm of $\Delta\theta_d$, which in turn is dictated by the saturation limits of the joint-level controller.

7 Experimental Results

Three sets of experiments focused on the time response of the closed-loop system for step disturbances in the platform joints, sinusoidal disturbances in the platform joints, and random disturbances in the platform joints. Here we present only the results for the step disturbance.

This section analyzes the time response of the closed-loop system for 10° and 30° step disturbances in the platform rotation. For each case, the control gain K_c was set to identity.

7.1 10° Step Disturbance

Figure 3 shows the inertial-space errors errors when a 10° step disturbance is applied to the platform rotational joint. The linear (X, Y, and Z) components of the error are shown in the upper plot and the orientation error in the lower plot. The components of $\Delta\theta_d$, the control vector, are plotted in Figure 4.

	Maximum Overshoot	4% Settling Time
X	$1.527\times10^{+0}$ cm	1.54 s
Y	$3.825\times10^{+0}$ cm	0.84 s
Z	6.366×10^{-1} cm	1.70 s
ϕ_e	$3.503\times10^{+0}$ deg	1.00 s

Table 2: Maximum Overshoot and 4% Settling Time for 10° Step Disturbance in Platform Rotation

Table 2 lists the maximum overshoot and 4% settling time for the X, Y, Z, and orientation errors. The 4% settling time refers to the time required for the error to enter and remain within $\pm\epsilon$ of zero, where ϵ is 4% of the peak absolute error.

7.2 30° Step Disturbance

The inertial-space errors and control signals for the 30° case are shown in Figures 5 and 6. The maximum overshoot and settling time for each coordinate are displayed in Table 3.

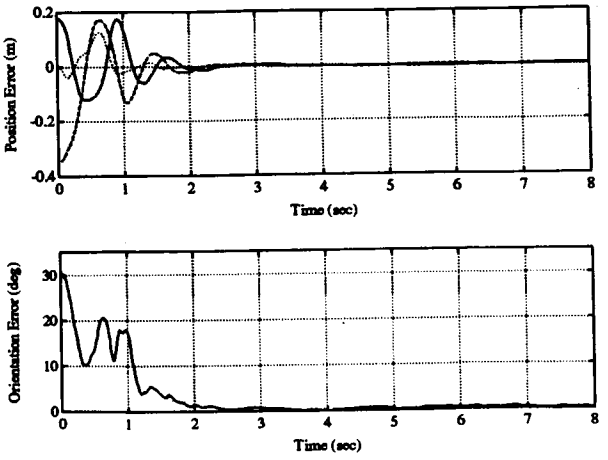


Figure 3: Position Error (X – solid curve; Y – dashed curve; Z – dotted curve) and Orientation Error for 10° Step Disturbance in Platform Rotation

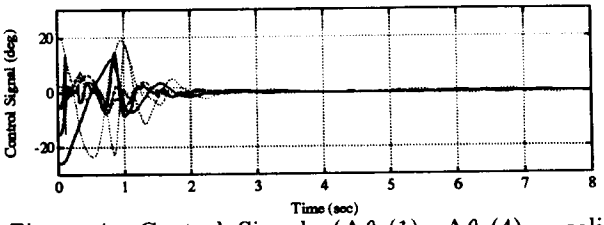


Figure 4: Control Signals ($\Delta\theta_d(1)$, $\Delta\theta_d(4)$ – solid curves; $\Delta\theta_d(2)$, $\Delta\theta_d(5)$ – dashed curves; $\Delta\theta_d(3)$, $\Delta\theta_d(6)$ – dotted curves) for 10° Step Disturbance in Platform Rotation

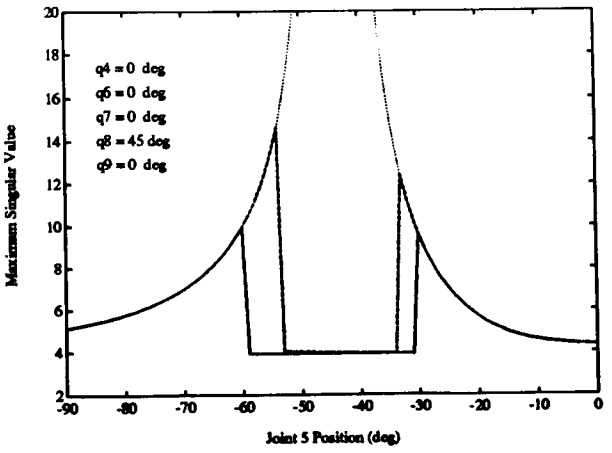


Figure 5: Position Error (X – solid curve; Y – dashed curve; Z – dotted curve) and Orientation Error for 30° Step Disturbance in Platform Rotation

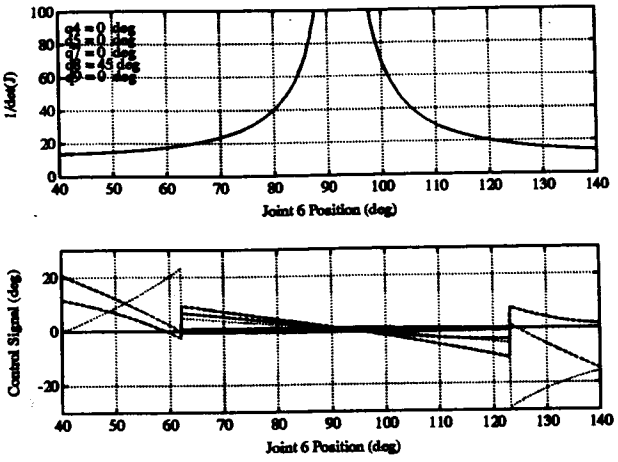


Figure 6: Control Signals ($\Delta\theta_d(1)$, $\Delta\theta_d(4)$ – solid curves; $\Delta\theta_d(2)$, $\Delta\theta_d(5)$ – dashed curves; $\Delta\theta_d(3)$, $\Delta\theta_d(6)$ – dotted curves) for 30° Step Disturbance in Platform Rotation

	Maximum Overshoot	4% Settling Time
X	$1.737\times10^{+1}$ cm	1.97 s
Y	$1.706\times10^{+1}$ cm	2.43 s
Z	$1.253\times10^{+1}$ cm	1.66 s
ϕ_e	$2.055\times10^{+1}$ deg	2.08 s

Table 3: Maximum Overshoot and 4% Settling Time for 30° Step Disturbance in Platform Rotation

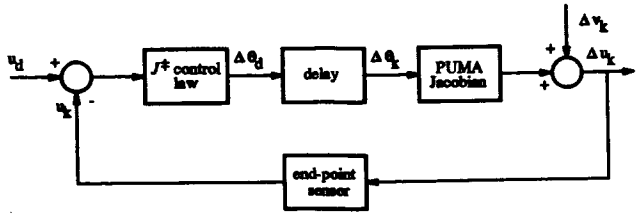


Figure 7: Behavior of $1/\det(J)$ and Open-Loop Control Signals ($\Delta\theta_d(1)$, $\Delta\theta_d(4)$ – solid curves; $\Delta\theta_d(2)$, $\Delta\theta_d(5)$ – dashed curves; $\Delta\theta_d(3)$, $\Delta\theta_d(6)$ – dotted curves) Near Arm Fully Stretched Singularity

8 Behavior Near Singularities

Figure 7 shows the vector of open-loop control signals near the Arm Fully Stretched singularity. The minimum singular value parameter, σ_{min}, was set to 0.1. At this value of σ_{min}, the control in the direction of the workspace boundary becomes very weak approximately 30° from the singular point. This prevents the end-effector from getting too close to the workspace boundary. Consequently, the manipulator will not switch between the `flex` and `noflex` configurations while the $J^\ddagger$ controller is running.

If the parameter σ_{min} is sufficiently small, however, the width of the singular region will be reduced to the point where the control signal for joint 6 ($\Delta\theta_d(3)$) could drive the arm through the singularity. This may lead to an undesirable "chattering" behavior, in which the arm rapidly oscillates between the `flex` and `noflex` configurations.

9 Summary

Several important conclusions can be drawn from the experimental results.

1. The relative stability of the closed-loop system is a function of the amplitude of the disturbance signal.

2. The relative performance of the controller is a function of the frequency of the disturbance signal.

In other words, the $J^\ddagger$ controller is like a high-pass filter; the lowest frequency components of the disturbance signal are attenuated the most.

3. The control in certain directions becomes very weak near singularities.

This implies that there may be an unavoidable tracking error in the "forbidden" directions when the arm is at or near a singularity.

References

[1] J. Joshi and A. A. Desrochers, "Modeling and Control of a Mobile Robot Subject to Disturbances," in *Proc. 1986 IEEE Robotics and Automation Conference*, (San Francisco, CA), 1986.

[2] S. Dubowsky, E. E. Vance, and M. A. Torres, "The Control of Space Manipulators Subject to Spacecraft Attitude Control Saturation Limits," in *Proc. NASA Conf. Space Telerobotics*, (Pasadena, CA), January – February 1989.

[3] E. Papadopoulos and S. Dubowsky, "On the Nature of Control Algorithms for Space Manipulators," in *Proc. 1990 IEEE Robotics and Automation Conference*, (Cincinnati, OH), 1990.

[4] M. A. Torres and S. Dubowsky, "Minimizing Spacecraft Attitude Disturbances in Space Manipulator Systems," *Journal of Guidance, Control, and Dynamics*, vol. 15, pp. 1010 – 1016, July – August 1992.

Direct Model Reference Adaptive Control of Robotic Arms

Howard Kaufman, David C. Swift,
Steven T. Cummings, and Jeffrey R. Shankey
Rensselaer Polytechnic Institute
Troy, New York

Direct Model Reference Adaptive Control of Robotic Arms*

Howard Kaufman, David C. Swift, Steven T. Cummings and Jeffrey R. Sankey
Electrical, Computer and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, NY 12180-3590
Tel: 518-276-6081
Fax: 518-276-6261
email: kaufman@ecse.rpi.edu

Abstract

This paper presents the results of controlling A PUMA 560 Robotic Manipulator and the NASA shuttle Remote Manipulator System (RMS) using a Command Generator Tracker (CGT) based Model Reference Adaptive Controller (DMRAC). Initially, the DMRAC algorithm was run in simulation using a detailed dynamic model of the PUMA 560. The algorithm was tuned on the simulation and then used to control the manipulator using minimum jerk trajectories as the desired reference inputs. The ability to track a trajectory in the presence of load changes was also investigated in the simulation. Satisfactory performance was achieved in both simulation and on the actual robot. The obtained responses showed that the algorithm was robust in the presence of sudden load changes. Because these results indicate that the DMRAC algorithm can indeed be successfully applied to the control of robotic manipulators, additional testing was performed to validate the applicability of DMRAC to simulated dynamics of the shuttle RMS.

1 Introduction

Direct adaptive control offers the potential for uniform control of robotic manipulators in the presence of uncertain flexibilities, changing dynamics due to unknown and varying payloads, and nonlinear joint interactions without explicit parameter identification.

One such direct adaptive algorithm that is especially attractive for robotic control is the direct model reference adaptive controller (DMRAC) discussed in [1-3]. This adaptive algorithm is very appealing for robotic control because of the following features:

- asymptotically zero output error with all states bounded,
- lack of dependence on plant parameter estimates,
- direct applicability to multiple input-multiple output plants,

*This paper is based upon research performed under NASA grant NAGW-1333 and under NSF grant ECS-9111565.

- sufficiency conditions which are independent of plant dimension,
- control calculation which does not require adaptive observers or the need for full state feedback,
- ease of implementation, and
- successful experimental validation.

This procedure has been previously used to control a single link flexible robotic joint and a nonlinear model of a two link Puma [4,5]. In view of the excellent tracking results demonstrated in these papers, it was concluded that this adaptive algorithm should be used to control an actual Puma arm. This effort has consisted of two main thrusts: namely, control of the representative simulation model developed in [6], and the transition of the tuned algorithm to the actual robotic arm.

Because results indicated that the performance of the DMRAC algorithm was robust with respect to representative load variations, additional applicational studies were initiated using the NASA shuttle Remote Manipulator System (RMS).

2 Direct MRAC Development

2.1 Basic algorithm

The linear time invariant model reference adaptive control problem is considered for a plant which is described by

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) \quad (1)$$

$$y_p(t) = C_p x_p(t) \quad (2)$$

where $x_p(t)$ is the $(n \times 1)$ plant state vector, $u_p(t)$ is the $(m \times 1)$ control vector, $y_p(t)$ is the $(q \times 1)$ plant output vector, and A_p, B_p are matrices with appropriate dimensions.

The objective is to find, without explicit knowledge of A_p, and B_p, the control $u_p(t)$ such that the plant output vector $y_p(t)$ approximates "reasonably well" the output of the following (and usually lower order) reference model:

$$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t) \quad (3)$$

$$y_m(t) = C_m x_m(t) \quad (4)$$

The MRAC algorithm is given by [1]:

$$u_p(t) = K_e(t)[y_m(t) - y_p(t)] + K_x(t)x_m(t) + K_u(t)u_m(t) \quad (5)$$

with the gains $K_e(t), K_x(t)$, and $K_u(t)$ being adaptive. The adaptive gains are concatenated into the matrix $K_r(t)$ which is given by

$$K_r(t) = [K_e(t), K_x(t), K_u(t)] \quad (6)$$

and the vector $r(t)$ is defined by

$$r(t) = [y_m(t) - y_p(t), x_m(t), u_m(t)]^T \tag{7}$$

Then

$$u_p(t) = K_r(t)r(t) \tag{8}$$

The adaptive gains are obtained as a combination of an integral gain and a proportional gain as shown below:

$$K_r(t) = K_p(t) + K_I(t) \tag{9}$$

$$K_p(t) = [y_m(t) - y_p(t)]r^T(t)\bar{T} \tag{10}$$

$$\dot{K}_I(t) = [y_m(t) - y_p(t)]r^T(t)T \tag{11}$$

Sufficient conditions for stability derived for a constant model command in [2]. These conditions require that the matrices T and $\bar{T}$ be respectively chosen as positive definite and positive semidefinite, and that the plant be almost strictly positive real (ASPR), that is, for the plant represented by the triple (A_p, B_p, C_p) there exists a matrix K_e (not needed for implementation) such that the fictitious stabilized plant described by the triple $(A_p - B_pK_eC_p, B_p, C_p)$ is positive real. If these sufficient conditions hold, then all states and gains are bounded and the output error vanishes asymptotically.

The adaptive control of plants that are not ASPR is a more difficult problem when utilizing the CGT based MRAC laws. BarKana [3] suggested augmenting the plant with parallel dynamics such that the augmented plant is ASPR in which case the previously described adaptive controller may be utilized. To illustrate this concept, consider the non-ASPR plant described by the transfer matrix

$$G_p(s) = C_p(sI - A_p)^{-1}B_p \tag{12}$$

Then, choose a matrix $H(s)$ such that the augmented plant transfer matrix

$$G_a(s) = H^{-1}(s) + G_p(s) \tag{13}$$

is ASPR. In [3] it is shown that $G_a(s)$ will be ASPR provided that

- $H(s)$ itself is ASPR
- $H(s)$ stabilizes the closed loop output feedback system with transfer function. $[I + G_p(s)H(s)]^{-1}G_p(s)$.

An easily implementable version of $H(s)$ which has had extensive use is

$$H(s) = K(1 + s/so)$$

resulting in:

$$G_a(s) = \frac{D}{1 + s/so} + G_p(s) \tag{14}$$

where $D = K^{-1}$.

Unfortunately, the error which is ensured to be stable is not the true difference between the plant and the model. Rather, the error is now the difference between the outputs of the augmented plant and the model. However, in [3] it is shown that if the plant is output stabilizable via high gain output feedback, then $\| D \|$ may be chosen to be small. Thus, the augmented plant error will be approximately equal to the original plant error.

One procedure for eliminating this output error is to also incorporate the supplementary feedforward of (14) into the reference model output as shown in [2]. To illustrate this, denote the augmented plant output as z_p where

$$\begin{aligned} z_p &= y_p + Hu_p && (15) \\ &= y_p + H[K_x x_m + K_u u_m + K_e e_y] && (16) \end{aligned}$$

and

$$H = D/(1 + s\tau). \tag{17}$$

In a similar manner, define an augmented model output as

$$z_m = y_m + H[K_x x_m + K_u u_m] \tag{18}$$

Now, for adaptive control of the augmented plant, consider the error between the augmented plant and model outputs. Thus,

$$\begin{aligned} (z_m - z_p) &= y_m - y_p - HK_e e_x \\ &= y_m - y_p - HK_e(z_m - z_p) \\ \text{or} \quad z_m - z_p &= (I + HK_e)^{-1}(y_m - y_p) && (19) \end{aligned}$$

Consequently if as in [2] z_p is forced to follow z_m, then y_p will follow y_m.

3 Puma Model Development

In order to test the performance of the Direct Model Reference Adaptive Controller (DMRAC), an accurate non-linear coupled model of the PUMA manipulator was needed. A full explicit dynamic model of the PUMA 560 manipulator, derived by Armstrong, Khatib, and Burdick [6], was selected. The formulation of the PUMA model was computationally efficient using 25% fewer calculations than a 6 degree of freedom recursive Newton-Euler method. The algebraic formulation of the model also allowed for the easy addition of a load by modifying the link 6 mass, center of mass, and inertia parameters.

Figure 1 shows the six rotational joint axis, $\{z_1, \ldots, z_6\}$, for the PUMA 560. Only the rotational, z_1, axis are shown in the figure. Positive rotations follow the right hand rule - counter-clockwise looking down the z axis. The six joint of the PUMA 560 are as follows:

- *Joint 1.* A vertical rotation about the base, z_1.

- *Joint 2.* A horizontal rotation about the shoulder, z_2.
- *Joint 3.* A horizontal rotation about the elbow, z_3.
- *Joint 4.* A twist of the wrist, z_4.
- *Joint 5.* A inclination of the wrist, z_5.
- *Joint 6.* A twist of the mounting flange, z_6.

The position of the manipulator in Figure 1 illustrates the zero position. Note that when Joint 5 is at zero, axis z_4 and z_6 coincide.

The dynamic equations of motion used to model the PUMA are:

$$A(\theta)\ddot{\theta} + B(\theta)[\dot{\theta}\dot{\theta}] + C(\theta)[\dot{\theta}^2] + g(\theta) = \Gamma \tag{20}$$

where

$A(\theta)$ is the 6 x 6 positive definite kinetic energy matrix,

$B(\theta)$ is the 6 x 15 matrix of coriolis torques,

$C(\theta)$ is the 6 x 6 matrix of centrifugal torques,

$g(\theta)$ is the 6 vector of gravity torques,

$\ddot{\theta}$ is the 6 vector of joint accelerations,

$[\dot{\theta}\dot{\theta}]$ is the 15 vector of velocity products, where

$$[\dot{\theta}\dot{\theta}] = [\dot{\theta}_1\dot{\theta}_2, \dot{\theta}_1\dot{\theta}_3, ..., \dot{\theta}_1\dot{\theta}_6, \dot{\theta}_2\dot{\theta}_3, ..., \dot{\theta}_4\dot{\theta}_6, \dot{\theta}_5\dot{\theta}_6]^T$$

$[\dot{\theta}^2]$ is the 6 vector of squared velocities, where

$$[\dot{\theta}^2] = [\dot{\theta}_1^2, \dot{\theta}_2^2, ..., \dot{\theta}_6^2]^T$$

and Γ is the 6 vector of joint torques.

The above model can be cast into state space form by solving Equation (20) for $\ddot{\theta}$ and choosing the following 12 x 1 state vector,

$$x^T = [\theta^T, v^T] \tag{21}$$

where

$$\begin{aligned} \theta &= [\theta_1, ..., \theta_6]^T \\ v &= [\dot{\theta}_1, ..., \dot{\theta}_6]^T \end{aligned}$$

Thus, the state space model is as follows,

$$\begin{aligned} \dot{\theta} &= v \\ \dot{v} &= A^{-1}(\theta)[\Gamma - B(\theta)[\dot{\theta}\dot{\theta}] - C(\theta)[\dot{\theta}^2] - g(\theta)] \end{aligned}$$

The controlled output vector for the plant was

$$y_{plant} = \theta + \alpha v \tag{22}$$

where, α is a diagonal 6 x 6 matrix of velocity weighting factors.

This velocity term is present to help remove high frequency oscillations caused by the controller. The maximum allowable torques (in n-m) were [97.6, 186.4, 89.4, 24.2, 20.1, 21.3]

4 Implementation Issues

4.1 Reference Model

The first decision to be made in implementing the DMRAC algorithm is the choice of reference model order. If one chooses the order too low, then excessively large gains may occur even in a well-tuned controller. This may produce greater than desired accelerations in the robot arm joints resulting in joint torque saturations leading to poor model following. If one chooses the order too high then excessive response delays may be incurred. For the PUMA 560, an independent second order reference model was selected for each of the six joints. This is not unreasonable since in a PUMA 560, as with many manipulators, the mass matrix is approximately diagonal for all θ making the system nearly decoupled.

Thus, for each joint, the reference model transfer function was:

$$y_{m_i}(s)/u_{m_i}(s) = \frac{w_{n_i}^2}{s^2 + 2\zeta_i w_{n_i} s + w_{n_i}^2}$$

where

$$w_{n_i} = 5$$

and

$$\zeta_i = 1 \qquad i = \{1, 2, 3, 4, 5, 6\}.$$

Critical damping was selected so as to reduce the possibility of joint angle overshoots. This conforms to a standard safety feature of robot arm controllers which tends to avoid obstacle collisions. Of course, once the choice of critical damping is made, the choice of natural frequency governs the speed of model response to inputs. A choice of $w_n = 5$ yields a 90% rise time of about 0.8 sec.

4.2 Command Generation

For Testing purposes a minimum jerk trajectory was generated through the following positions at the noted times.

Joint Position (deg)						
1	2	3	4	5	6	Time (sec)
0	-45	180	0	45	90	0
90	-90	90	45	0	45	6
0	0	180	0	90	90	13
0	-45	180	0	45	90	18

The resulting angular paths for each joint were then used as the reference model commands $u_{m_i}(t)$.

4.3 Bias Introduction

For the PUMA 560 manipulator, the origin of the coordinate system should be such that the adaptation gains have a non-zero excitation throughout the range of interest. For example, assume that in order to maintain an output of $y_p = [0, ..., 0]^T$, a non-zero input, u_p, is required. However a zero command to the reference model, $u_m = [0, ..., 0]^T$, will result in a zero model output and a zero state vector. Thus in this case $e_y = y_m - y_p$ will also be zero, and the vector, $r(t)$, defined by (3.10) will be zero resulting in a zero control. Since the plant requires a non-zero control to maintain a zero output, the DMRAC algorithm requires a non-zero error signal in order to apply a non-zero control which will result in a steady-state error at the zero output position.

If the reference model coordinates are shifted by a constant bias term, then a zero command to the reference model, $u_m = [0, ..., 0]^T$ will produce non-zero outputs for the model state and output vectors which, in turn, will produce a non-zero command to the plant. This bias term is subtracted from the model command, u_m, and the plant output, y_p, as follows,

$$u_m(t) = \hat{u}_m(t) - q_{bias} \quad (23)$$

$$y_p(t) = \hat{y}_p(t) - q_{bias} \quad (24)$$

where $\hat{u}_m(t)$ is the original model command in the original coordinate system, $u_m(t)$ is the new biased model command to be applied to the model dynamics, $\hat{y}_p(t)$ is the actual plant output, $y_p(t)$ is the new biased plant output to be used to form the error signal, and q_{bias} is a constant bias term. For robotic manipulators, q_{bias} has units of radians and should be selected such that a new plant output of $y_p = [0, ..., 0]$ correspond to an equilibrium position. By examining the zero position of the robot, Figure 1, it is clear that $y_p = \{0, 0, 0, 0, 0, 0\}$ is not an equilibrium. However bias of,

$$q_{bias} = \{0, 90, 90, 0, 0, 0\} \text{ degrees} \quad (25)$$

will shift the zero position to that shown in Figure 2.

4.4 Feedforward Design

The feed-forward filter dynamics for joint i is given as,

$$D_i(s) = \frac{K_{d_i}}{1 + \tau_i s} \tag{26}$$

where K_{d_i} is the DC gain and τ_i is the time constant.

5 PUMA Simulation Results

In this section, we briefly discuss the tuning process and present plots of a simultaneous, six joint response of the PUMA 560 under DMRAC control.

5.1 Tuning

Once the reference model has been chosen, one must choose values for the various DMRAC parameters. Specifically these are

$\bar{T} \equiv$ proportional gain weighting matrix, eq. (10)

$T \equiv$ integral gain weighting matrix, eq. (11)

$D \equiv$ plant/model feedforward gain, eq. (17)

$\tau \equiv$ plant/model feedforward time constant, eq. (17)

$\underline{\alpha} \equiv 6$ vector of plant rate feedforward gains, eq. (22)

For the fully centralized DMRAC algorithm with the plant derivative output term and the supplementary feed-forward in the reference model and plant, there are 1182 parameters to be selected as shown in Table 5. At first, this number seems very intimidating, but as will be shown, the number of tuning parameters can be greatly reduced by some simplifications and by adjusting the parameters in groups rather than individually.

The most drastic reduction in the number of tuning parameters can be achieved by forcing the integral and proportional adaptation weighting matrices, T and $\bar{T}$ to be diagonal. This reduces the number of tuning parameters from 1182 to 78.

The reference model dynamics have 12 tuning parameters, six w_m's and six ζ_i's.

It is customary in robotic applications to tune controllers for critical damping so that there is no overshoot. Overshoot may cause a robot end effector to penetrate the surface of the work environment. The undamped natural frequency terms, w_{n_i} are chosen such that the reference model will have a specified step response. Typically, the reference model dynamics are chosen such that they are "reasonable"for the plant to follow since the DMRAC algorithm will try to force the plant to follow the model output. For the case

of a PUMA 560 Manipulator, all of the w_{n_i} were initially set to 5.0. The model's dynamic parameters can be changed as needed if the robot is having problems tracking the model.

Initially, the plant output derivative weights, α, are set to zero. These weights are used to remove high frequency components from the plant control signal, u_m, and should only be used when needed as they will affect the transient response.

The feed-forward filter has 12 tuning parameters, six gains K_{d_i} and six time constants τ_i. A good first choice for the τ_i is approximately one-tenth the model time constant.

Initially the value of τ_i were all set to 0.1 s, and the six DC filter gains K_{d_i} were set to 1.0. Increasing the filter gain was seen to typically improve the tracking performance.

The diagonal components of $\bar{T}$ and T were initially, all set to unity. A reasonable method of tuning a DMRAC controller is to start the plant at an equilibrium position and apply small step inputs. After a reasonable performance is achieved with the step inputs, the DMRAC should be fine tuned using typical plant trajectories.

If the closed loop system is very sensitive to initial conditions, start with small steps as described above, let the system reach steady-state, and then save all of the DMRAC controller state information (integral adaptation matrix, K_I; reference model state vector, x_m; and the filter state vector) to be used as initial conditions for the next run. This will significantly cut down the adaptation time required for the gains to reach their steady-state values.

In order to compare the tuning results, some criterion must be established. For this excessive, the goal was to kept the peak model following errors small and to keep the error trajectory as close to zero as possible. Small errors were tolerable during motion. It was also desired to achieve zero error in steady-state.

The step response with the initial tuning values was sluggish for Joints 1, 4, 5, and 6 with overshoot and oscillations. Joints 2 and 3 settled into their steay-state values quickly but with very large steady-state errors. The process used to complete the tuning was as follows:

1. Refine the tuning for a 10 degree step from the equilibrium position.
2. Using the refined parameter values, move the robot to the shutdown position of Figure 12 and save the DMRAC internal state values at that position for use as initial conditions.
3. Refine the tuning for a 10 degree step from the shutdown position using the initial conditions from Step 2.
4. Refine the tuning from typical min-jerk trajectories from the shutdown position.

The final tuning parameter values after Step 4 are shown in Table 6. The weighting matrix values for Joints 1, 2, and 3 differ from the weighting matrix values for the last three joints by a factor of about 100 which reflects the mass/inertia difference between the upper arm and the wrist. The weighting matrix values which are multiplied by the "x_{m_2}" products are about a factor of seven lower than the values multiplying the "x_{m1}" products since the

second state variable of each decoupled reference model had a higher peak value in a transient. The Joint 1, 2, and 3 reference models have an undamped natural frequency of 4.0 rad/sec where the wrist model used 7.0 rad/sec which again reflected the inertia difference between the upper arm and the wrist. The feed-forward filter values were set to $K_d = 6.0$ and $\tau = 0.1$ for all joints. The alpha values were increased from the initial values of zero to damp out some high frequency oscillations.

5.2 Response

Initially the PUMA arm is in the [0 0 0 0 0 0] position. The final shutdown position was (0, -45, 180, 0, 45, 90) degrees as shown in Fig. 3.Simulation results of the PUMA 560 dynamics responding to the tuned DMRAC controller are displayed in Figure 4.

Note that the model following is excellent for all 6 joints. Furthermore it was observed that all joint torques were smooth and below their saturation limits. In addition, for this specific case, the use of the feedforward component did not significiantly affect the response, although in other cases (eg. step response) use of the feedforward resulted in significant improvements.

6 PUMA Experimental Results

Because the simulator results of the previous section indicated that DMRAC should be useful for robot control, a set of experiments was performed on an actual PUMA 560 manipulator. The tuning process was similar to that described in the previous section. All parameters were initalized at those values from the simulation studies. Only minor variations were required. Final values are in Table 1.

Examples presented illustrate performance of the DMRAC for tracking various trajectories in the presence of static and dynamic load changes. In all cases the robot starts at the shutdown position and follows a trajectory which finishes at the shutdown position.

6.1 Three Joint Trajectory Tracking Study

The trajectory listed in Table 1 is very similar to the one used in the simulation (Section 5). The arm first moves to a straight up position, curls up, and then moves back to the safe position. The wrist joints remain locked in their shutdown positions of {0.0,45.0.90.0} degrees.

The response to the first trajectory is shown in Figure 5. The response is quite good. The effects of stiction can be seen on Joint 2 at $t = 15$ seconds in Figure 5. Figures 6-8 show the model following error and the link torques for Joints 1, 2, and 3 respectively. Figure 6b shows that the Joint 1 torque signal was quite noisy. This noise did not have a physically detectable effect on the actual arm motion. Typically one can feel or hear a noisy torque signal on the actual arm.

The stiction effect mentioned above for Joint 2 can also be seen in figure 4a at $t = 15$ *sec* near the 'X' at the peak error location. When stiction grabs a joint, the error ramps up as does the torque (Figure 4b).

6.2 Static Load Changes

This section describes the ability of the DMRAC algorithm to adjust to static load variations. The trajectory of Table 3 will be run with different loads in the gripper. The algorithm will first be allowed to adjust to the load, and then the trajectory will be started.

The wrist joints remained locked in their shutdown positions of {0.0, 45.0, 90.0} degrees. Five different loads were run for the trajectory - $0kg, 1kg, 2kg, 3kg$, and $4kg$.

Figures 9 and 10 show the response for Joints 2 and 3 respectively. The numbers on the plots are to help identify which curve represents which payload. For Joint 3, the peak errors vary from 2.4390 degrees for the no load case to 3.9972 degrees for the $4kg$ load case. The load changes make up only about 50% of the error. The other 50% is due to the adaptation to the changing arm dynamics. For Joint 2, the peak errors are around 0.8-1.0 degrees. As with Joint 3, the portion of the error due to the load change for Joint 2 is small compared to the no load case.

For Joint 1, the error signals did not vary by more than 0.1 degrees between the five different load cases.

6.3 Dynamic Load Changes

To illustrate the effects of dynamic load change, the trajectory of Table 4 was considered. While running the same trajectory, various loads were added to the gripper while the robot were in motion. The same loads used in the previous section were employed. The wrist joints remained locked in their shutdown positions of {0.0, 45.0, 90.0} degrees. Note: The $1kg$ and $4kg$ loads were added at about $t = 6.76$ seconds and the $2kg$ and $3kg$ loads were added at about $t = 7.34sec$.

Figure 11 shows the model following error for Joint 2 for all loads. The numbers on the graphs indicate which peaks in the error plots match up with the various loads. This figure shows that the DMRAC algorithm has a good load disturbance rejection. The transient period only lasts about 2 seconds.

Figure 12 shows the error for Joint 3 for the various loads. Joint 3 suffers more with a load disturbance having a peak error of almost 5 degrees when the $4kg$ load is added. Again, the transient period is roughly 2 seconds. After the transient, good tracking performance was achieved with the additional loads.

As with the static load case, the model following errors for Joint 1 did not vary by more than 0.1 degrees.

7 Adaptive Control of the Shuttle RMS

7.1 Introduction

Because of the previous demonstrated capabilities of DMRAC, consideration was given to its application to the NASA shuttle Remote Manipulator System (RMS). This system experiences damped oscillations of the end effector after the motion control input by the shuttle operator has been removed [7]. It is desired to design a controller that will take control of the RMS, after the operator releases the motion control joystick, and increase the damping of the oscillations. Linear models have been developed for three manipulator orientations expected to be encountered during normal payload handling [7]. This section discusses work on a direct model reference adaptive controller design for the RMS based on the adaptive algorithm discussed in Section 2.

7.2 Linear Plant and Feedforward

The three linear plants are 3-input, 3-output with 6 states. The plants all have a feedforward compensator $H(s)$, since they are not ASPR. Three types of algorithm feedforward were examined in the course of this work.
static:

$$H(s) = \text{diag3}\{d11, d22, d33\}$$

1st order:

$$H(s) = \text{diag3}\{d11/(\tau_{ii}s + 1)\}$$

2nd order:

$$H(s) = \text{diag3}\{d_{ii}/[(\tau_{1ii}s + 1) * (\tau_{2ii}s + 1)]\}$$

The scalar feedforward provided the best results (based on work to date) and was used for all presented simulations. It was found that, for the scalar feedforward, the combined plants (plant 1,2 or 3 in parallel with feedforward) were all ASPR for:

$$0.125 < d < 1.0$$

where $H(s)$ was $\text{diag3}\{d, d, d\}$. That is, the closed loop system formed from the inverse of $H(s)$ in negative feedback with the respective plant, had all the characteristic roots in the left half plane for:

$$1.0 < d^{-1} < 8.0$$

After simulations were performed with many of the possible combinations of values within this range, it was found that $d = 0.25$ provided the best results for all three orientations.

No ASPR analysis was performed for the 1st and 2nd order feedforwards and the position 1 plant. The stability of the adaptive algorithm for the dynamic feedforwards was somewhat a function of the adaptive gains, T and $\bar{T}$, for given feedforward time constants and gains. Tuning for the dynamic feedforwards was difficult, and very little to

no improvement over the uncontrolled system could be achieved. The dynamic feedforward compensator might possibly provide improved results with further work on the time constants and gains.

7.3 Reference Model

Originally, a reference model were developed for each position using a LQR design based on the uncontrolled plant at the respective orientations. The model used the uncontrolled plant's B, C, and D matrices with the model A matrix formed as follows:

$$A_{m_i} = (A_{p_i} + B_{p_i} * K_i)$$

where K_i is from the LQR design for the i-th orientation.

Satisfactory control of the plants could not be achieved by the adaptive algorithm using these models. A new reference model was then developed using two dominant eigenvalues from the original LQR model for position 1. The new model has a damping ratio of ς and a natural frequency of 1.0 r/sec so that:

$$h_m(s) = 1/(s^2 + 2\varsigma s + 1)$$

This new model was utilized as the reference for each plant output, that is:

$$\begin{aligned} H_m(s) &= \text{diag3}\{h_m(s), h_m(s), h_m(s)\} \\ Y_m(s) &= H_m(s) * u_m(s) \end{aligned}$$

7.4 Simulation Sequence

The sequence for simulation represented a 3.0 second perturbation followed by use of the controller to dampen out oscillations.

The three plant outputs were:

$$\begin{aligned} Y1 &= \text{shoulder yaw} \\ Y2 &= \text{shoulder pitch} \\ Y3 &= \text{elbow pitch} \end{aligned}$$

The plant inputs were limited to 0.7 deg/sec..

In order to simulate the perturbation, the following control sequence was input to the uncontrolled plant:

$$\begin{aligned} u_1 &= 0.7 \; 0 < t < 1.5 \\ u_1 &= -0.7 \; 1.5 < t < 3.0 \\ u_2 &\equiv u_3 \equiv 0. \end{aligned}$$

The resulting plant states at the end of this perturbation were the plant initial conditions for all controller simulations. The plant outputs at the end of the perturbation were the model initial condition (s). The model rate initial condition (s) were set to zero.

7.5 Simulation Results

Figures 13 a,b,c show the position 1 outputs for the following parameters:

$$T = \text{diag9}(6000, 10, 6000, 1, 1, 1, 1, 1, 1)$$
$$\bar{T} = \text{diag9}(1, .000001, 1, 100, 100, 100, 100, 100, 100)$$
$$D = \text{diag3}(.25, .25, .25)$$
$$\text{model damping} = 0.15$$

Note that all controlled outputs decay faster than do the outputs with no control.

Figures 14 a,b,c show the position 2 outputs using the above position 2 controller tuning parameters, and Figures 15 a,b,c show the position 1 outputs using controller parameters, tuned for position 2. For these cases the differences between the controlled and uncontrolled responses were not remarkable.

These results and other experiments show that a satisfactory level of control can be achieved by the MRAC with tuning tailored for the individual positions. Attempts to develop one set of controller tuning parameters that would provide satisfactory control for all three positions were not successful.

8 Conclusions and Recommendations

In summary, the DMRAC algorithm has been found to be an effective robotic control algorithm in both simulation and on the actual robotic manipulator. Its performance was robust with respect to static and dynamic load variations and also disturbances.

At present the DMRAC is being considered for all six joints of the actual PUMA and further tuning with dynamic feedforward is begin considered for the shuttle RMS.

References

[1] K. Sobel and H. Kaufman, "Direct Model Reference Adaptive Control for a Class of MIMO Systems", *Advances in Control and Dynamics Systems*, C.T. Leondes, (ed.), Vol. XXIV, Academic Press, 1986, pp. 245-314.

[2] H. Kaufman, G. Neat and R. Steinvorth, "Asymptotically Stable MIMO Direct Model Reference Adaptive Controller for Processes not Necessarily Satisfying a Positive Real Constraint," European Cont. Conf., Grenoble, July 1991, pp. 1872-1877.

[3] I. BarKana, "Adaptive Control - A Simplified Approach," in Academic Press Advances in Control and Dynamic Systems, C.T. Leondes (ed.), Vol. 25, 1987.

[4] R. Steinvorth, H. Kaufman, and G. Neat, "Direct Model Reference Adaptive Control with Application to Flexible Robotics," Proc. of IASTED Conf. on Adaptive Control and Signal Proc., NY, Oct. 1990, pp. 9-12.

[5] R. Steinvorth and H. Kaufman, "Direct Model Reference Adaptive Control of Robots," CISS, Baltimore, March 1991, pp. 667-672.

[6] B. Armstrong, O. Khatib, and J. Bardick, "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", Proc. 1986 IEEE Robotics and Automation Conf., San Francisco, pp. 510-518.

[7] M.A. Scott, M.G. Gilbert, and M.E. Demeo, "Active Vibration Damping of the Space Shuttle Remote Manipulator System", NASA Report TM- 104149, August 1991.

Table 1: Parameter Values for 3 Joint Trajectory Tracking Runs

$\bar{T}$ (diag component)	"e_z"	20	40	40
	"x_m"	140	20	200
		30	200	30
	"u_m"	140	200	200
T (diag component)	"e_z"	30	30	40
	"x_m"	200	30	400
		60	400	60
	"u_m"	200	400	400
joint		1	2	3
Model	w_n	10	10	10
	ζ	1	1	1
Feed Forward	K_d	6	6	6
	τ	0.05	0.05	0.05
alpha	α	0.02	0.02	0.02

Table 2: First Three Joint Tracking Test Trajectory

Knot Point	Joint Positions (deg)			Time (sec)
	1	2	3	
0	0	-45	180	-
1	-90	-90	90	6
2	0	0	180	8
3	0	-45	180	6

Table 3: Static Load Change Trajectory

Knot Point	Joint Positions (deg)			Time (sec)
	1	2	3	
0	0	-45	180	-
1	0	-45	180	3
2	45	0	0	10
3	0	-45	180	10

Table 4: Dynamic Load Change Trajectory

Knot Point	Joint Positions (deg)			Time (sec)
	1	2	3	
0	0	-45	180	-
1	0	-45	180	3
2	45	-90	90	10
3	0	-45	180	10

Table 5: Tunable Parameters

Parameters	Description	Values
T	24 x 24 integral weighting matrix	576
$\bar{T}$	24 x 24 proportional weighting matrix	576
w_{n_i}	Undamped natural frequency for Joint i model	6
ς_i	Damping ratio for Joint i model	6
α	6 x 6 diagonal plant derivative weighting matrix	6
K_{d_i}	DC gain of Joint i supplementary feed-forward block	6
τ_i	Time constant of Joint i supplementary feed-forward block	6
	Total	1182

Table 6: Final Parameter Values

T (diag component)	"e_z"	20	40	22	0.2	0.2	0.2
	"x_m"	140	20	140	35	100	22
		1.4	0.2	1.4	0.2	1.4	0.2
	"u_m"	140	160	110	1.4	1.4	1.4
T (diag component)	"e_z"	20	60	25	0.2	0.2	0.2
	"x_m"	140	20	150	35	140	25
		1.4	0.2	1.4	0.2	1.4	0.2
	"u_m"	140	160	130	1.4	1.4	1.4
Joint		1	2	3	4	5	6
Model	w_n	4	4	4	7	7	7
	ς	1	1	1	1	1	1
Feed Forward	K_d	6	6	6	6	6	6
	τ	0.1	0.1	0.1	0.1	0.1	0.1
alpha	α	0.0.35	0.02	0.02	0.01	0.01	0.01

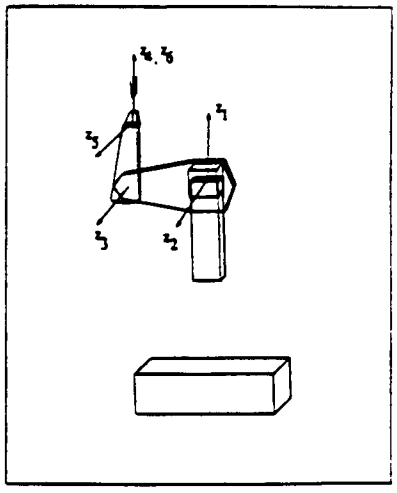


Figure 1: PUMA 560 Coordinate Frame

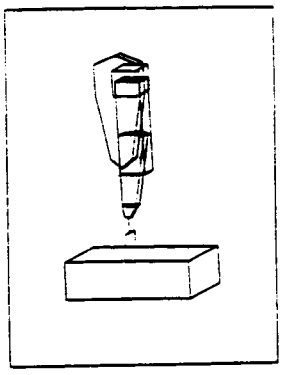


Figure 2: PUMA 560 in Stable Equilibrium

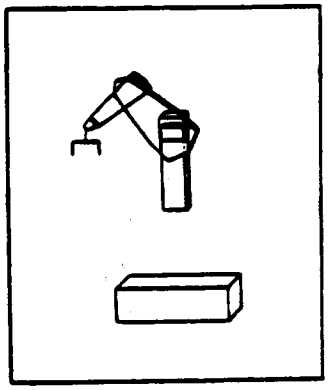


Figure 3: Shutdown Position

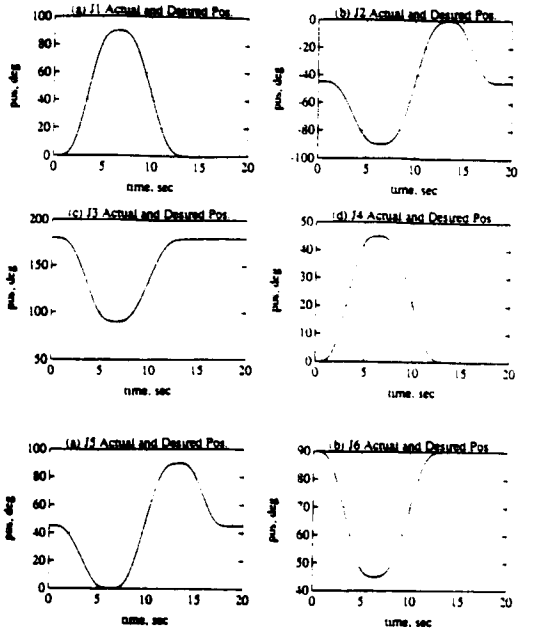


Figure 4: Simulation results for all 6 joints

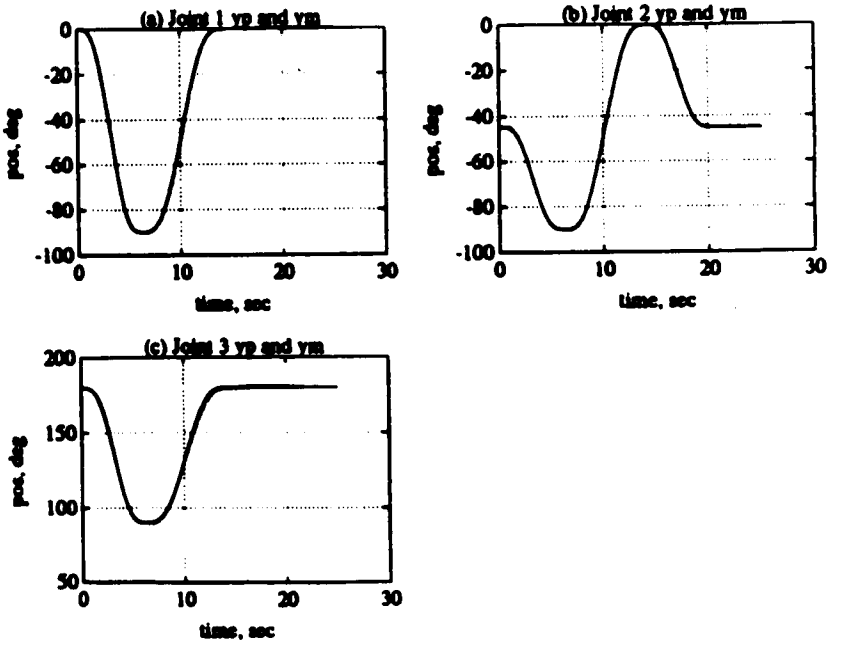


Figure 5: Plant and Model Output for First Trajectory. (a) Joint 1. (b) Joint 2. (c) Joint 3.

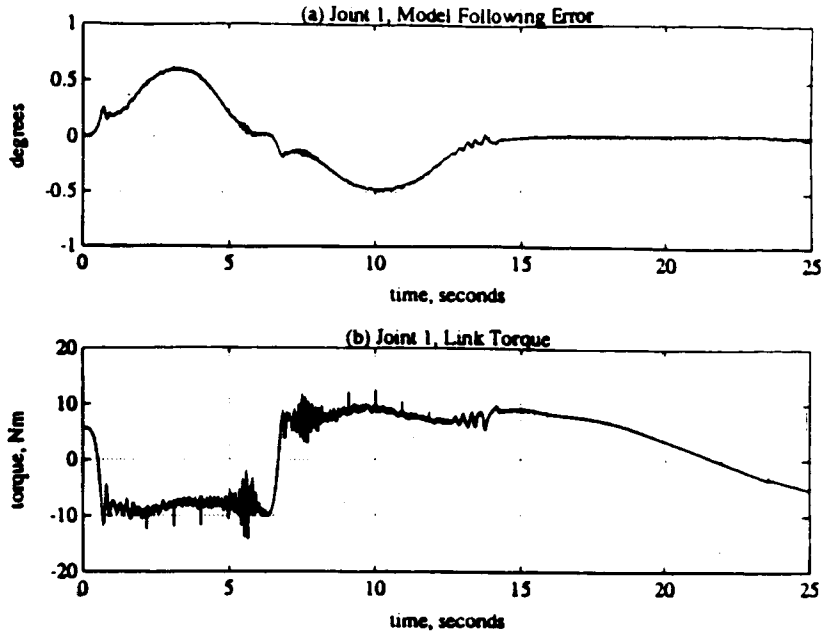


Figure 6: Joint 1 Data for First Trajectory. (a) Model following error. (b) Joint Troque.

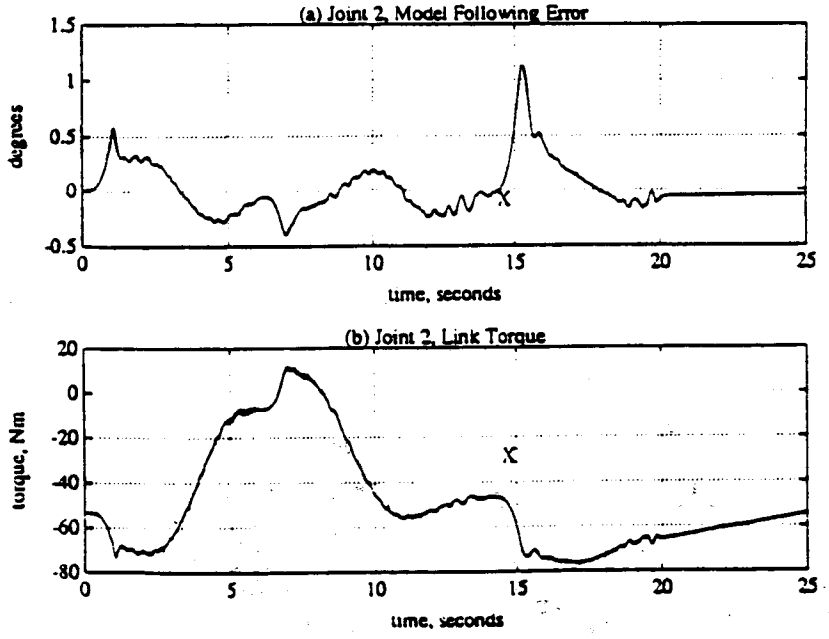


Figure 7: Joint 2 Data for First Trajectory. (a) Model Following error. (b) Joint Torque.

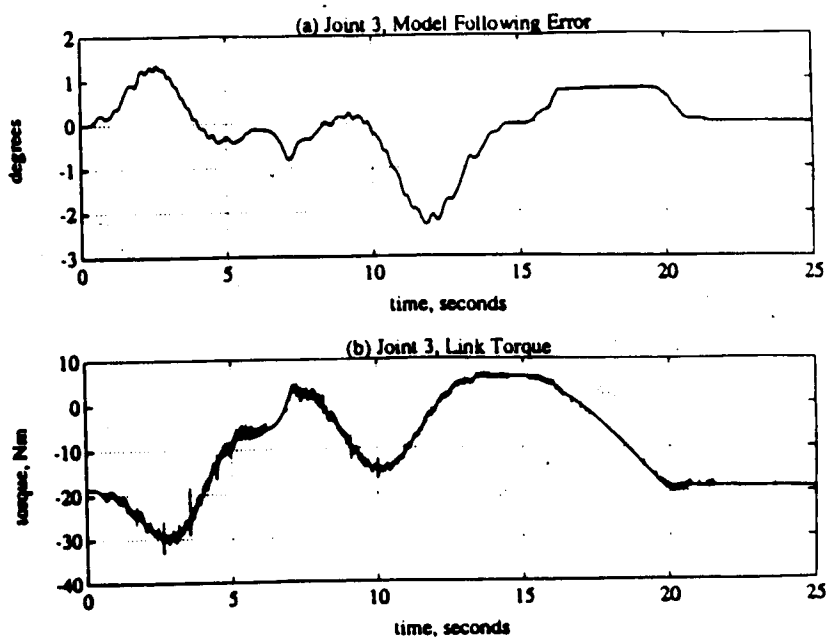


Figure 8: Joint 3 Data for First Trajectory. (a) Model following error. (b) Joint Torque.

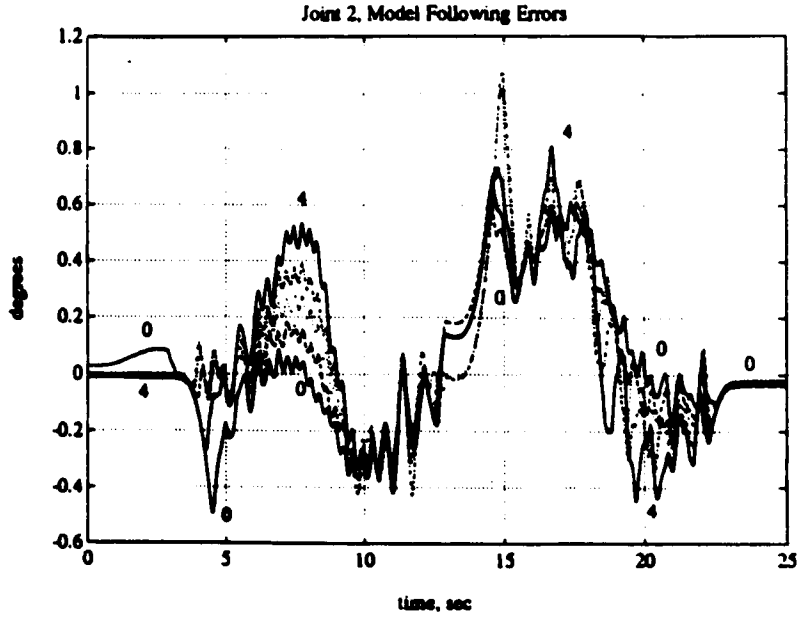


Figure 9: Joint 2 Static Load Model Following Error

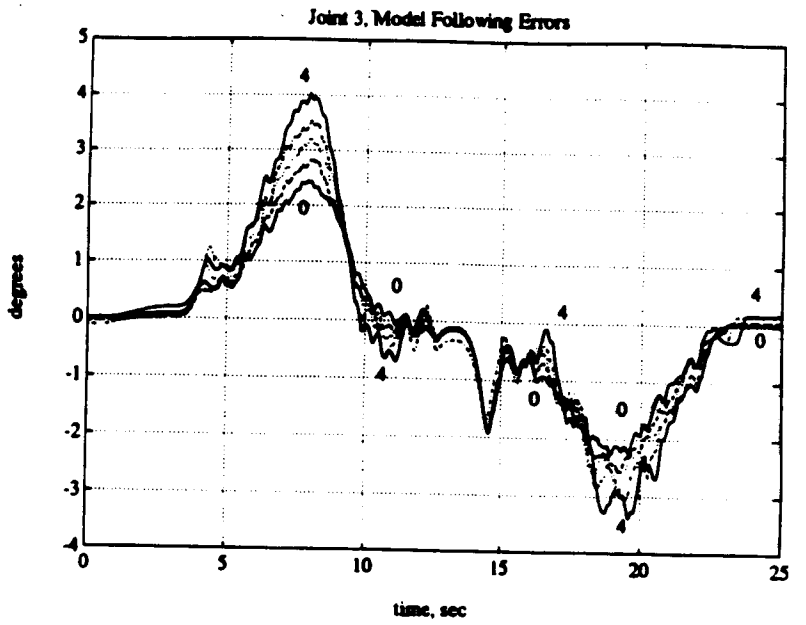


Figure 10: Joint 3 Static Load Model Following Error

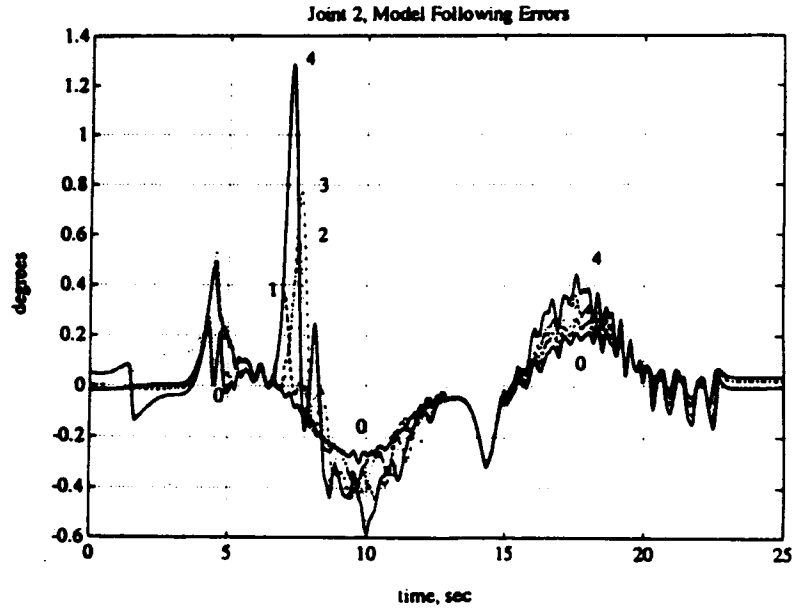


Figure 11: Joint 2 Dynamic Load Model Following Errors

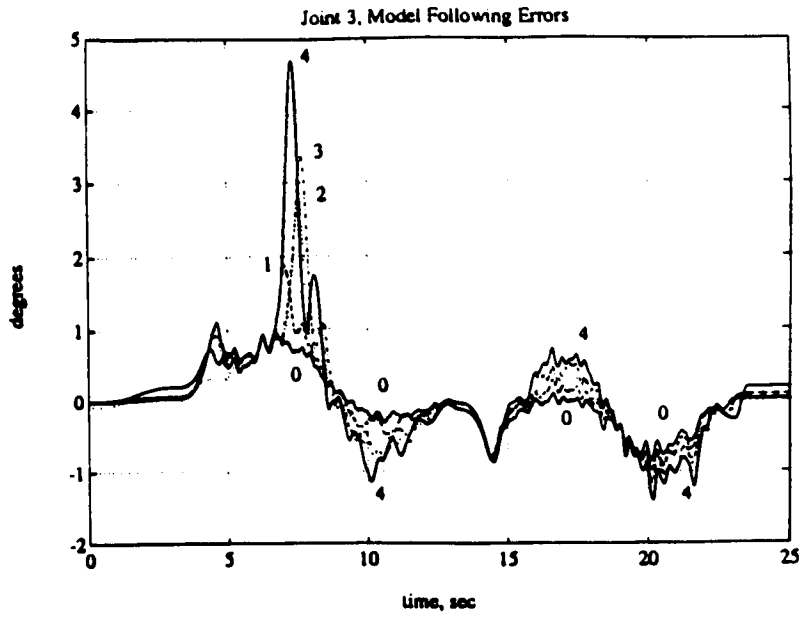


Figure 12: Joint 3 Dynamic Load Model Following Errors

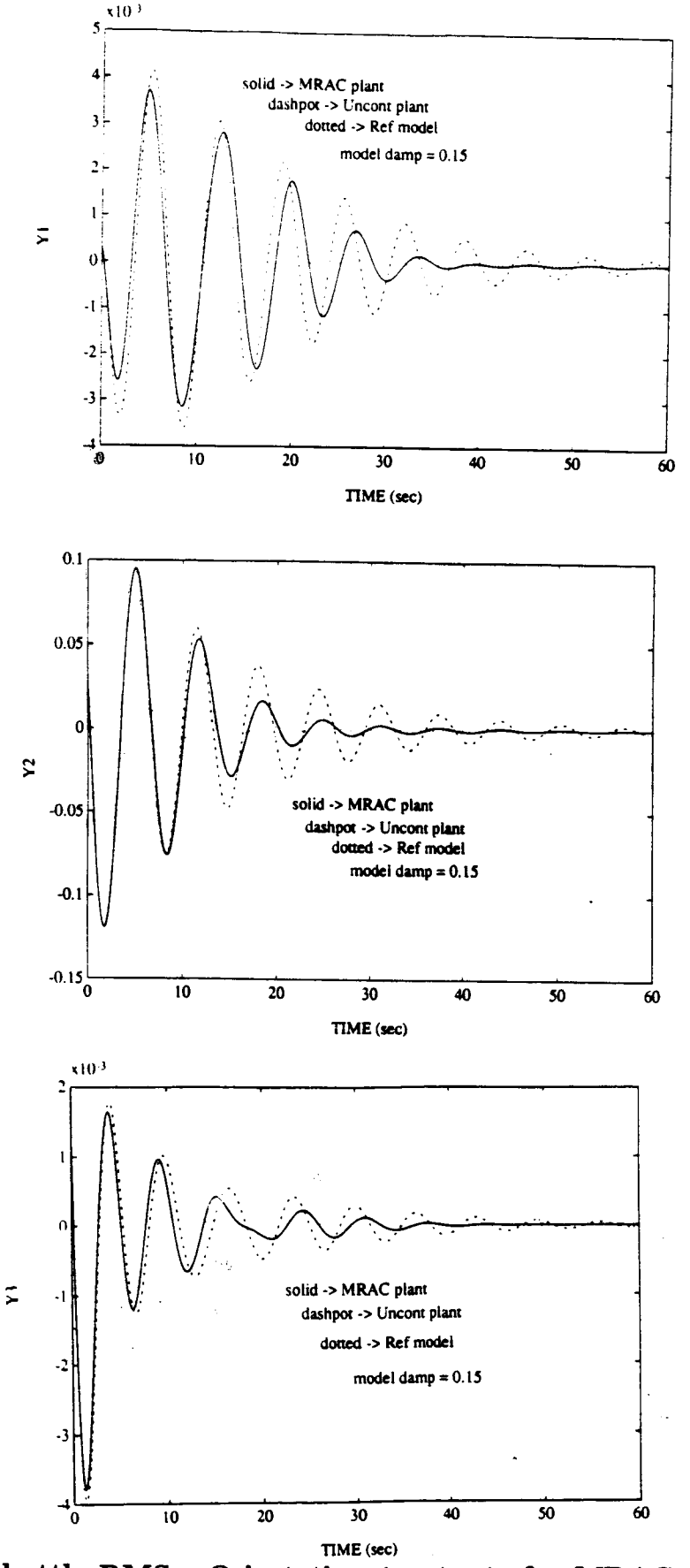


Figure 13: Shuttle RMS - Orientation 1 outputs for MRAC tuned to this position

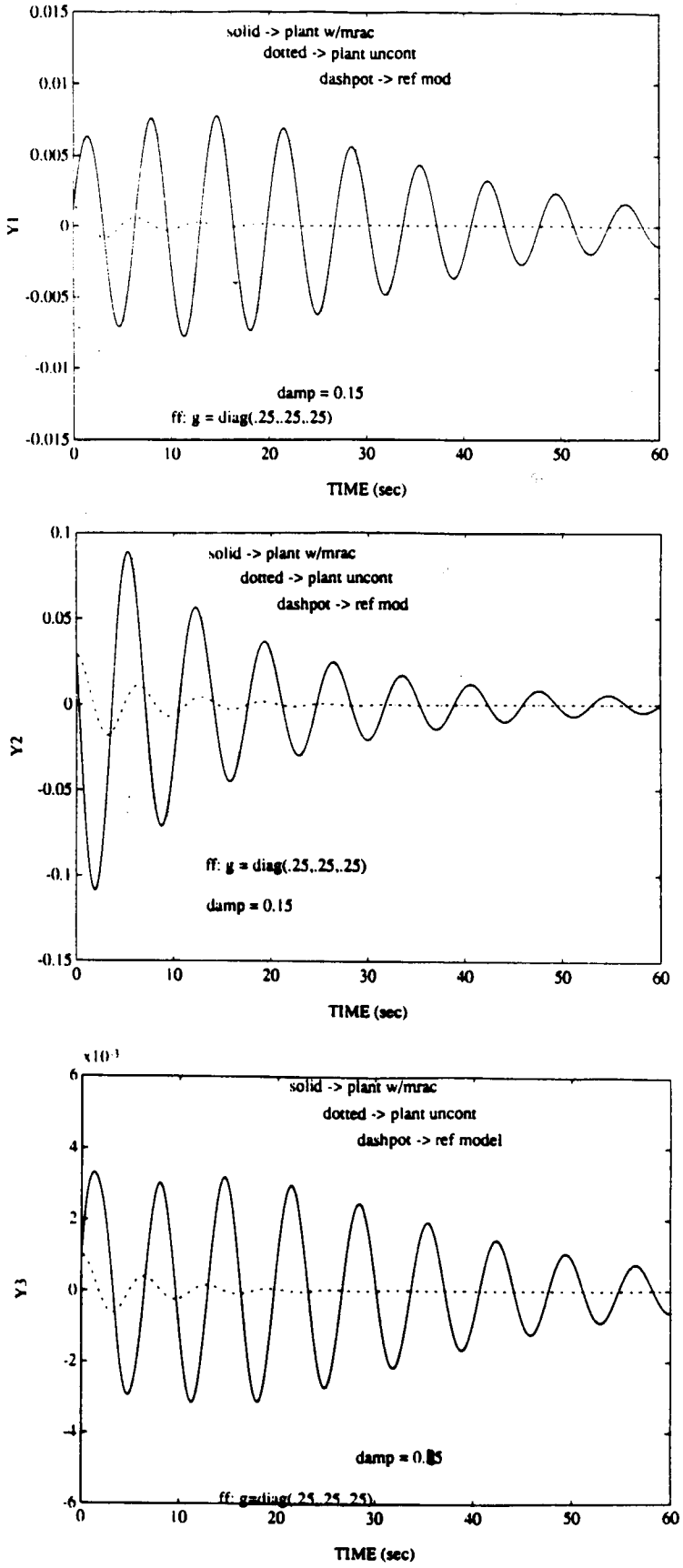


Figure 14: Shuttle RMS - Orientation 2 outputs using the MRAC tuned for orientation 1

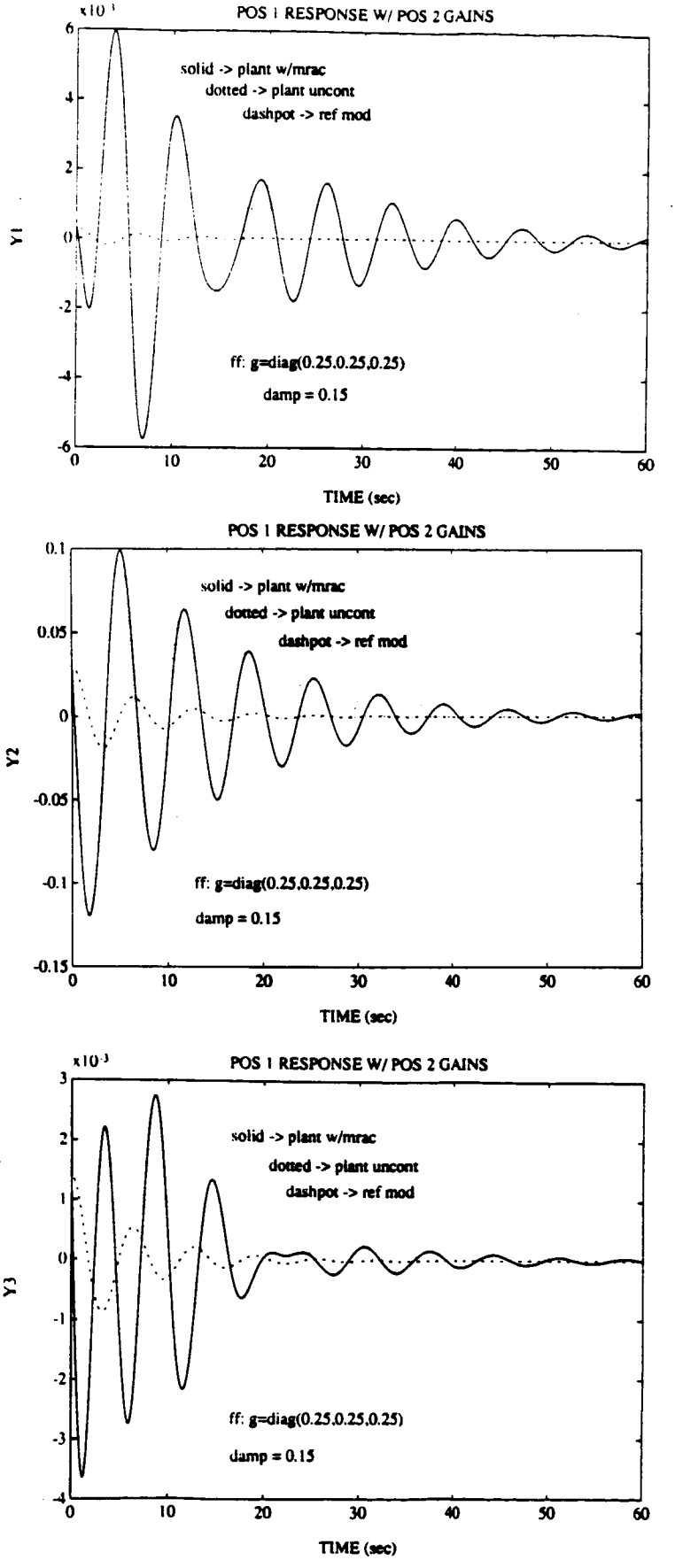


Figure 15: Shuttle RMS - Orientation 1 outputs using the MRAC tuned for orientation 2

Performance Measures from the Explorer Platform Berthing Experiment

Stephen Leake
NASA Goddard Space Flight Center
Greenbelt, Maryland

Performance measures from the Explorer Platform berthing experiment

Stephen Leake
NASA Goddard Space Flight Center
Robotics Lab
February 17, 1993

1 Introduction

The Explorer Platform is a Modular Mission Spacecraft; it has several sub-units that are designed to be replaced on orbit. The Goddard Space Flight Center Robotics Lab undertook an experiment to evaluate various robotic approaches to replacing one of the units; a large (approximately 1 meter by 1 meter by 0.5 meter) power box. The hardware (see figure 1) consists of a Robotics Research Corporation K-1607 (RRC) manipulator mounted on a large gantry robot, a Kraft handcontroller for teleoperation of the RRC, a Lightweight Servicing Tool (LST) mounted on the RRC, and an Explorer Platform mockup (EP) with a removable box (MMS) that has fixtures that mate with the LST. Sensors include a wrist wrench sensor on the RRC, and Capaciflectors [Vranish91] mounted on the LST and the MMS. There are also several cameras, but no machine vision is used. The control system for the RRC is entirely written by Goddard [Leake91]; it consists of Ada code on three Mulitbus I 386/387 CPU boards doing the real-time robot control, and C on a 386 PC processing Capaciflector data. The gantry is not moved during this experiment.

The task is the exchange of the MMS; it is removed and replaced. This involves four basic steps: mating the LST to the MMS, demating the MMS from the EP, mating the MMS to the EP, demating the LST from the MMS. Each of the mating steps must be preceeded by an alignment to bring the mechanical fixtures within their capture range.

Two basic approaches to alignment are explored; teleoperation with the operator viewing thru cameras, and Capaciflector based autonomy. To eval-

uate the two alignment approaches, we ran several runs with each approach, and recorded the final pose. Comparing this to the ideal alignment pose gives accuracy and repeatability data. In addition, the wrenches exerted during the mating tasks were recorded; this gives information on how the alignment step affects the mating step.

There are also two approaches to mating; teleoperation, and impedance-based autonomy. The wrench data taken during mating using these two approaches is used to evaluate them.

Section 2 describes the alignment results, Section 3 describes the mating results, and finally Section 4 gives some conclusions.

2 Alignment

The two alignment tasks are aligning the LST for mating with the MMS, and aligning the MMS for mating with the EP. Two methods were used for each task; teleoperation, and Capaciflector-based autonomy.

For teleoperation, we used the Langley rate control algorithm. One experienced operator performed all the runs. The Kraft hand-controller acts like a 6 DOF joystick; the rate of the RRC tool frame is proportional to the displacement of the Kraft from a reference frame. In a traditional joystick, there is a centering spring force returning the joystick to the reference frame; in the Langley algorithm, this centering force has a constant magnitude, not proportional to the displacement. This allows wrench feedback to be added to the centering force without operator confusion. On the RRC, there is a Cartesian impedance algorithm using the wrench sensor, that makes the RRC tool frame behave like a pure damper; it relaxes when any force is applied. Thus if the tool is against a surface, and the operator pushes the hand-controller into the surface, the hand-controller commands a constant rate, which is turned into a constant force by the damper algorithm on the RRC. At the same time, the wrench sensed by the RRC wrench sensor is fed back to the motors on the hand-controller. Wrench feedback ratios of 1:1 can be achieved with this algorithm. Since the alignment task is primarily free-space positioning, the wrench feedback was low for this task, to mask noise and errors in the gravity model of the loads carried by the wrench sensor. Three cameras were used during teleop; one on the LST (only used

for mating the LST to the MMS), one giving an overall view of the RRC and EP, and one giving a good view of the MMS mounted on the EP.

For Capaciflector-based autonomy, there are two Capaciflectors mounted on the LST, and six on the MMS. Alignment is a 6 DOF task; this is easily accomplished with the six sensors on the MMS. For the LST, there is no way to place six sensors to get a full alignment. So a sequential approach is used; first the two sensors are leveled against a surface, then they find an edge, then they find a bump along the edge, etc. Each step in the sequence can find two degrees of freedom; we used a 7 step sequence to help eliminate errors.

For each task, a "perfect" goal pose was defined manually (using rulers and direct vision to align the LST and MMS). Then 10 runs for each combination of task and approach were made, recording the final pose for each run. The accuracy is defined as the mean error between the "perfect" goal pose and the 10 actual poses; the repeatability is the standard deviation of the same error. For the MMS, there are not enough visual cues to allow the operator to align the MMS with the EP without contact. So the MMS_TELEOP task did both alignment and initial contact; the MMS_CAPACIFLECTOR task did not contact. The following table summarizes the results, and the scatter plots in figures 2 thru 5 show the raw data.

run	time (sec)	translation (mm)		rotation (radians)	
		accuracy	repeat	accuracy	repeat
LST_TELEOP	53 +- 10	7.39	5.29	0.02789	0.07483
LST_CAPACIFLECTOR	123 +- 0.5	5.69	0.45	0.01883	0.00096
MMS_TELEOP	42 +- 10	7.49	15.5	0.01920	0.04911
MMS_CAPACIFLECTOR	61 +- 20	2.88	11.7	0.01092	0.03554

For both tasks, teleoperation is significantly faster. The LST task shows a 23% improvement in translation accuracy from teleop to Capaciflector, and a 32% improvement in rotation accuracy. The repeatability is significantly better with the Capaciflector; a factor of 11 for translation, 77 for rotation. For the MMS task, the accuracy shows a 61% improvement in translation, and 43% improvement in rotation. The repeatability shows a 24% improvement in translation, and 27% in rotation. Remember that the teleoperation MMS task used contact for the final alignment, while the Capaciflector made no contact; the Capaciflector algorithm is more repeatable than the mechanical contact!

For the MMS, it often took 1 or 2 incorrect contacts before the final correct contact was made.

For the LST, the operator felt teleoperation was more reliable, while for the MMS, he felt the Capaciflector was more reliable.

3 Mating

Wrench data was logged for both LST and MMS mating, recording data from just before contact until after full contact. The fixtures in both tasks guarantee 6 DOF alignment. The LST is essentially rigid; the contact between the LST and the MMS fixture is basically a narrow cone, with a plate and two posts at the top for roll alignment. The clearance between the LST and the mating fixtures on the MMS is about 2 mm and 0.1 radians. The contact between the MMS and the EP is at three points (before the screw is fastened); there is no clearance, but the contacts are actually spherical, so some misalignment is possible. No screws were tightened or latches fastened, to simplify data analysis. Two methods were used for each mating task; Cartesian impedance control, and teleoperation. The LST mating was then repeated using a more complex impedance control. For teleoperation, one experienced operator performed all the runs, starting from the same starting point as the teleop aligment task. For autonomy, the start pose was representative of the final alignment pose using Capaciflectors. Thus the differences include differences in starting alignment as well as mating algorithm.

For teleoperation, we again use the Langley rate algorithm. The wrench feedback and RRC damper gains are adjusted to give the best operator feel, while maintaining stability. The best operator feel is achieved when the joint stiffness is very low. If the joint stiffness is high, the Cartesian impedance loop has too work very hard to overcome it, and this shows up as instability at high wrench feedback gains. However, to use a low joint stiffness, we must use gravity compensation torque, to keep the arm from sagging. Unfortunately, a design flaw in the RRC analog servo hardware prevents us from using a torque command when carrying the MMS payload, so we had to use relatively high joint stiffness for the MMS. We repeated the LST runs using a softer joint servo with gravity compensation. The actual

gains for each run are given in the appendix.

The performance measures are the time from first contact to stable contact, rms wrench error, and the maximum wrench. The Z axis is the mating direction; wrench error in this direction is measured only after stable contact. The following table summarizes the results; figures 6 thru 13 show the raw wrench logs for a representative run.

run	time	rms wrench error						max wrench	
		TX	TY	TZ	RX	RY	RZ	tran	rot
LST_MATE_TELEOP	14.6	1.07	3.34	14.60	0.95	0.40	0.12	53.6	2.48
LST_MATE_AUTO	6.0	1.58	1.58	22.97	1.65	3.76	0.14	79.4	8.29
MMS_MATE_TELEOP	32.2	11.3	30.7	55.6	4.69	8.07	2.36	250.3	27.8
MMS_MATE_AUTO	2.2	3.82	2.38	0.71	0.83	1.77	0.39	82.1	4.65
LST_MATE_SOFT_TELEOP	7.2	4.49	1.88	6.14	0.67	1.26	0.52	51.6	4.19
LST_MATE_SOFT_AUTO	4.5	4.60	5.01	15.7	0.82	1.77	0.27	105.4	8.11

For both tasks, teleop was slower than autonomy.

For the LST task, teleop gave lower wrench errors, and lower maximum wrenches. This is attributed to the fact that the operator used vision to refine the alignment as mating proceeded.

For the MMS task, there is a factor of at least 5 improvement in wrench errors for autonomy; the off-insertion-axis portion can be attributed to the more accurate and reliable Capaciflector alignment, while the on-insertion-axis portion is due to the more stable mating algorithm.

Note that the LST_MATE run was slightly unstable when fully mated, and that MMS_MATE_TELEOP typically made contact twice incorrectly before finally seating.

The operator would always prefer to use the autonomous impedance algorithm, not teleop. The control system is actually more complex for teleop.

The more complex gravity compensation control system gave faster times and lower on-insertion-axis forces, but higher off-insertion-axis wrench errors. It was more stable than the non-gravity compensation system.

4 Conclusions

For both tasks, the Capaciflector gave an improvement of at least 25% in alignment accuracy and repeatability. For the LST task, the repeatability improved by a factor of at least 11.

For the LST task, teleoperation alignment followed by teleoperation mating gave lower wrench errors, by a factor of about 2. For the MMS task, Capaciflector alignment followed by autonomous mating gave lower wrench errors by a factor of at least 5. These results are not conclusive; more work needs to be done to distinguish between the effects of initial alignment and mating algorithm.

We anticipate significant reductions in the wrench errors and maximum wrenchs with future control system improvements, for both autonomy and teleoperation.

5 Acknowledgments

Ed Cheung of the Goddard Robotics Lab wrote the C code for the Capaciflectors, and developed the Capaciflector alignment algorithms.

6 References

[Vranish91] J. Vranish, R. McConnell, S. Mahalingam, "Capaciflector Collision-Avoidance Sensors for Robots", International Journal of Computer and Electrical Engineering, Vol 17 #3, 1991

[Leake91] Stephen Leake, "A Cartesian Force Reflecting Teleoperation System", International Journal of Computer and Electrical Engineering, Vol 17, no 3, 1991.

7 appendix

LST_MATE_TELEOP uses teleoperation in the Langley mode, with the following parameters:

```
Motion_Scale => (others => 0.1),
Wrench_Feedback => (Active => TRUE, Scale => (others => 0.25)),
Joint_Servo =>
    (Joint_Servo_Label => ANALOG_DAMPING_NOGRAV,
     Pos_Error_Action => CLIP,
     ADN_Vel_Gain => (6.0, 6.0, 6.0, 6.0, 6.0, 4.0, 4.0)),
Cart_Impedance => (Active => TRUE,
     Bias => (others => 0.0),
     Spring => (others => 0.0),
     Damper => (1000.0, 1000.0, 1000.0, 200.0, 200.0, 100.0))
```

LST_MATE_AUTO uses Cartesian impedance control, with the following parameters:

```
Joint_Servo =>
    (Joint_Servo_Label => ANALOG_DAMPING_NOGRAV,
     Pos_Error_Action => CLIP,
     ADN_Vel_Gain => (6.0, 6.0, 6.0, 6.0, 6.0, 4.0, 4.0)),
 Cart_Impedance => (Active => TRUE,
     Bias => (TZ => 40.0, others => 0.0),
     Spring => (others => 0.0),
     Damper => (4000.0, 4000.0, 4000.0, 1600.0, 1600.0, 400.0))
```

MMS_MATE_TELEOP uses teleoperation in the Langley mode, with the following parameters:

```
Motion_Scale => (others => 0.05),
Wrench_Feedback => (Active => TRUE, Scale => (others => 0.125)),
Joint_Servo =>
    (Joint_Servo_Label => ANALOG_DAMPING_NOGRAV,
     Pos_Error_Action => CLIP,
     ADN_Vel_Gain => (6.0, 6.0, 6.0, 6.0, 6.0, 4.0, 4.0)),
```

```
Cart_Impedance => (Active => TRUE,
     Bias => (others => 0.0),
     Spring => (others => 0.0),
     Damper => (4000.0, 4000.0, 4000.0, 800.0, 800.0, 800.0)),
```

MMS_MATE_AUTO uses Cartesian impedance control, with the following parameters:

```
Joint_Servo =>
    (Joint_Servo_Label => ANALOG_DAMPING_NOGRAV,
     Pos_Error_Action => CLIP,
     ADN_Vel_Gain => (6.0, 6.0, 6.0, 6.0, 6.0, 4.0, 4.0)),
 Cart_Impedance => (Active => TRUE,
     Bias => (TZ => 80.0, others => 0.0),
     Spring => (others => 0.0),
     Damper => (4000.0, 4000.0, 4000.0, 1600.0, 1600.0, 1600.0))
```

LST_MATE_SOFT_TELEOP uses teleoperation in the Langley mode, with the following parameters:

```
Motion_Scale => (others => 0.2),
Wrench_Feedback => (Active => TRUE, Scale => (others => 0.5)),
Joint_Servo =>
    (Joint_Servo_Label => PD_GRAV,
     Pos_Error_Action => CLIP,
     PDG_Stiffness => (9000.0, 3562.55, 2625.63, 2341.13, 341.75, 385.42,  80.
     PDG_Damping =>   (900.0,   580.59,  318.80,  281.04,  33.40,  37.04,   1.
Cart_Impedance => (Active => TRUE,
     Bias => (others => 0.0),
     Spring => (others => 0.0),
     Damper => (1000.0, 1000.0, 1000.0, 100.0, 100.0, 100.0)),
```

LST_MATE_SOFT_AUTO uses Cartesian impedance control, with the following parameters:

```
Joint_Servo =>
    (Joint_Servo_Label => PD_GRAV,
```

```
    Pos_Error_Action => CLIP,
    PDG_Stiffness => (9000.0, 3562.55, 2625.63, 2341.13, 341.75, 385.42,  80.
    PDG_Damping =>   (900.0,   580.59,  318.80,  281.04,  33.40,  37.04,   1.
Cart_Impedance => (Active => TRUE,
    Bias => (TZ => 40.0, others => 0.0),
    Spring => (others => 0.0),
    Damper => (1000.0, 1000.0, 1000.0, 100.0, 100.0, 100.0)).
```

File: EP_LST_TELEOP.PLOT

Time: 14:11:39.73 17-FEB-1993

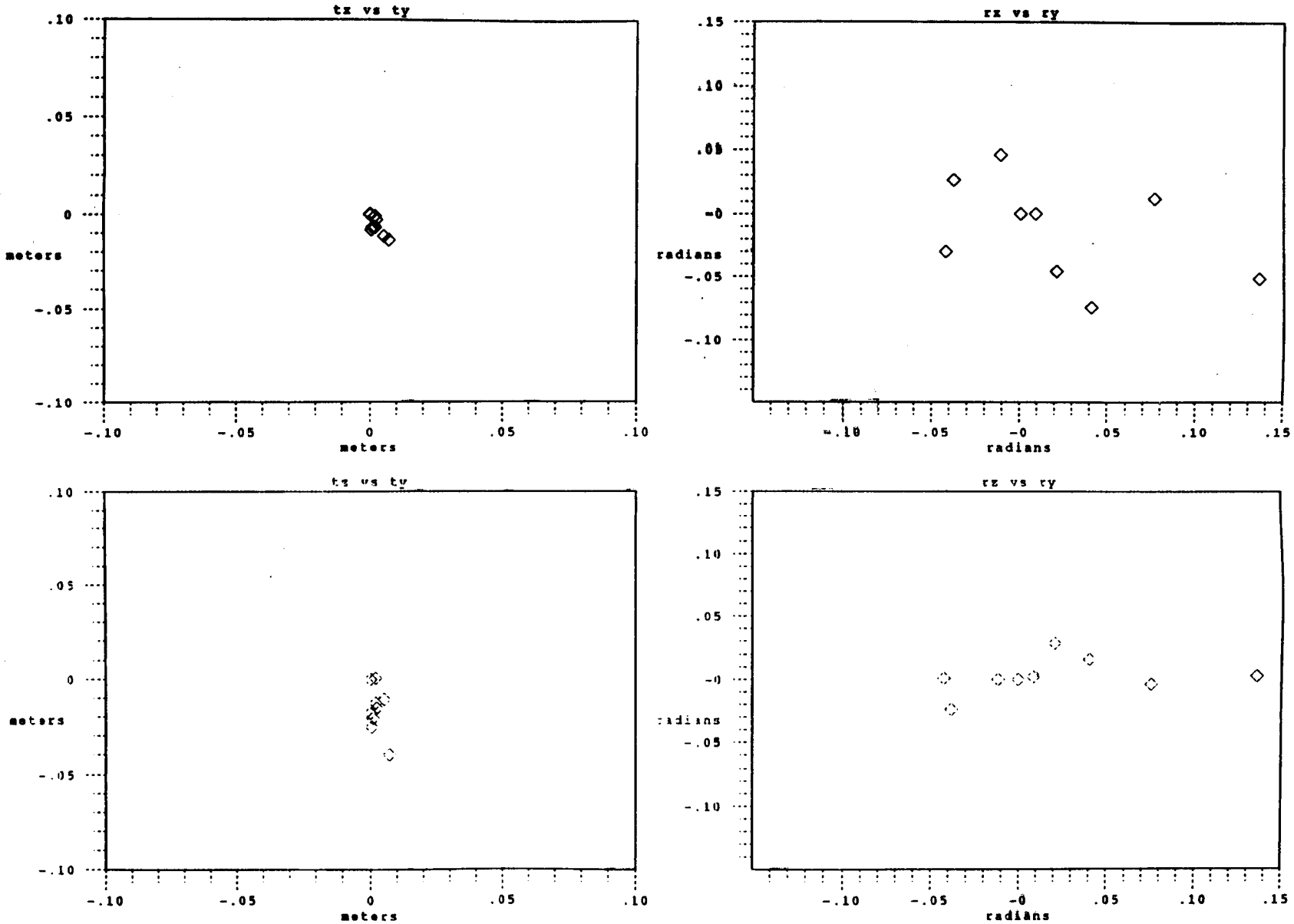
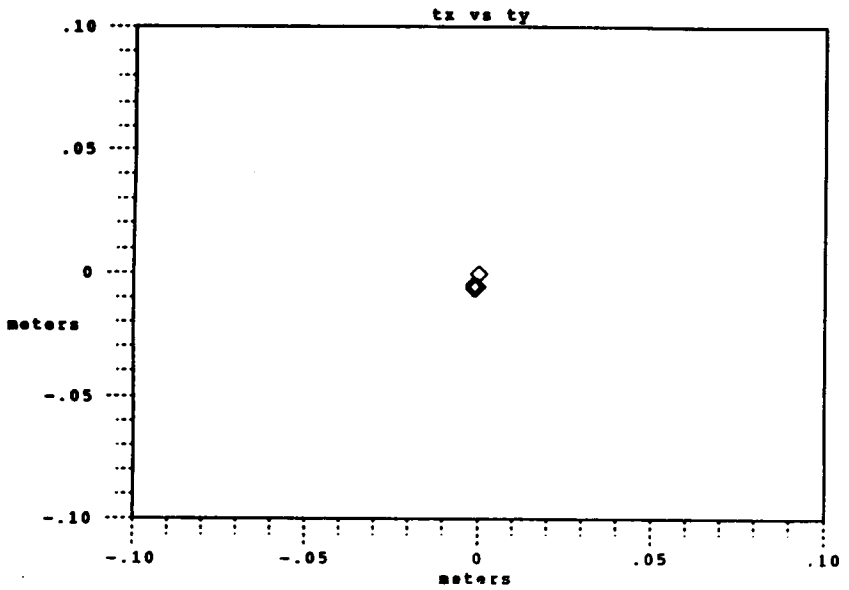
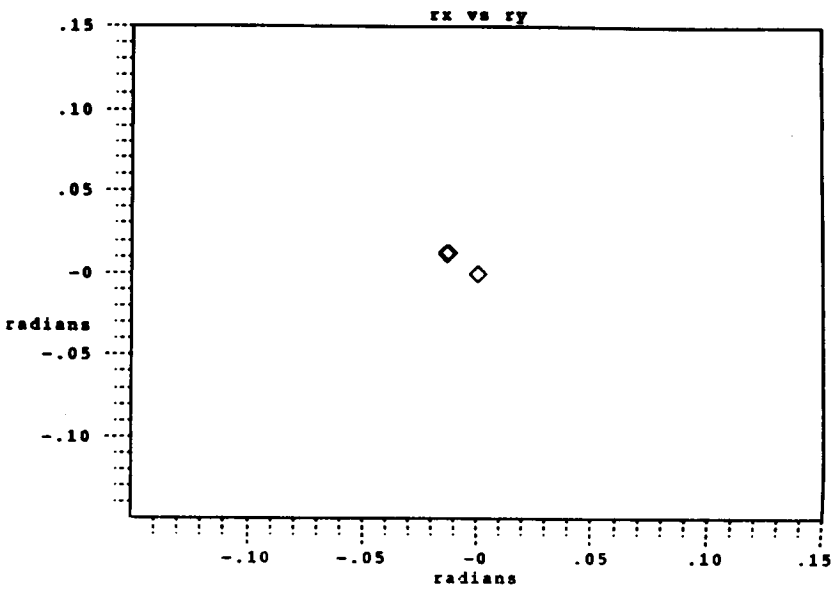


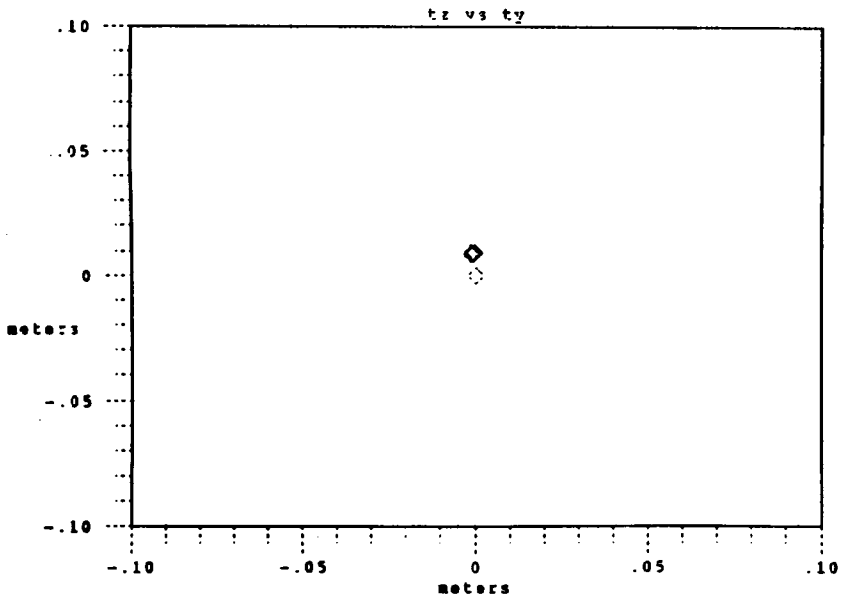
figure 2

File: EP_LST_CAPACIFLECTOR.PLOT

Time: 14:09:55.60 17-FEB-1993







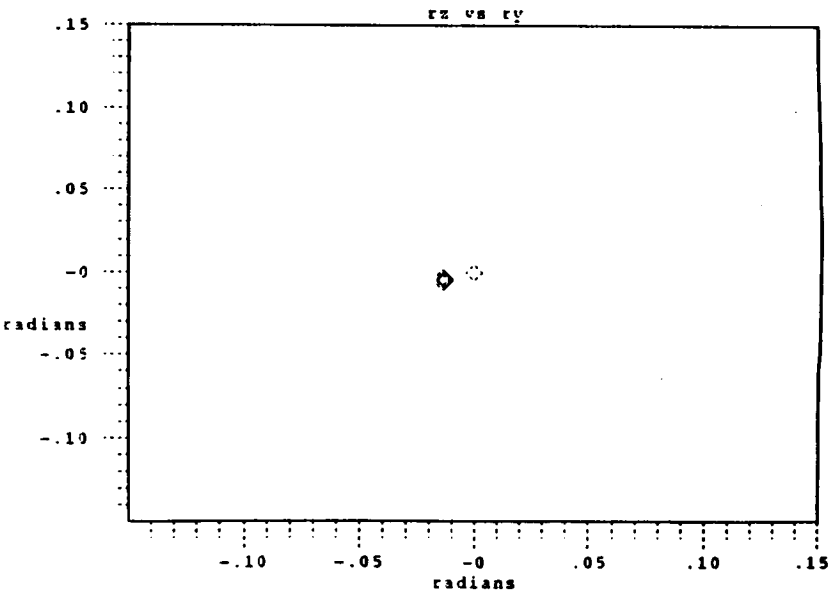
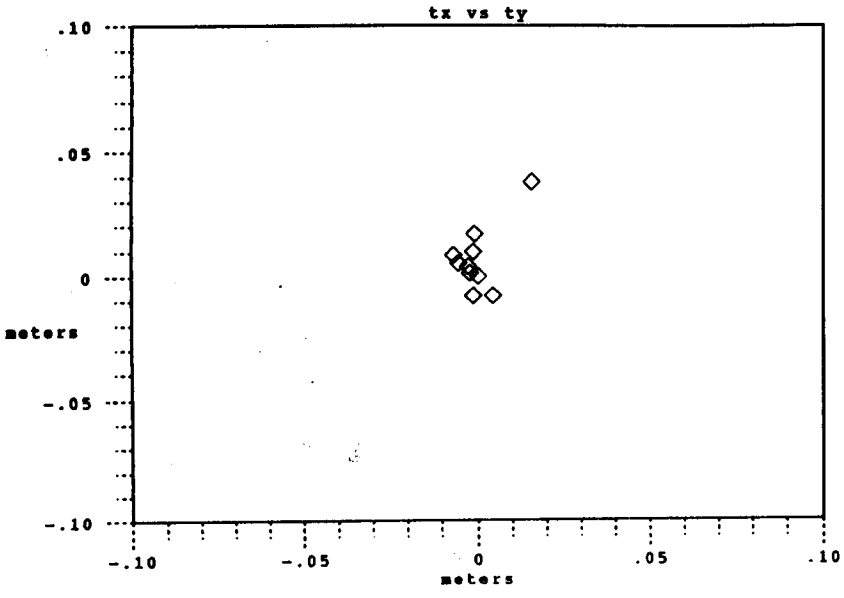
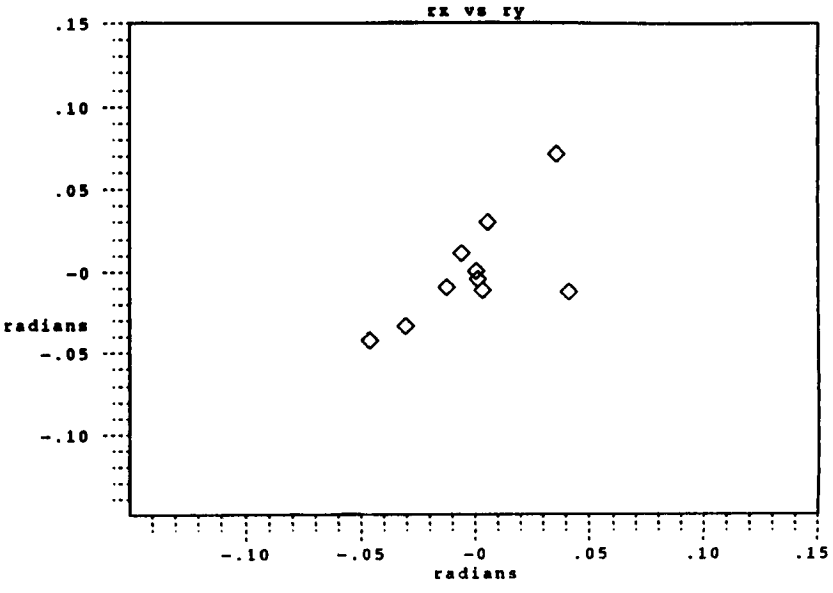


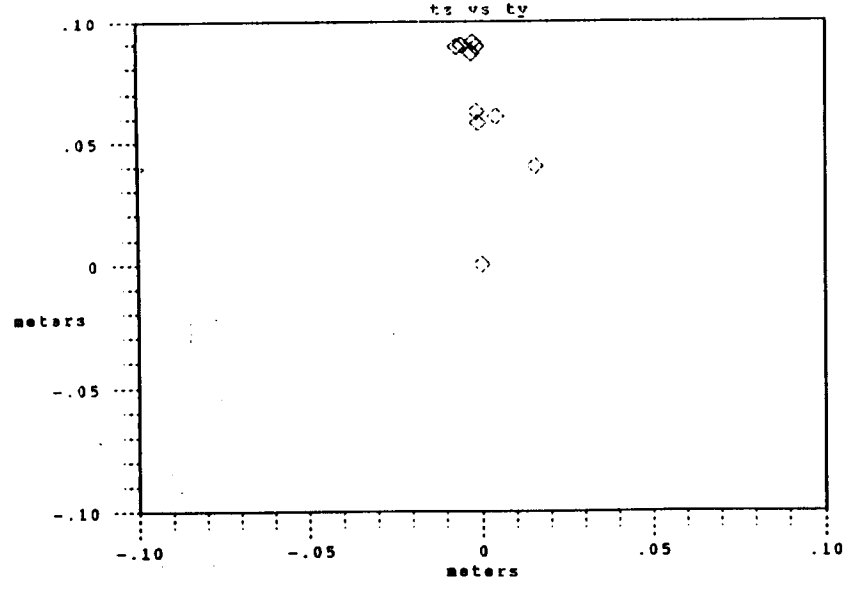
figure 3

File: EP_MMS_TELEOP.PLOT

Time: 14:15:04.50 17-FEB-1993







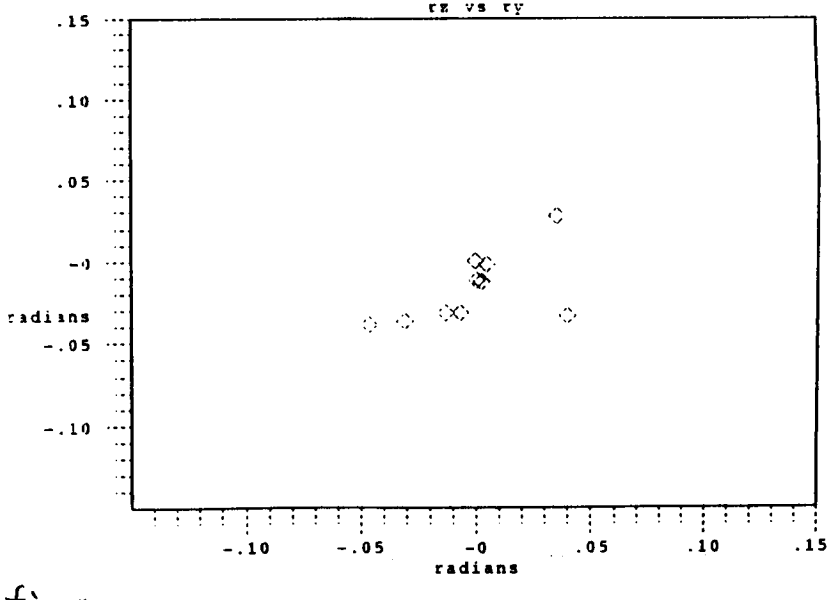


figure 4

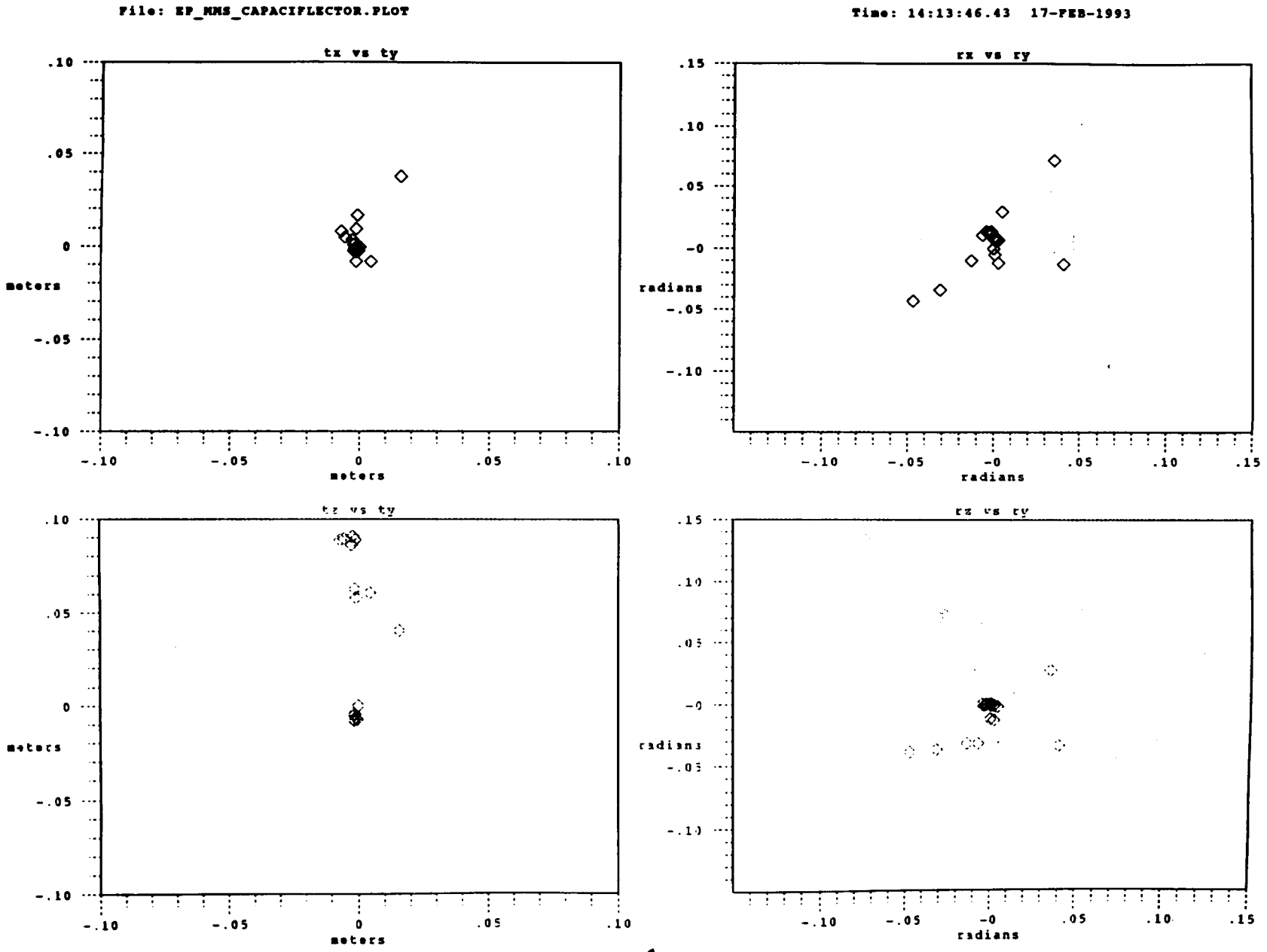

File: EP_MMS_CAPACIFLECTOR.PLOT
Time: 14:13:46.43 17-FEB-1993
tx vs ty
ty vs tz
rx vs ry
rz vs ry
meters
radians
.10
.05
0
-.05
-.10
.15
-.15

figure 5

File: [.LST]MATE_TELEOP.PLOT

Time: 15:36:58.30 17-FEB-1993

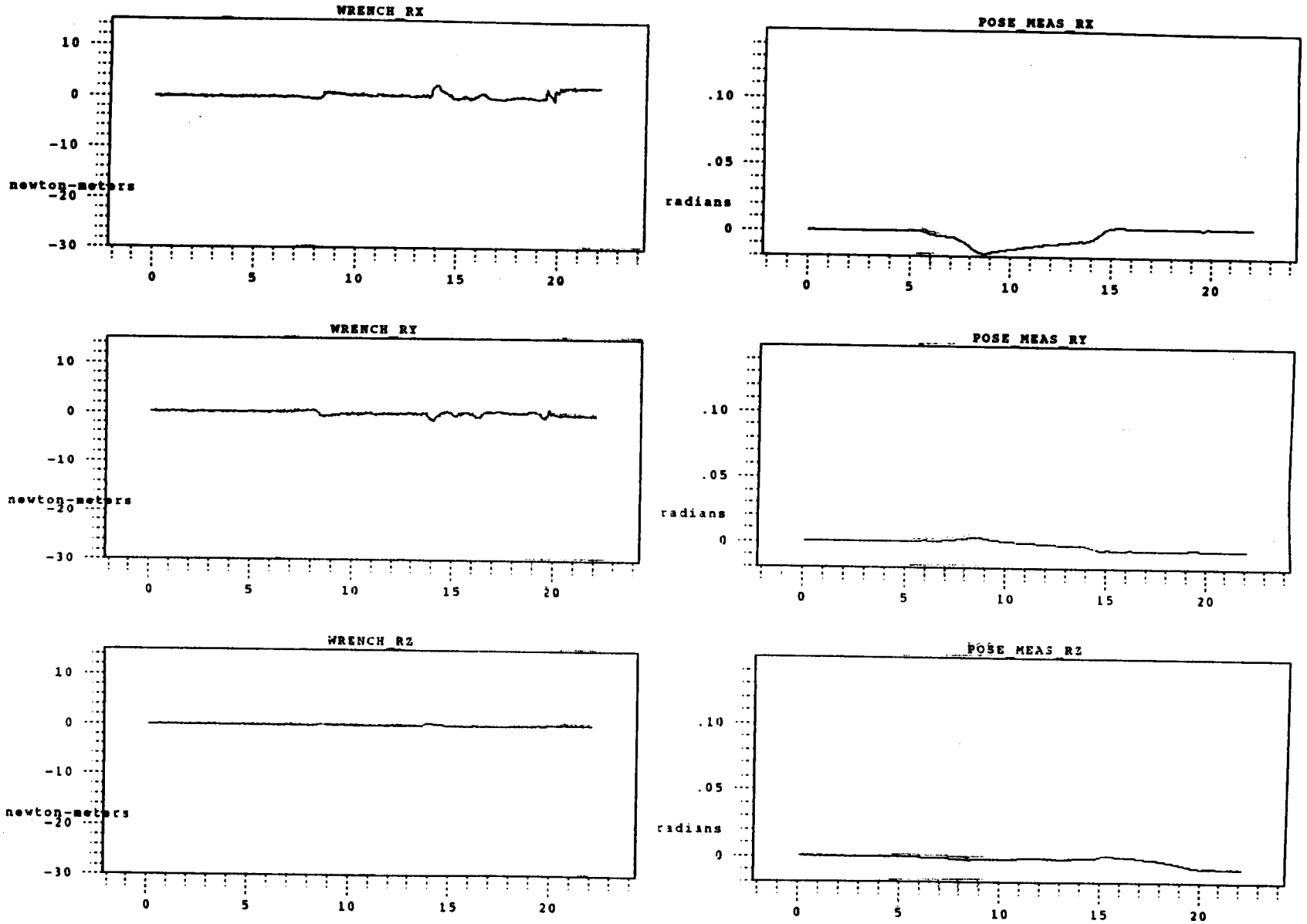


Figure 6

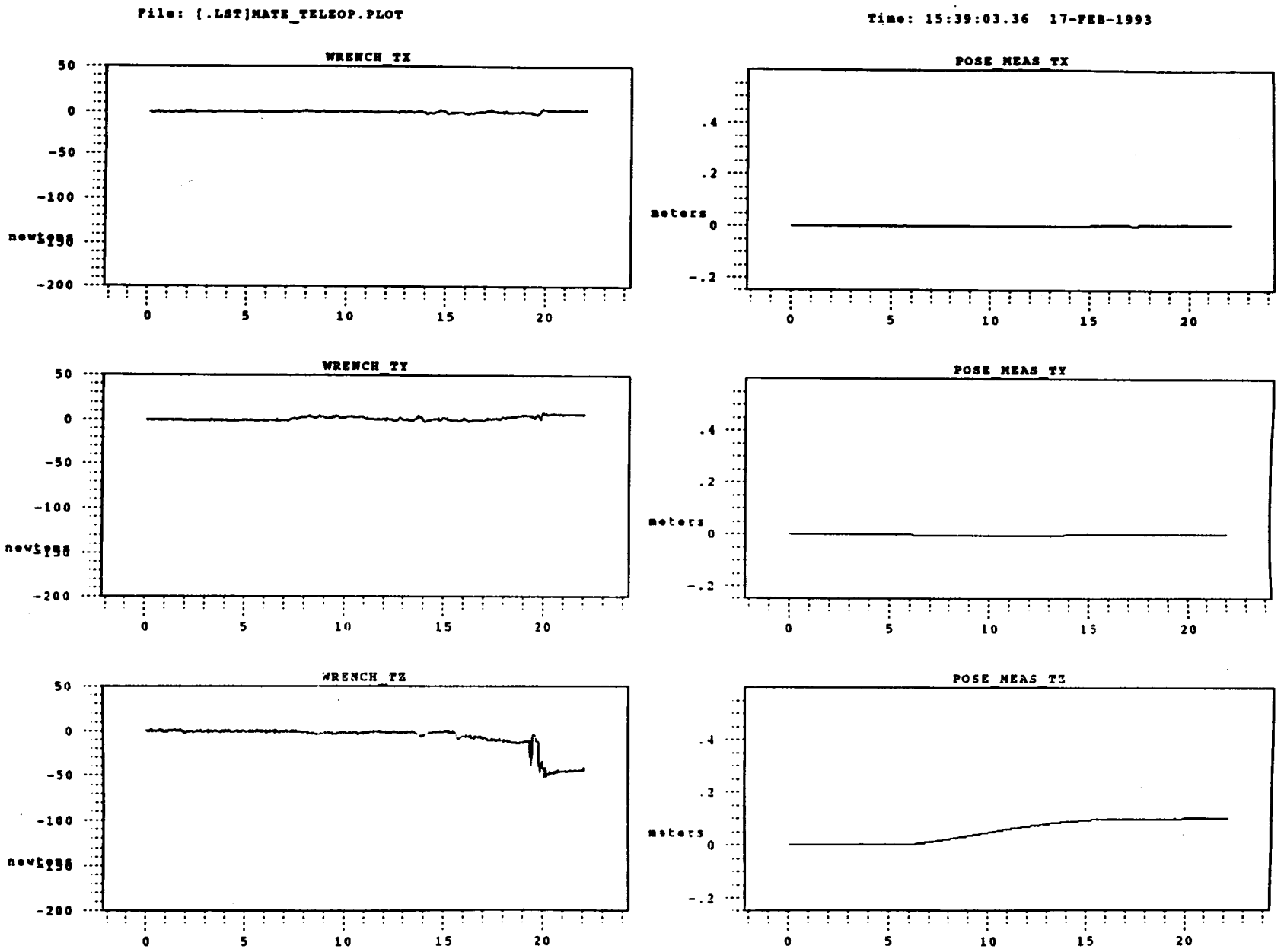

File: [.LST]MATE_TELEOP.PLOT
Time: 15:39:03.36 17-FEB-1993
WRENCH_TX
POSE_MEAS_TX
WRENCH_TY
POSE_MEAS_TY
WRENCH_TZ
POSE_MEAS_TZ
newtons
meters
50
0
-50
-100
-150
-200
.4
.2
-.2
0
5
10
15
20

figure 7

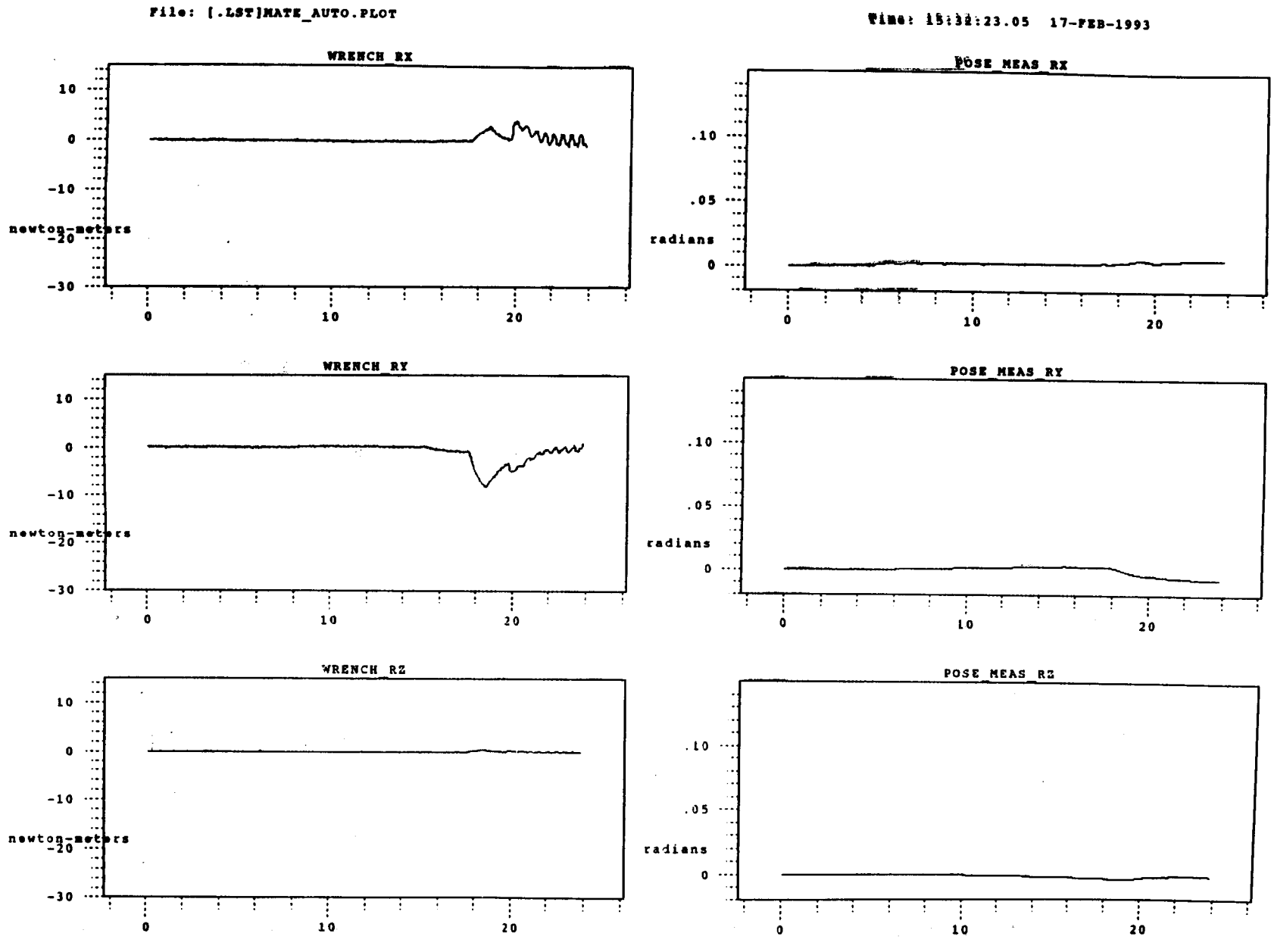

File: [.LST]MATE_AUTO.PLOT
Time: 15:32:23.05 17-FEB-1993
WRENCH RX
POSE MEAS RX
WRENCH RY
POSE MEAS RY
WRENCH RZ
POSE MEAS RZ
newton-meters
radians
10
0
-10
-20
-30
.10
.05
0
10
20

Figure 8

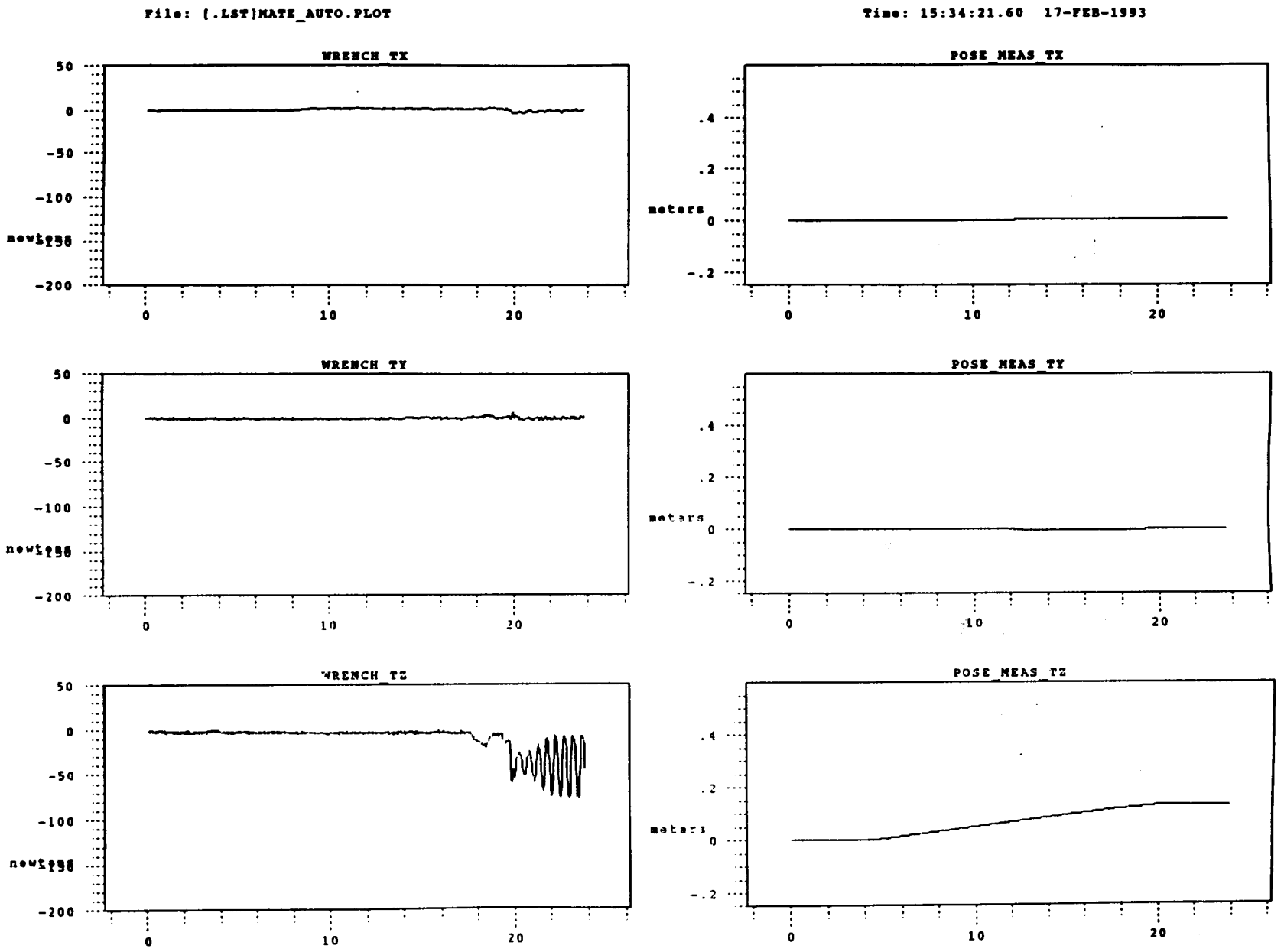

File: [.LST]MATE_AUTO.PLOT
Time: 15:34:21.60 17-FEB-1993
WRENCH TX
POSE MEAS TX
WRENCH TY
POSE MEAS TY
WRENCH TZ
POSE MEAS TZ
50
0
-50
-100
-200
.4
.2
-.2
meters
0
10
20

figure 9

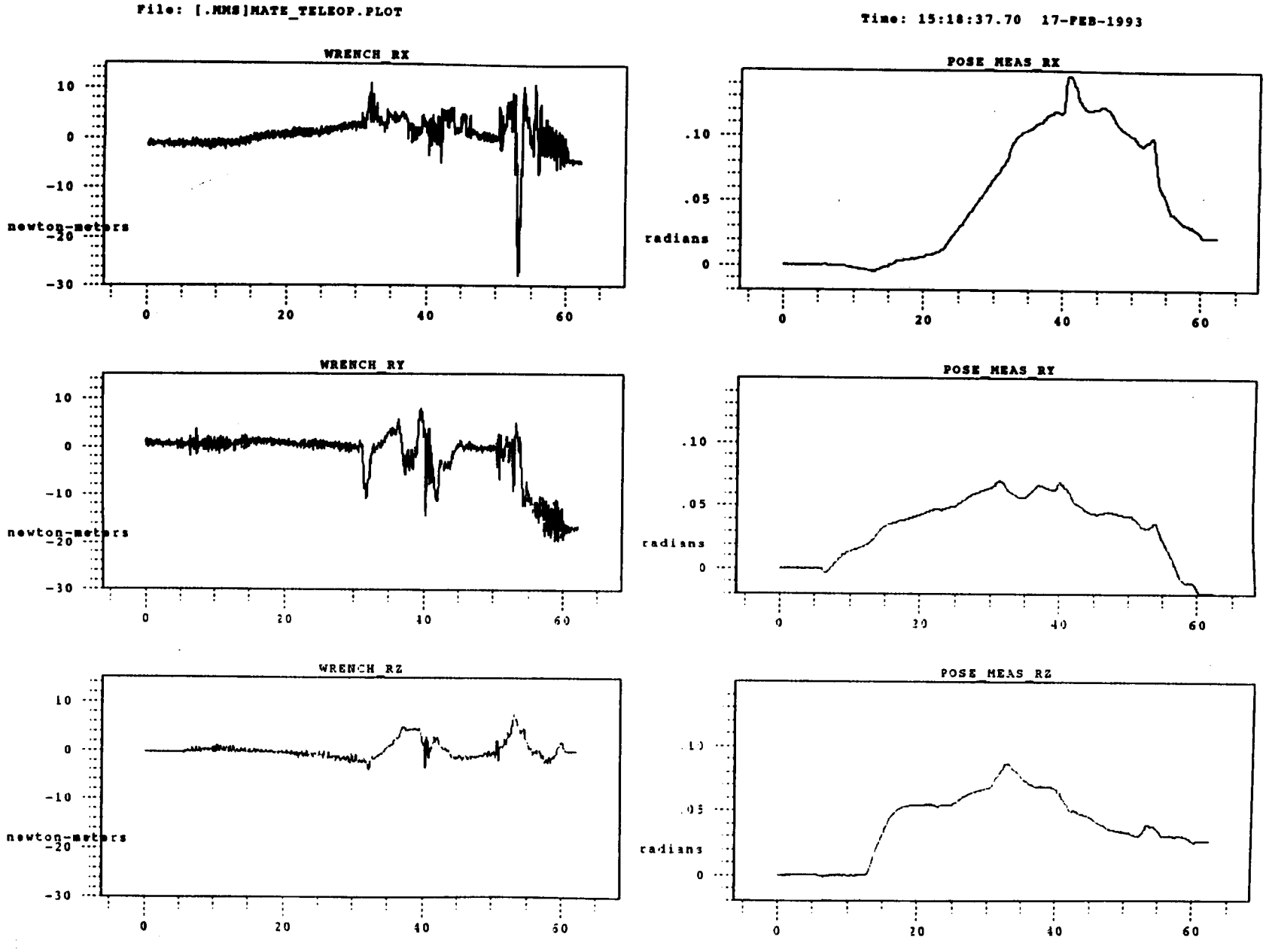

File: [.MMS]MATE_TELEOP.PLOT
Time: 15:18:37.70 17-FEB-1993
WRENCH RX
POSE MEAS RX
WRENCH RY
POSE MEAS RY
WRENCH RZ
POSE MEAS RZ
newton-meters
radians
10
0
-10
-20
-30
.10
.05
0
20
40
60

figure 10

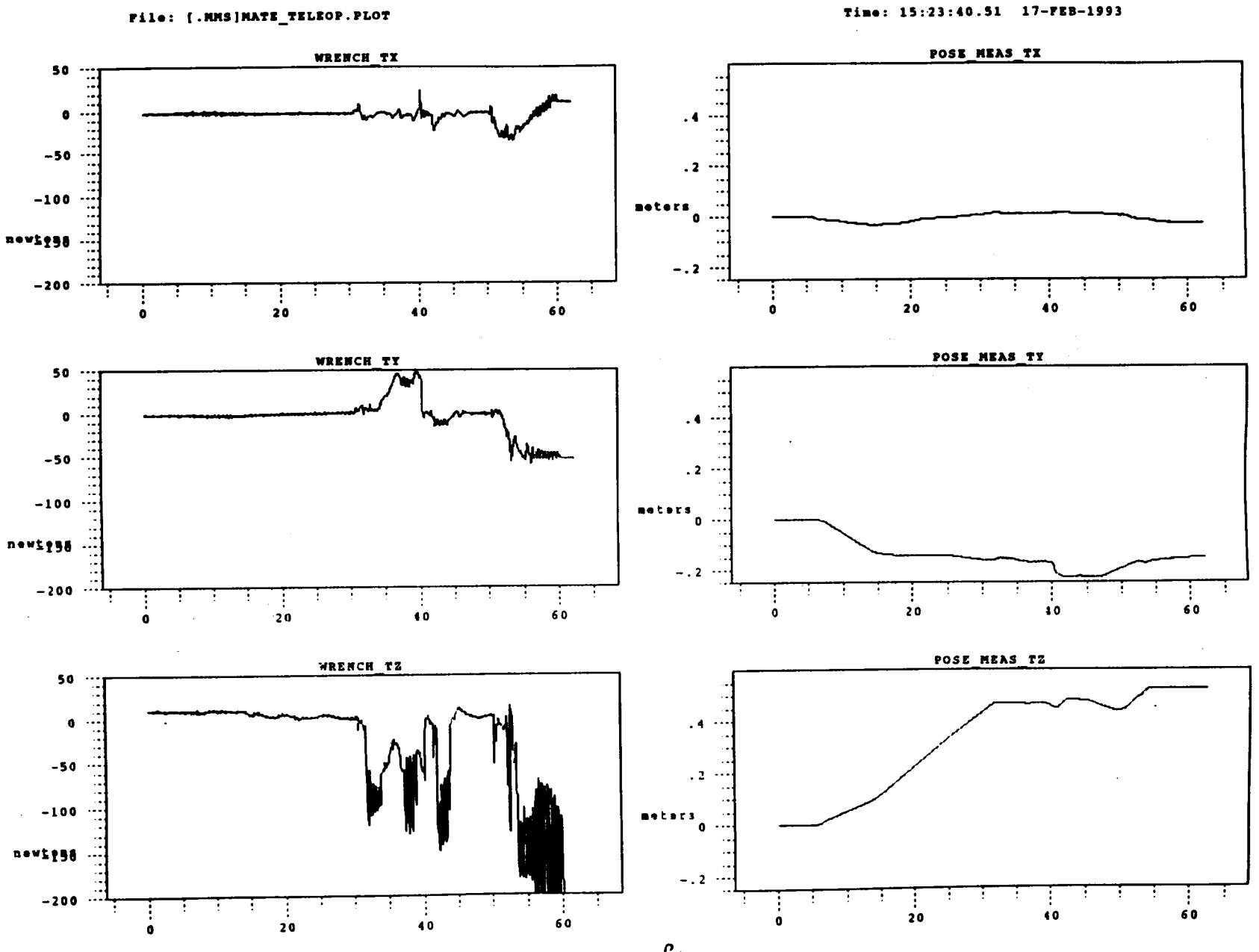

File: [.MMS]MATE_TELEOP.PLOT
Time: 15:23:40.51 17-FEB-1993
WRENCH TX
POSE MEAS TX
WRENCH TY
POSE MEAS TY
WRENCH TZ
POSE MEAS TZ
newtons
meters
50
0
-50
-100
-150
-200
.4
.2
0
-.2
0
20
40
60

Figure 11

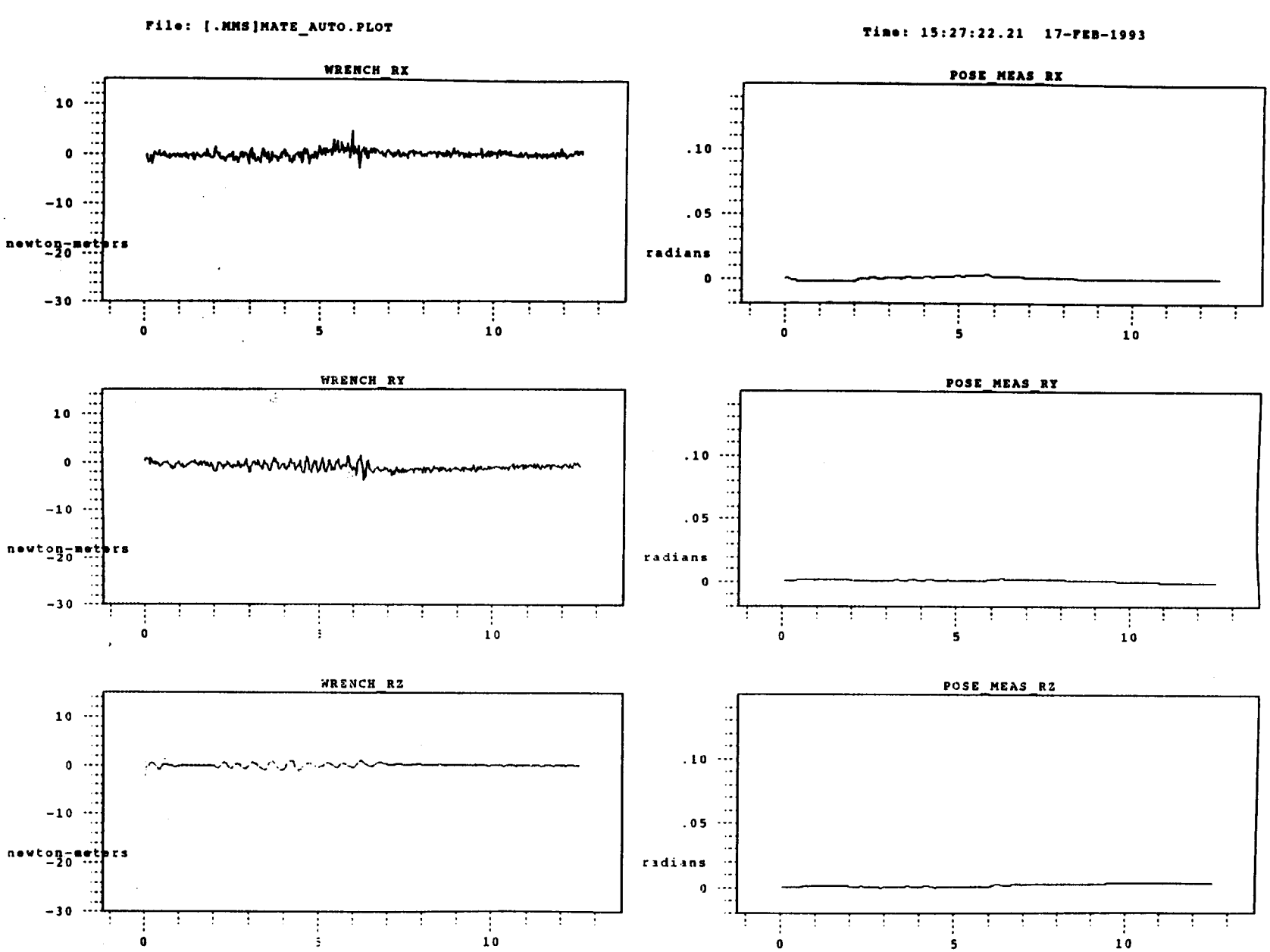

File: [.MMS]MATE_AUTO.PLOT
Time: 15:27:22.21 17-FEB-1993
WRENCH RX
POSE MEAS RX
WRENCH RY
POSE MEAS RY
WRENCH RZ
POSE MEAS RZ
newton-meters
radians
10
0
-10
-20
-30
.10
.05
0
5
10

Figure 12

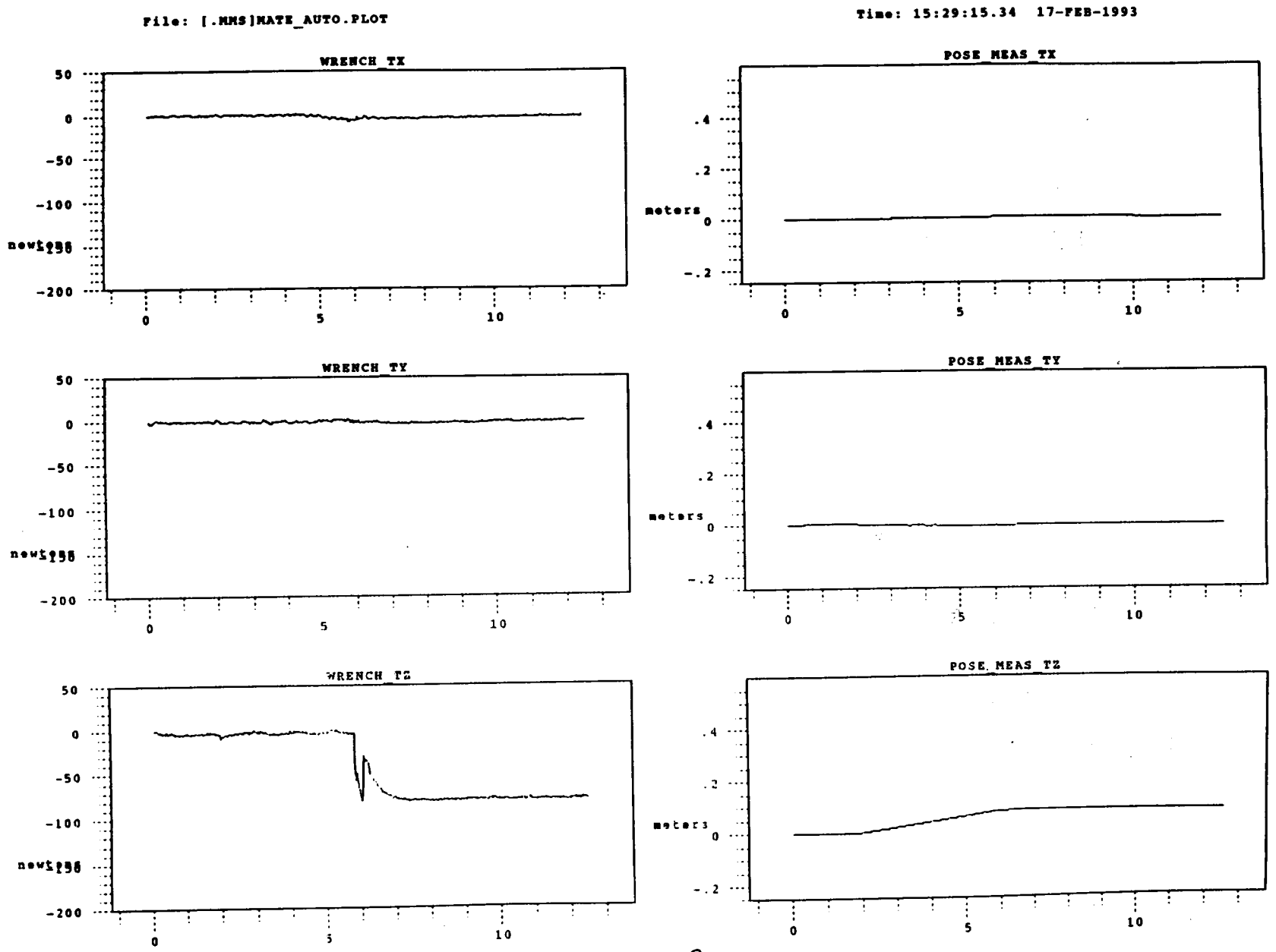

File: [.MMS]MATE_AUTO.PLOT
Time: 15:29:15.34 17-FEB-1993
WRENCH TX
POSE MEAS TX
WRENCH TY
POSE MEAS TY
WRENCH TZ
POSE MEAS TZ
newtons
meters
50
0
-50
-100
-150
-200
.4
.2
-.2
0
5
10

Figure 13

Berthing of Space Station Freedom Using the Shuttle Remote Manipulator System

Paul Cooper
NASA Langley Research Center
Hampton, Virginia

and

Martha E. Demeo
VIGYAN
Hampton, Virginia

Berthing of Space Station Freedom Using the Shuttle Remote Manipulator System

by
Paul A. Cooper
NASA, Langley Research Center
and
Martha E. Demeo
ViGYAN, Inc.

Presented March 17, 1993
at
Selected Topics in Robotics for Space Exploration Workshop
NASA Langley Research Center
Hampton, Virginia

ABSTRACT

A large-angle, flexible, multi-body, dynamic modeling capability was developed to help validate analytical simulations of the dynamic motion and control forces which occur while berthing of Space Station Freedom to the Shuttle Orbiter during early assembly flights. The paper describes the dynamics and control of the station, the attached Shuttle Remote Manipulator System, and the Orbiter during a berthing maneuver. Emphasis is placed on the modeling of the Shuttle Remote Manipulator Sytem in the multi-body simulation. The influence of the elastic behavior of the station and of the Remote Manipulator System on the attitude control of the station/Orbiter system during the maneuver is investigated.

STUDY PURPOSE AND PROCEDURE

STUDY PURPOSE: Provide independent data to help JSC validate Space Station Freedom Program on-orbit assembly simulation analyses.

STUDY PROCEDURE: Perform time simulations of the berthing of an intermediate-build configuration of Space Station Freedom to the orbiter.

A dynamic simulation of the berthing process is fairly complex since it involves the interaction of large, highly flexible components during a large motion maneuver while in orbit, where the components are subject to active control forces and gyroscopic, drag, and gravity gradient forces and moments. The complexities of the Space Station Freedom (SSF) assembly analytical simulator are such that it was advisable to develop independently a comparable tool to help validate the simulator. This paper is concerned with a description of a large-angle, multi-body, dynamic modeling capability developed to help validate the SSF program analytical berthing simulator which will be used to analyze each assembly flight.

The berthing simulations are used to calculate the dynamic motion and control forces that occur while berthing early build configurations of SSF to the Orbiter during assembly flights when attitude control of the stack resides with the station control systems. The sixth assembly flight is the first flight that will use the station control systems rather than the Orbiter Digital Auto Pilot to maintain the attitude of the stack. Berthing during this sixth flight was selected as the validation simulation since the control systems of both the station and the Shuttle Remote Manipulator System (SRMS) are active during this maneuver.

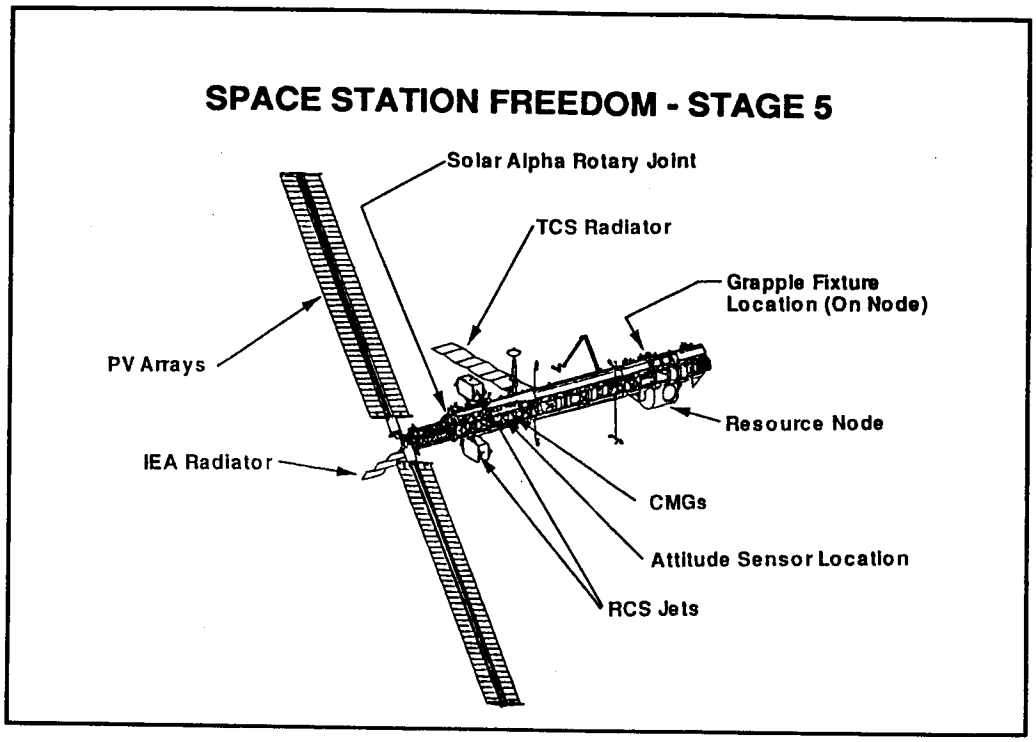
MOTIVATION FOR SELECTION OF THE SIMULATION SCENARIOS

Wanted to simulate a SSF assembly sequence which would exercise the:

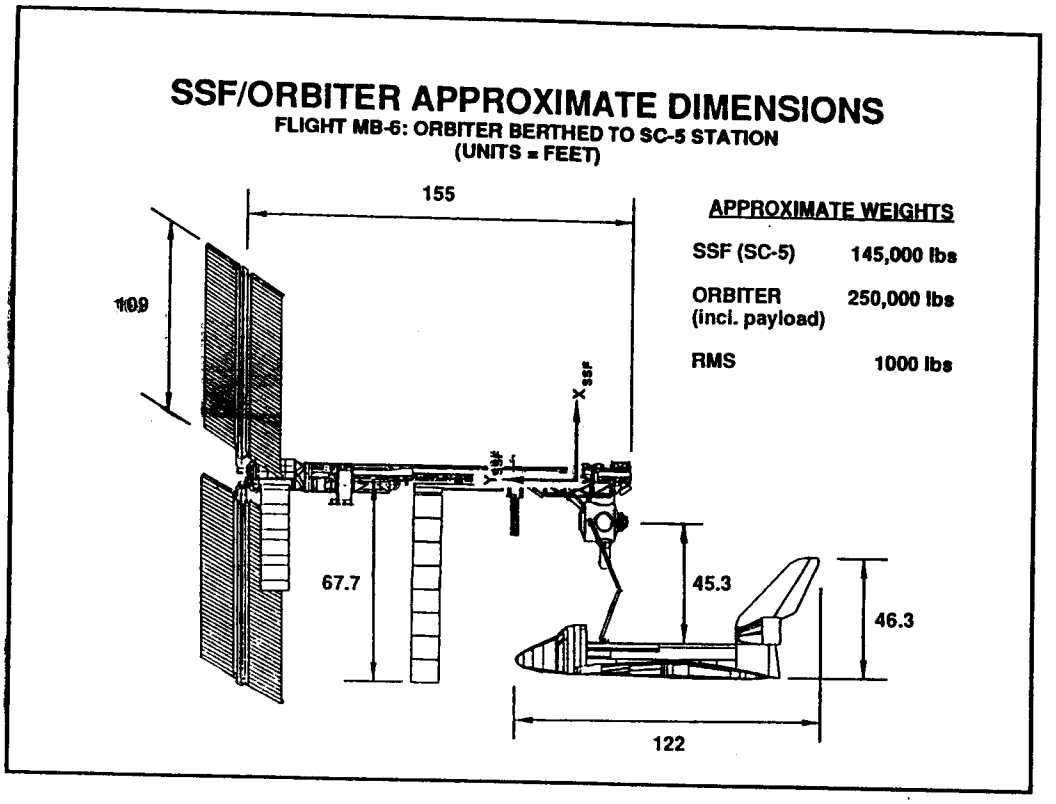
- SSF RCS attitude command system
- SSF CMG attitude and momentum management system
- Orbiter SRMS control system
- Orbiter SRMS brake system
- Orbiter SRMS flexible body
- SSF flexible body
- Environmental conditions during a typical orbit

=> Selected with JSC the berthing of SSF- Stage 5 to the Orbiter (MB-6 flight).

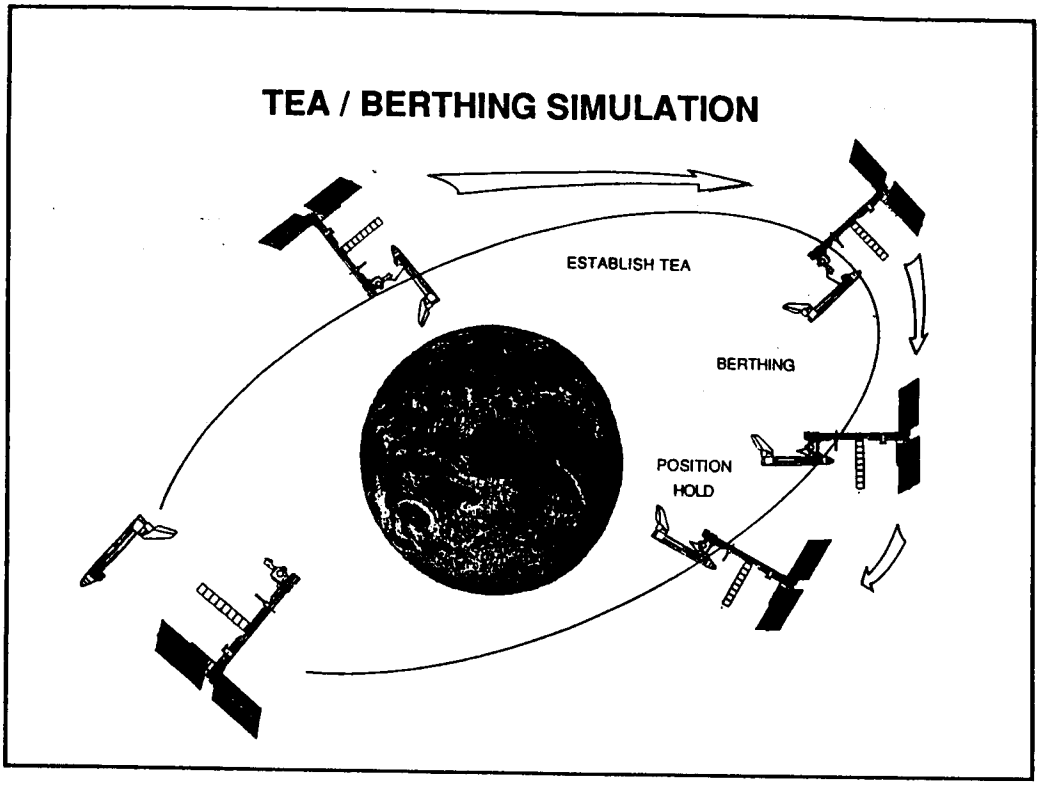
The simulation scenarios were selected to capture the critical components of the space station assembly operation when attitude control of the stack resides with the SSF. The specific features which were to be exercised are outlined above. Since the MB-6 flight was the first flight to capture all of these features, it was chosen by JSC and LaRC to be the study scenario.



The figure shows the Space Station Freedom - Stage 5 configuration. The location of the avionics platform containing sensors which provide attitude and attitude rate information is indicated. The attitude can be controlled by firing jets, located on the top and bottom of the inboard station framework, at a constant force level of 25 lbs per jet, or by a set of four double-gimbaled CMGs, each with a capacity of 3500 ft-lb-sec, located on a platform close to the avionics platform. Also shown is the resource node, a pressurized shell attached to the station framework inboard of the alpha joint, to which a grapple fixture is mounted.

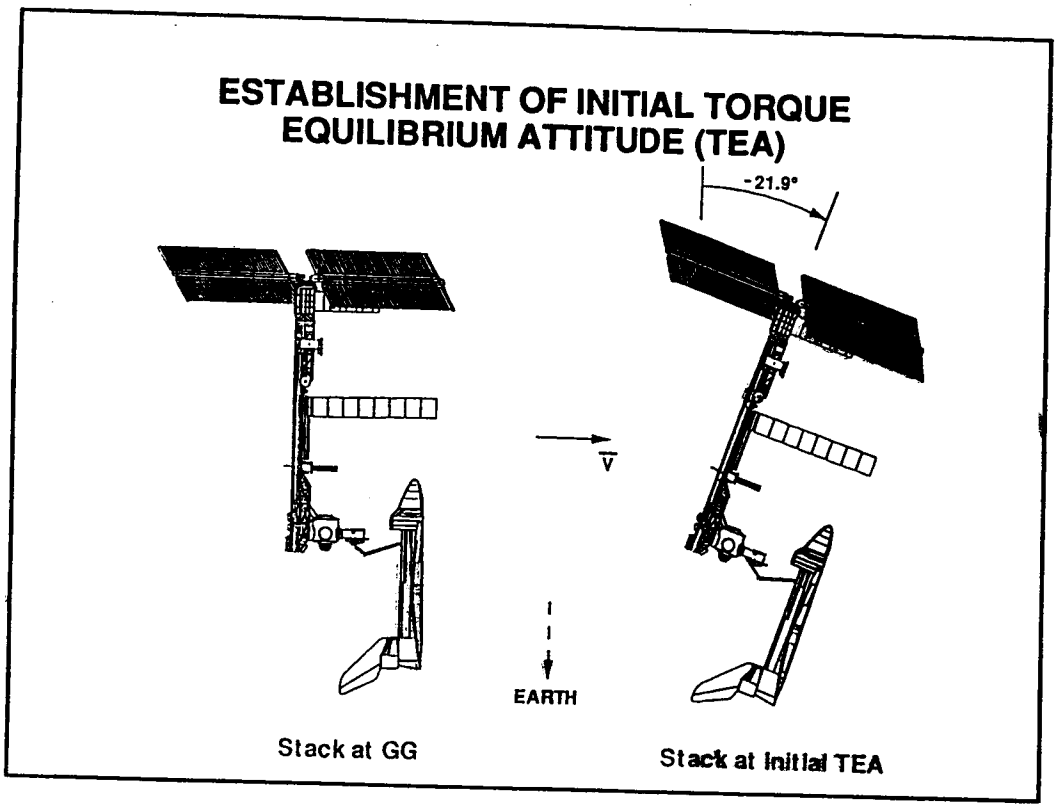


The figure shows the relative size and location of the stage 5 station, the Orbiter, and the extended SRMS at the beginning of the simulation. At this assembly stage, the SSF is over 150 feet in length. It has a weight of 145,000 lbs and the Orbiter, with the lab module in the cargo bay, has a weight of 250,000 lbs. The SRMS has a weight of only 1,000 lbs.



The scenario under investigation is the stage 5 assembly sequence depicted in the Figure. For the purposes of this study, this scenario is broken down into two simulations: 1) simulation of the Torque Equilibrium Attitude (TEA) maneuver and 2) simulation of the berthing of stage 5 to the orbiter.

Before the first simulation begins, the photovoltaic (PV) arrays are feathered and the alpha joint is locked to minimize plume loads from the Orbiter jets during the final approach of the Orbiter before grappling occurs. The alpha joint remains locked during the entire berthing maneuver. The Orbiter approaches the station along the direction opposite the orbital velocity vector and flies in tandem with the station maintaining a distance of about 30 feet from the V guides in the cargo bay to the trunnion pins on the berthing adapter. The SRMS end effector grapples the station by snaring the grapple fixture located on the resource node. This is the start of the first simulation. The SRMS joint brakes are applied and the RCS jets are fired to move the station from a GG attitude to a computed Torque Equilibrium Attitude (TEA). Once TEA has been established, the second simulation begins. The brakes on the SRMS are released, the station RCS jets are inhibited from firing, and the attitude of the stack is now maintained by the station CMG momentum management system. Berthing is accomplished by the SRMS joint motors which draw the station and the Orbiter together.



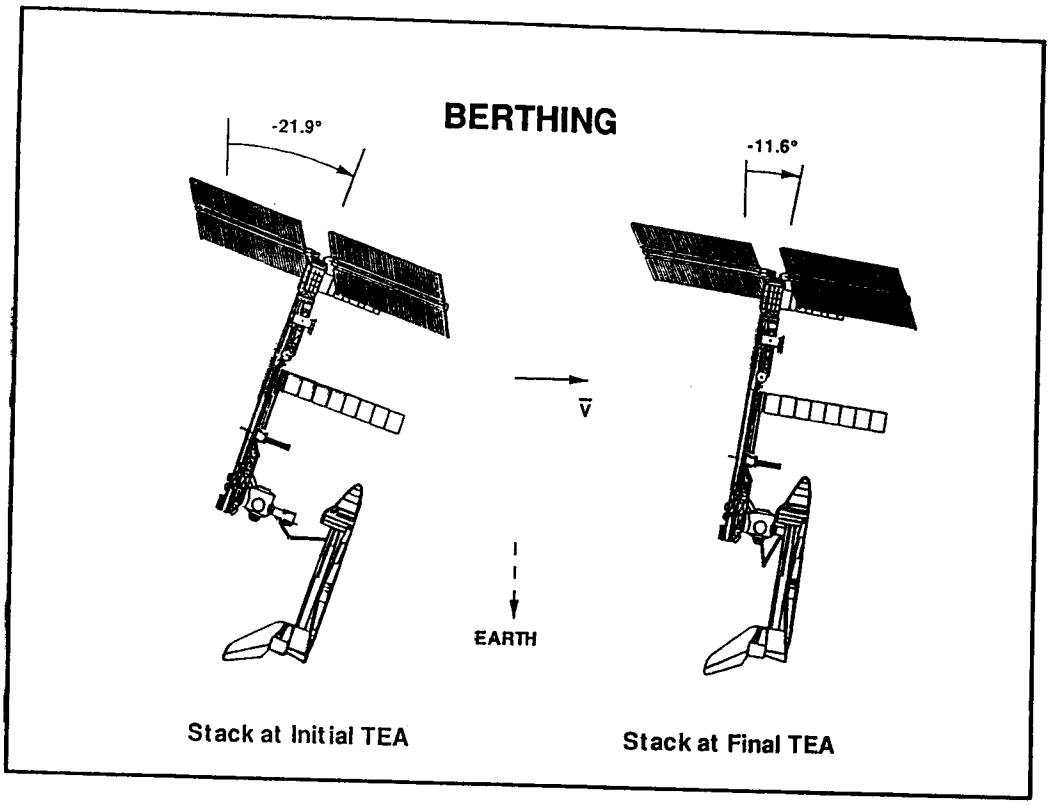
The torque equilibrium attitude (TEA) is the average attitude which must be held during an orbit so that the net angular momentum accumulated over one orbit, in the presence of gravity gradient, aerodynamic and orbital gyroscopic disturbances, is zero. The figure shows a schematic of the stack configuration at Gravity Gradient (GG) and at the initial TEA.

FIRST SIMULATION: ESTABLISH TORQUE EQUILIBRIUM ATTITUDE

Use station RCS jets to move stack from attitude at grapple to a computed TEA.

- Station RCS control system active, responding to commanded attitude.
- Brakes with friction modelled are applied to all six SRMS joints.
- Flexible representation of the SRMS long booms.
- Flexible representation of SSF (36 mode model).
- Orbital mechanics including aerodynamic and gravity gradient moments.

The specific components exercised during the simulation of TEA maneuver are outlined here. The SSF Reaction Control System (RCS) is used to maneuver the stack to its TEA. During this maneuver, the SRMS brakes are applied. The SSF is modelled as a flexible body as are the upper and lower long booms of the SRMS. Aerodynamic and gravity gradients forces and torques are included.

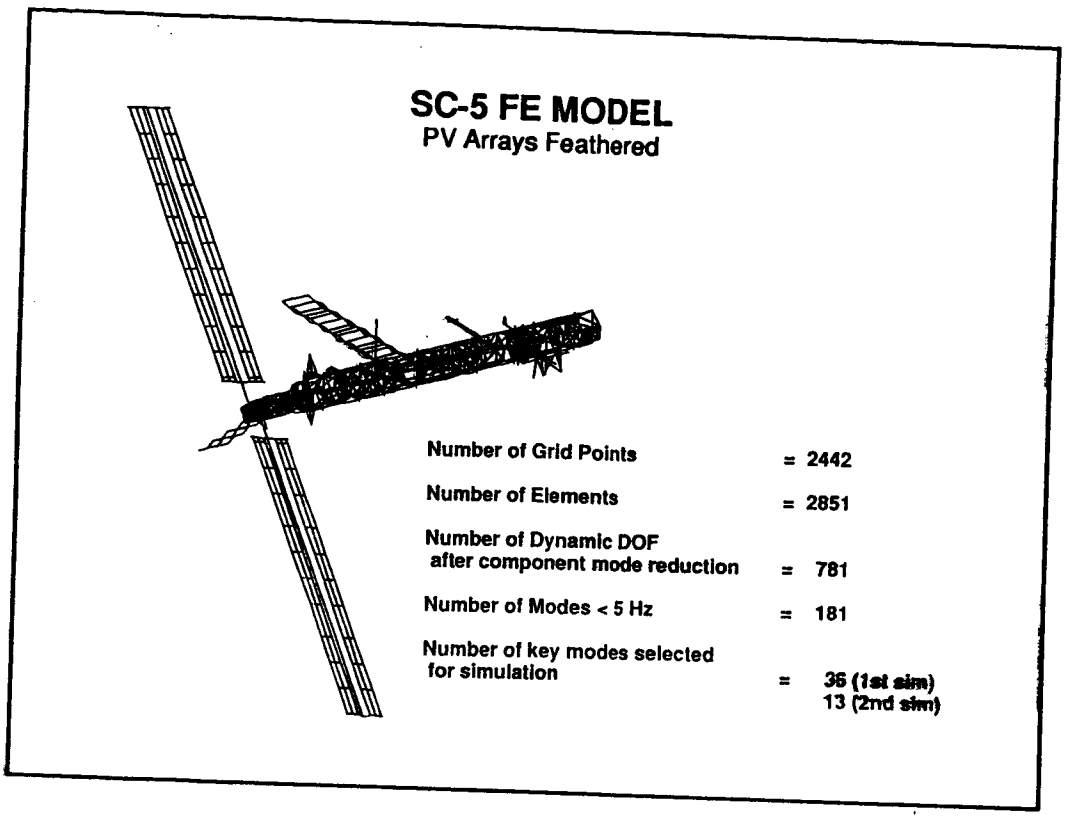


During the second simulation, the TEA changes since the inertia of the stack changes as the station is berthed to the Orbiter. The change in the TEA pitch angle during this simulation is shown in the Figure.

SECOND SIMULATION: USE SRMS TO BERTH STATION TO ORBITER

- Station CMG control system active.
- Flexible representation of the SRMS long booms.
- Flexible representation of SSF (13 mode model).
- Orbital mechanics including aerodynamic and gravity gradient moments.
- Mass matrix and estimated TEA recomputed every second.
- SRMS operational modes exercised:
  - Automatic mode
  - Manual Augmented mode
  - Position Hold mode

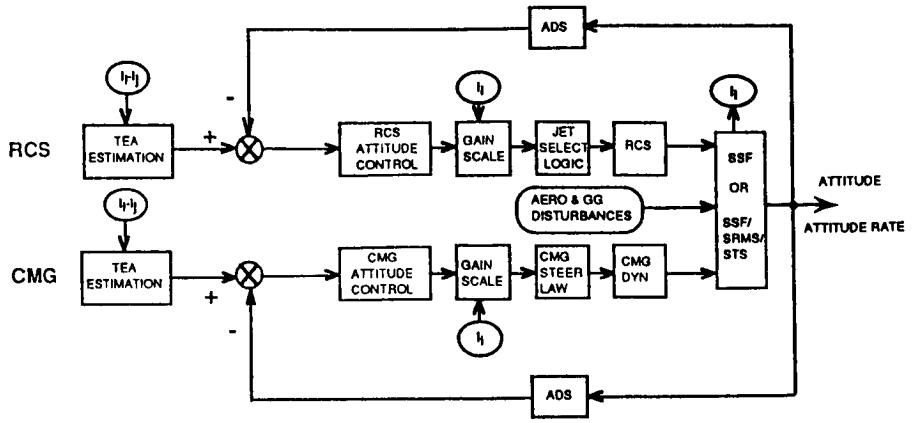
The specific components exercised during the simulation of the berthing maneuver are outlined here. The SSF Control Moment Gyros (CMGs) are used to maintain the attitude of the Orbiter/SRMS/station stack. As in the TEA maneuver, the SSF and long booms of the SRMS are modelled as flexible bodies and the aerodynamic and gravity gradients forces and torques are included. The SRMS control system is active and is used to command the SRMS to berth the station to the Orbiter using Manual Augmented and Automatic Mode maneuvers. The changes in the TEA are estimated using a mass property estimator which computes the composite inertia of the stack as a function of SRMS end effector position and attitude.



The original finite element model of SSF consisted of almost 15,000 degrees of freedom (DOF). After component mode reduction, the number of DOF was reduced to 781. Mode shapes and frequencies up to 5Hz were calculated.

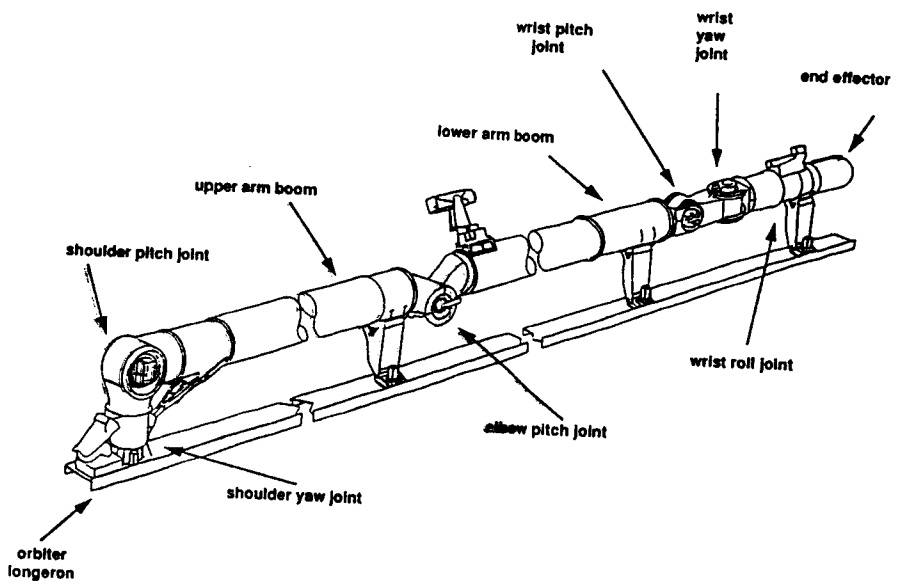
The space station structural dynamics were represented during the TEA simulation by a set of 36 natural modes which range in frequency from 0.1 Hz to close to five Hz. The modes were selected to provide an accurate representation of the flexible response at the station sensor location caused by forces applied at the RCS jet locations. The modes were obtained for the model fixed at the grapple fixture point.

SSF ATTITUDE CONTROLLER DESIGN



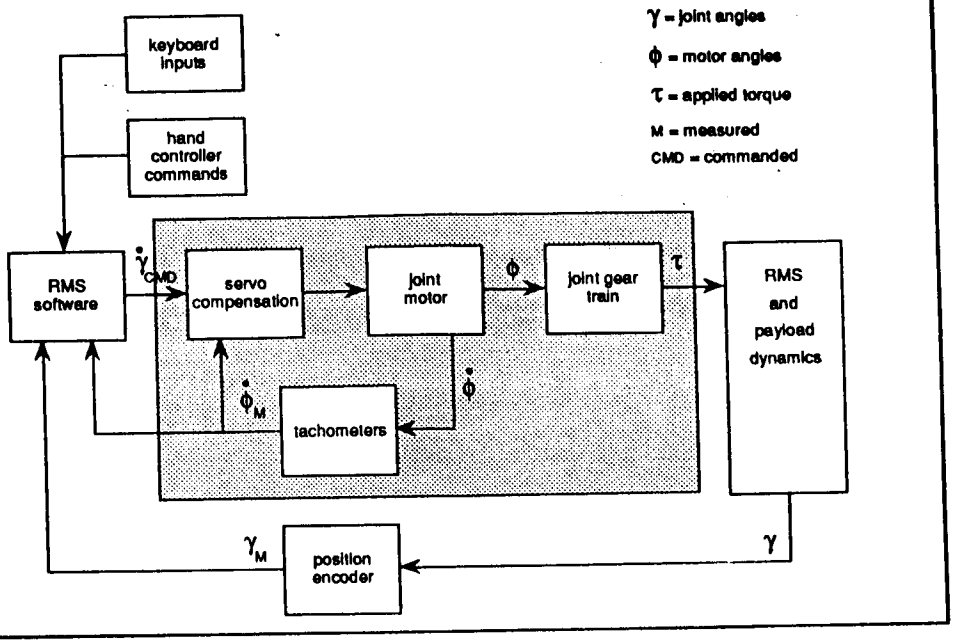
A simplified block diagram of the RCS and CMG control systems is shown. The attitude determination system (ADS), which measures the attitude and feeds this information back to the controller, is assumed to be accurate within the controller bandwidth so that a transfer function of unity is assumed for the ADS for the current simulations. The control system is designed for use in all configurations of the station covering a large range of inertias during the 3-year assembly process. To accommodate this wide range of system parameters, a mass estimator is provided to determine the on-orbit inertias and to adjust the control gains for acceptable performance. Normally the gains will not change significantly for a given flight configuration since the inertia matrix will remain nearly constant until the next assembly flight; however, during the berthing process, the system inertia matrix is continuously changing so that the gains are also continuously changing. A mass estimation program based on knowledge of the SRMS end effector location has been written to provide an updated system inertia matrix as berthing progresses. There are two bending filters (low pass filters designed to remove higher frequency components of the feedback position and rate signals) in the control design.

SHUTTLE REMOTE MANIPULATOR SYSTEM



The Shuttle Remote Manipulator System is a six-joint anthropomorphic arm which was originally designed to deploy payloads weighing up to 65,000 lbs. and retrieve payloads weighing up to 35,000 lbs. The 50 foot arm is mounted to the port longeron of the Orbiter cargo bay by means of a Manipulator Positioning Mechanism (MPM). This is the so-called swing-out joint which is used to rotate and lock the arm (19.48°) outboard for adequate clearance during arm/payload operations. From this attachment point, the arm is comprised of two single degree-of-freedom (DOF) shoulder joints, a 21 foot long upper boom, a single DOF elbow joint, a 23 foot long lower boom, 3 single DOF wrist joints and a snare type end effector capable of mating with a payload mounted grapple fixture.

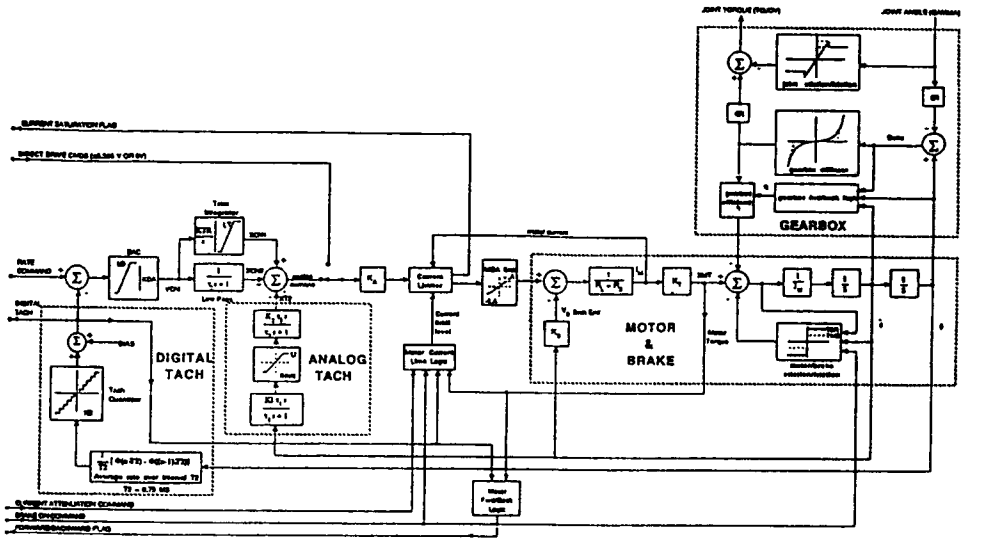
MOTOR SERVO CONTROLLER FOR EACH SRMS JOINT



Each of the six SRMS joints is comprised of a reversible dc drive motor, a mechanical joint brake, an inductosyn tachometer, an epicyclic gear train and an electro-optical encoder and servo compensation as shown. The SRMS is telerobotically controlled from the aft cockpit of the Orbiter by way of translational and rotational hand controllers and control panel command inputs. Joint rate commands are sent from software algorithms resident in the Orbiter General Purpose Computer (GPC) to the joint servos by way of the Manipulator Control Interface Unit (MCIU) (not shown). The joint gear train applies the required torque to the SRMS/payload system. The encoders and tachometers provide measurement of the joint position and rate, respectively.

DETAILED SERVO MODEL

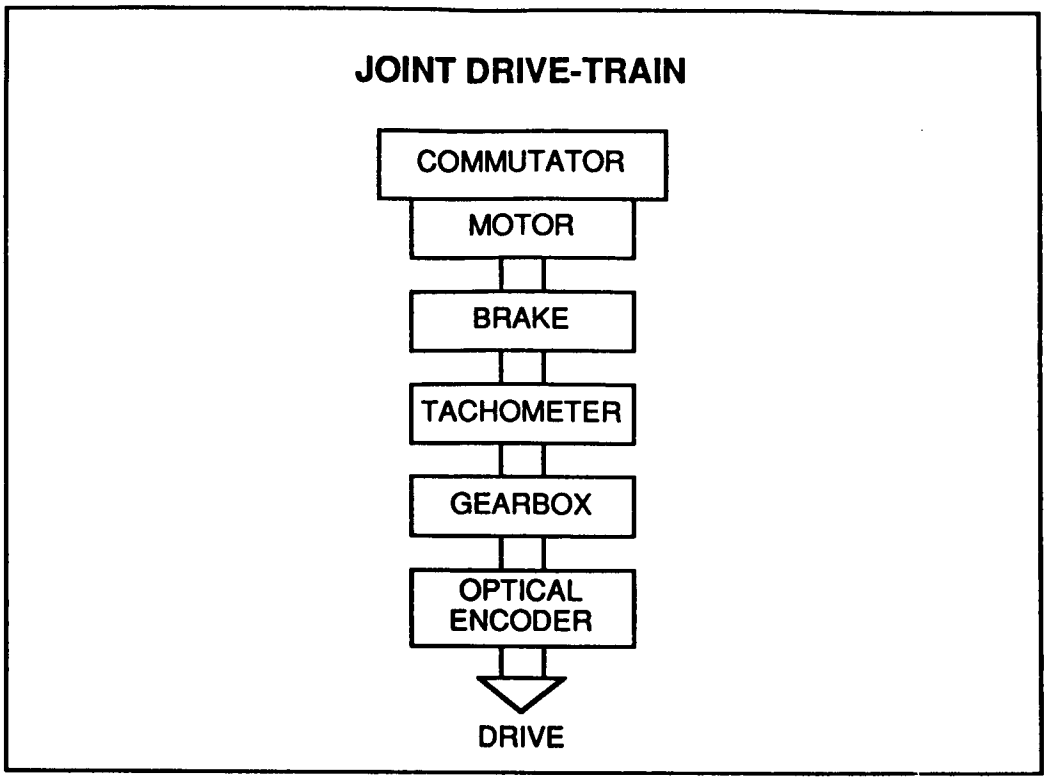
DRAPER RMS SIMULATION / JULY 1988



This a detailed block diagram of the servo system modelled in the assembly simulator. The servo math model was adapted from an existing high fidelity simulation tool of the SRMS called the DRS, Draper RMS Simulation.

As shown, a digital joint rate command (in counts) is received from the GPC by way of the MCIU. This input rate demand is compared with the actual motor rate from a digital tachometer feedback to form an error signal. This error is then converted to an analog voltage signal and processed in through a trim integrator and a low pass filter. The purpose of the integrator is to provide a high gain at low frequencies needed to break motor and gear train stiction and to overcome small errors. The output of these analog electronics are summed with negative feedback of the analog tachometer signal. This continuous part of the tachometer is run through a high pass filter which serves to provide stability. In contrast, the purpose of the digital tachometer is to improve tracking accuracy. The analog voltage signal is then sent to the Motor Drive Amplifier (MDA) and a current limiter which results in attenuation of the voltage applied to the joint motor. The resulting motor rate is then passed to the gearbox to generate the required output torque.

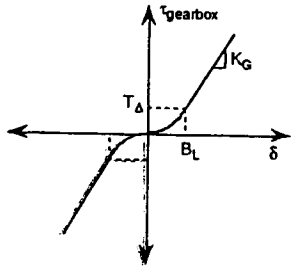
Nonlinear friction losses are modeled on both the motor or input side of the gearbox and on the output or joint side of the gearbox. The output friction models include both the joint friction and the gearbox friction.



Shown is a schematic of the SRMS joint drive-train. The optical encoder is physically mounted on the joint side of the gearbox and the tachometer is mounted on the motor side of the gearbox. The friction and freeplay which occurs in the drive-train have been modelled in the simulation.

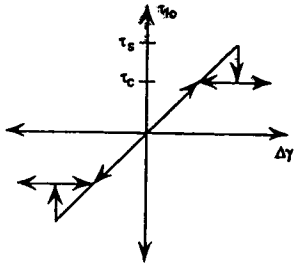
NONLINEAR SERVO MODELS

Gearbox Stiffness



where,
T_Δ = gearbox transition torque (ft-lbs)
B_L = gearbox transition angle (rad)
K_G = gearbox linear stiffness (ft-lbs / rad)

Joint Friction/Stiction



where,
Δγ = difference in actual joint angle (deg)
τ_c = coulomb torque level (ft-lbs)
τ_s = stiction torque level (ft-lbs)

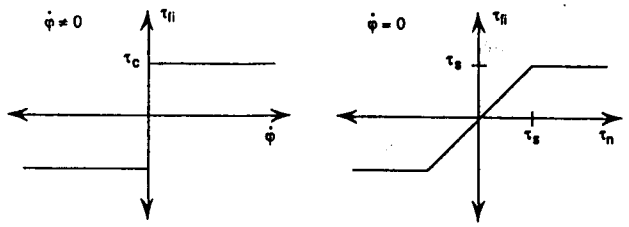
"the old hairbrush model"

The nonlinear gearbox model is represented by an asymptotic linear compliance and a quadratic stiffness relation at low torque levels. This stiffness is computed as a function of the backlash angle of the gearbox as shown.

For the joint friction/stiction model, the friction torque is computed as a function of the joint angle as shown. When the actual joint angle is less than a stiction (static friction) reference angle, friction torque acts like a spring restraint. In this region, a steady torque produces a small rotational displacement. When an applied torque is removed, the joint returns to an equilibrium position. If the displacement exceeds the stiction reference angle, the friction torque drops to a coulomb torque level. While the joint rate remains positive, the friction torque is constant. If the rate becomes negative, the friction reference angle is reset and the model reverts to its spring-like behavior about the new reference point. This is the so-called "old hairbrush" model used in the DRS.

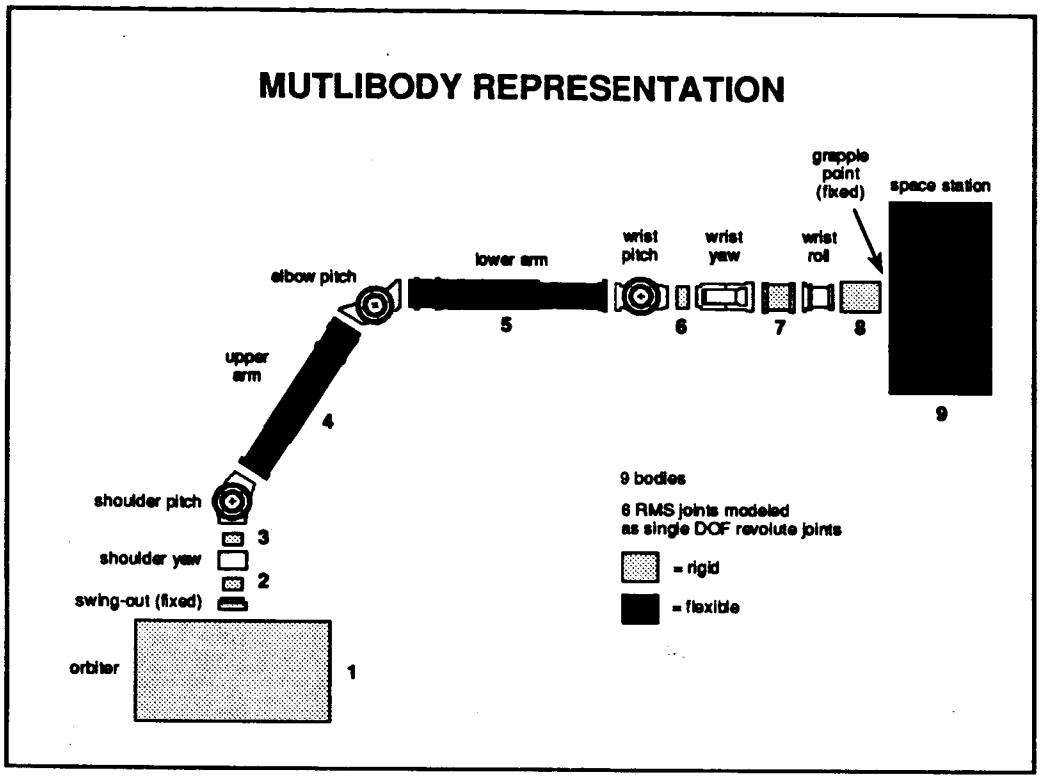
NONLINEAR SERVO MODELS (cont'd)

Motor/brake Friction



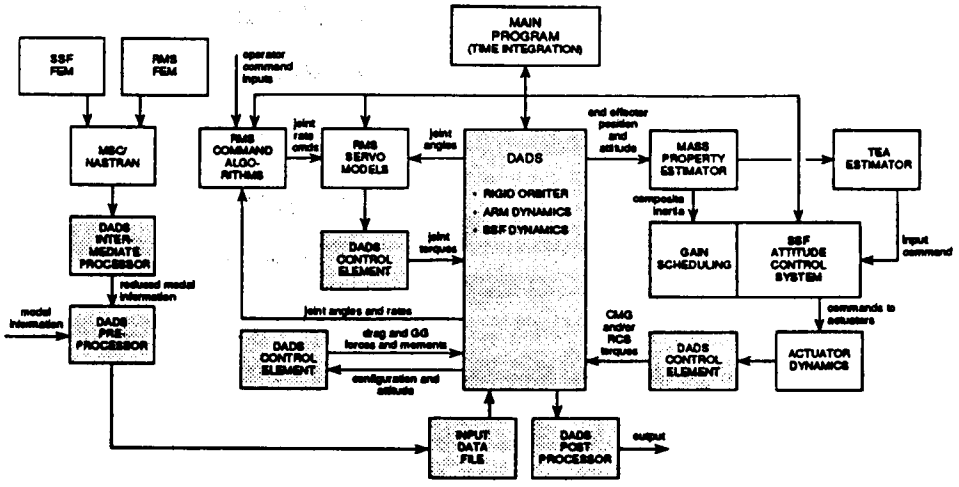
where,
$\dot{\varphi}$ = motor rate (rad/sec)
τ_C = coulomb torque level (ft-lbs)
τ_S = stiction torque level (ft-lbs)

The SRMS motor/brake friction/stiction is modelled as shown. When the joint is moving, the friction torque is equal to the coulomb (sliding) friction value. When the joint is not moving and the torque is larger than the stiction level, the friction torque is set to the stiction level. If the torque is less than the stiction level, just enough stiction is applied to the joint to make the net torque output zero.



Nine bodies are used to model the complete multibody system as shown. The nine bodies include the orbiter, the seven links of the SRMS, and the SSF. Three components, the two long booms of the SRMS and SSF, are modeled as flexible bodies. Eight joints are defined to connect each of the bodies in the system. The swing-out joint at the base of the SRMS and the connection between the end-effector and payload are modeled as bracket (rigid) joints. The remaining six joints are modeled as single degree-of-freedom revolute joints, which accommodate the six degrees-of-freedom of the SRMS.

SIMULATION ARCHITECTURE



The detailed architecture of the computational tool is shown. It consists of four major parts; multibody dynamics code DADS (Dynamic Analysis and Design System), the SRMS controller, the SSF ACS, and the MAIN program. The DADS code is used to generate equations of motion of the system, including the SRMS arm, the orbiter, and SSF. Each of these modules has its own integration routines and integrate its state equations at its respective integration step sizes. In order to synchronize the simulation process of different sampling rates, the MAIN routine was added to control the timing and program execution flow.

For the SRMS controller, joint angles and rates from DADS, along with operator command inputs, are fed into the SRMS command algorithm to compute joint rate commands. The SRMS controller model calculates driving torques, based on the joint rate commands, which are then applied back to DADS through control elements. For the ACS, the DADS code provides the attitude and attitude rate of the stack to the ACS. Along with the commanded attitude, the ACS computes attitude errors and rate errors that are used to compute required commanded torques to be applied to the system. At the same time, the mass property estimator is used to estimate the inertia of the composite system. This estimated information is used to compute proper gain scheduling in the ACS and to update the commanded attitude. Depending on the type of actuator used, the commanded torques are converted to either RCS forces or to CMG torques which are fed back to DADS using control elements. Environmental disturbances from aerodynamics moments and gravity gradient torques, recomputed on the estimated inertia change, are also applied to the system through control elements.

SIMULATION RESULTS

All sims performed using an SGI 4D/440 workstation

ESTABLISH TORQUE EQUILIBRIUM ATTITUDE

36 mode representation of SSF

4 mode representation of SRMS

TEA was established in 20 minutes real-time

BERTHING

13 mode representation of SSF

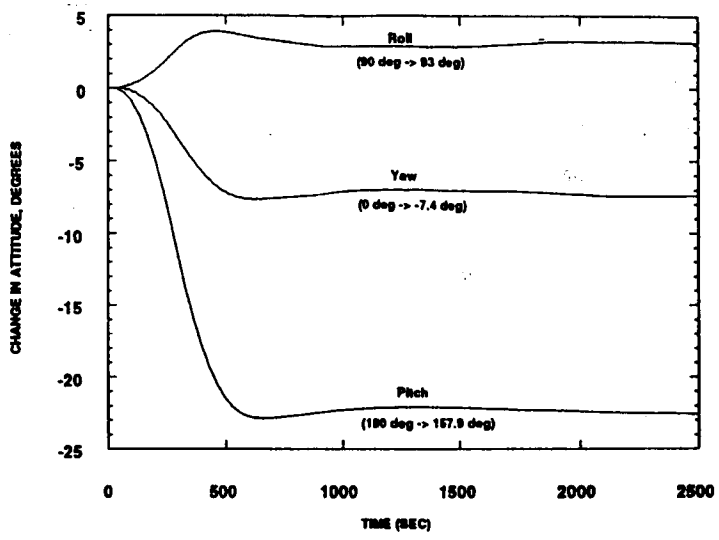
4 mode respresentation of SRMS

Berthing to within 2 feet was completed in 14 minutes real-time.

For the TEA maneuver, the time integration was performed with a time step of 0.001 seconds and the computations took approximately 72 hours of dedicated CPU time on an SGI 4D/440 workstation. The station structural dynamics were represented using 36 normal modes. A comparison was made with a simulation using ten normal modes to represent the station dynamics. The response differences were within the accuracy of the computation and thus the ten-mode model was deemed to be sufficient in representing the dynamics of the station. A conservative proportional damping level of 0.2 percent of critical damping was assumed for each mode.

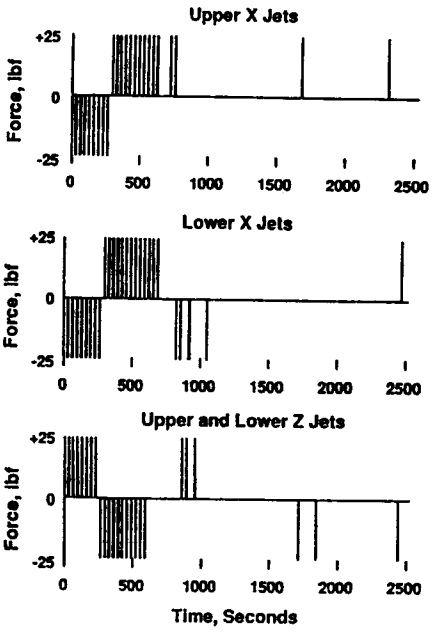
For the Berthing maneuver, the time integration was performed with a time step of 0.002 seconds and the station dynamics were represented using a 13 mode model. The maneuverwas completed in 14 minutes.

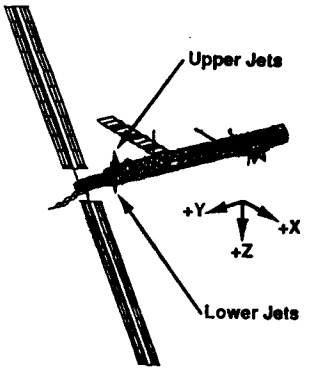
ATTITUDE CHANGE DURING TEA MANEUVER



The TEA maneuver required an attitude change from a gravity-gradient position to an attitude orientation of pitch, yaw and roll of -22.1°, -7.4° and 3°, respectively. The resulting attitude-change time history is shown. The TEA of the stack was successfully established within 1200 seconds, a little more than a quarter of an hour.

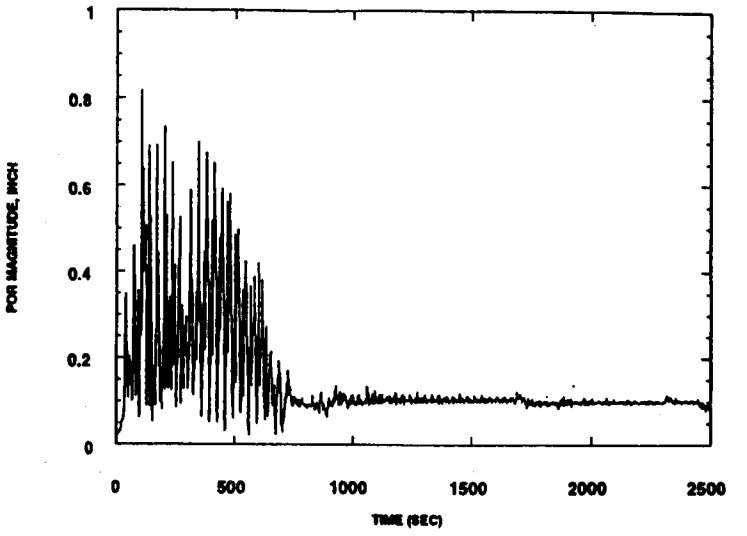
RCS JET FIRINGS DURING TEA MANEUVER





The actual jet firing times for the upper and lower x-axis and z-axis jet clusters are shown. The RCS commanded torque is realized through the jet selection logic as firing pulses of approximately 0.2 to 1 second in duration. The RCS jets are inhibited from firing more often than once every 33 seconds to reduce structural dynamic response.

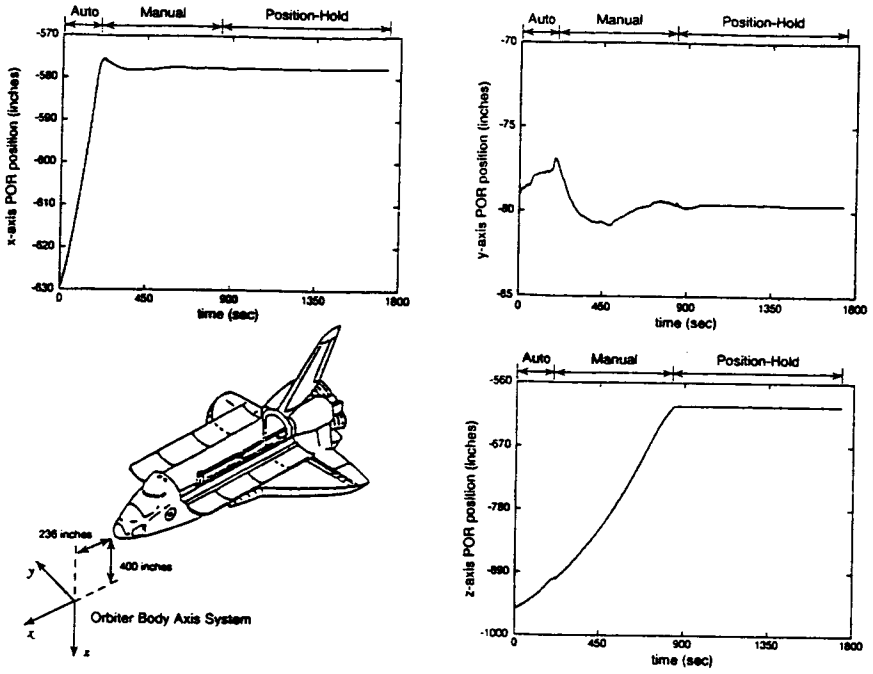
CHANGE IN POR MAGNITUDE DURING TEA MANEUVER



Shown is a plot of the relative magnitude of the position of the Point of Resolution (POR) (located at the SRMS end-effector) with the brakes engaged. The SRMS joints were slightly overloaded when the jets were fired to establish TEA and two of the wrist joints exhibited some brake slip during the first 750 seconds of the simulation. The largest position change computed during the jet firings was less than one inch and the resultant slip after the jet firings were completed was less than 0.2 inches. The total position change is a combination of brake slip, arm flexibility and joint flexibility in the six-joint drive-trains.

POR LOCATION FOR BERTHING SCENARIO

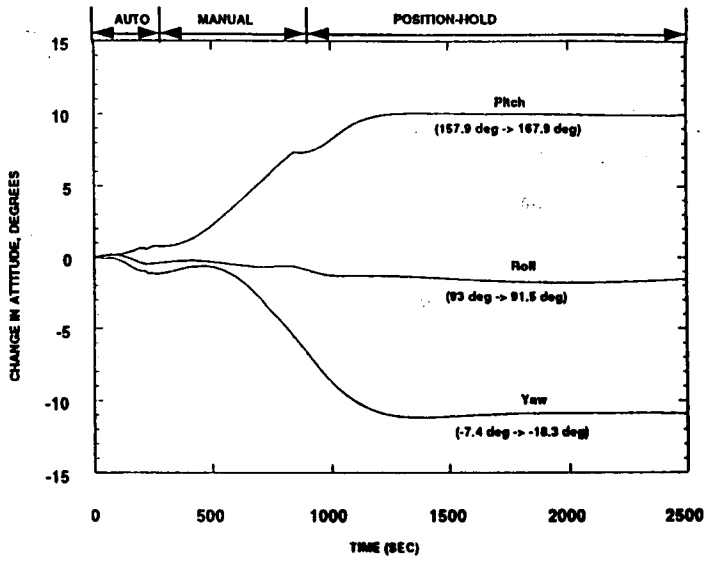
Orbiter Body Axis System (OBAS) Coordinates



Results from the berthing simulation are shown. The SRMS POR is commanded to move along a three-point berthing trajectory. From the initial grapple point, position 1, the SRMS is commanded to move 4.5 feet in the x and z-axis directions and 2 inches in the y-axis direction using an Operator Commanded Auto Sequence (OCAS) maneuver. From position 2, the SRMS is immediately commanded to move 25 feet vertically to position 3 using Manual Augmented Loaded mode z-axis translational hand controller inputs. Position 3, which is two feet from a full berthed position, is reached in approximately 800 sec. At this time, the Position Hold function is automatically enabled by the SRMS command algorithms to maintain its commanded position and attitude. Very little residual vibratory motion is observed following completion of the berthing maneuver.

This simulation was conducted using translational and rotational end-effector rate limits of 0.14 ft/sec and 0.14 °/sec (coarse mode).

ATTITUDE CHANGE DURING BERTHING MANEUVER



The inertia of the stack, and thus the TEA, changes continuously during the berthing maneuver. The instantaneous mass of the stack is computed every second and used to modify the gains of the CMG momentum management control system. This information is also used to compute the current TEA which is subsequently applied to the CMG control system as an updated commanded attitude change. The change in attitude of the stack during the berthing maneuver is shown. The final Pitch, Yaw and Roll attitude of the stack is designated.

SIMULATION OBSERVATIONS

ESTABLISH TORQUE EQUILIBRIUM ATTITUDE

Minor SRMS brake slip occurs (continuous firing unacceptable).

Influence of structural response on control is small.

Structural response and loads are small.

RCS control functioned well.

BERTHING

CMGs saturate if maneuver starts with unbiased CMGs.

SRMS Position Hold mode shows tendency toward instability (SRMS upgrades not implemented).

Influence of structural response on control is negligible.

Periodic update of stack inertia is required (trace of I matrix changes by as much as 10% during berthing).

The following observations may be made about the simulated TEA and berthing maneuvers. The SSF RCS successfully established a TEA for the stack with only a small amount of SRMS brake slip. During the berthing maneuver, the CMGs were able to track the changing TEA while the orbiter and station were pulled together by the SRMS when given an initial bias.

Although not shown, the Position Hold mode did exhibit a tendency for instability when left on for an extended period of time. This may be attributable to the known instability of Position Hold with massive payloads. An enhancement to Position Hold mode along with two other SRMS upgrades are presently being implemented by JSC to facilitate assembly operations.

In both simulated maneuvers, the elastic behavior of the station and of the SRMS was found to have only a minor influence on the attitude control of the stack and the control loads caused only minor internal structural loads and structural response.

LaRC/MSFC Loosely Coupled Multibody Spacecraft Controls Research Facility

Raymond C. Montgomery
NASA Langley Research Center
Hampton, Virginia

and

Dave Ghosh
Lockheeed Engineering and Sciences Company
Hampton, Virginia

LaRC/MSFC Loosly Coupled Multibody Spacecraft Controls Research Facility

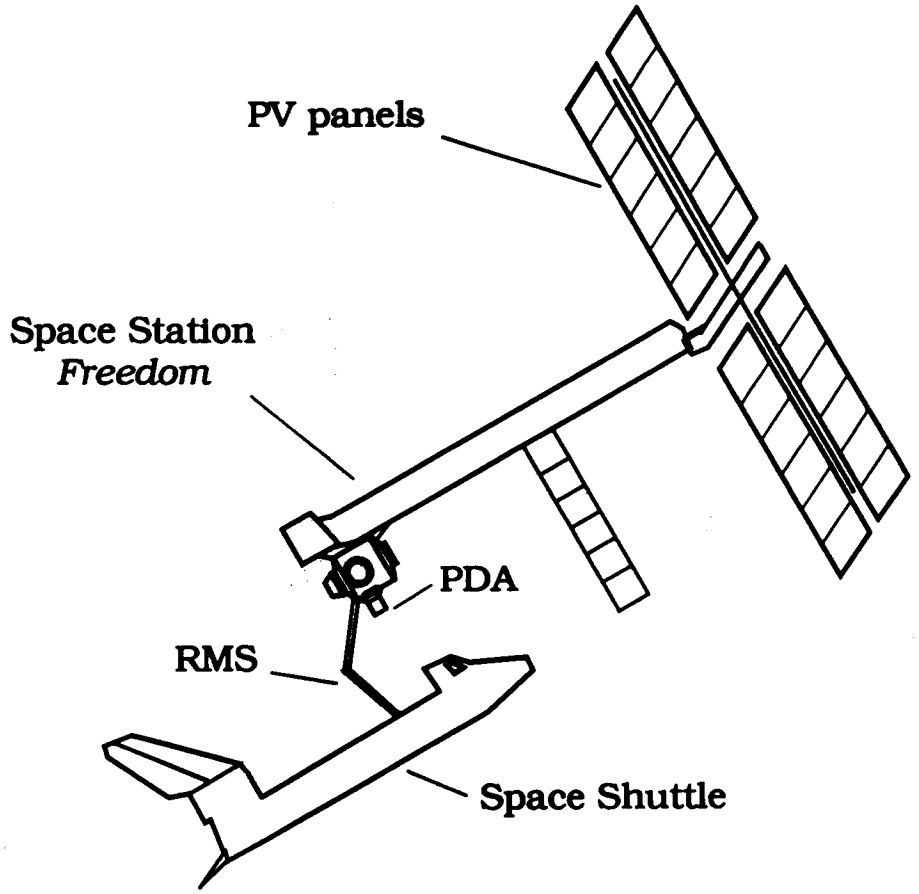
Raymond C. Montgomery and Dave Ghosh

SPACECRAFT CONTROLS BRANCH
NASA LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA

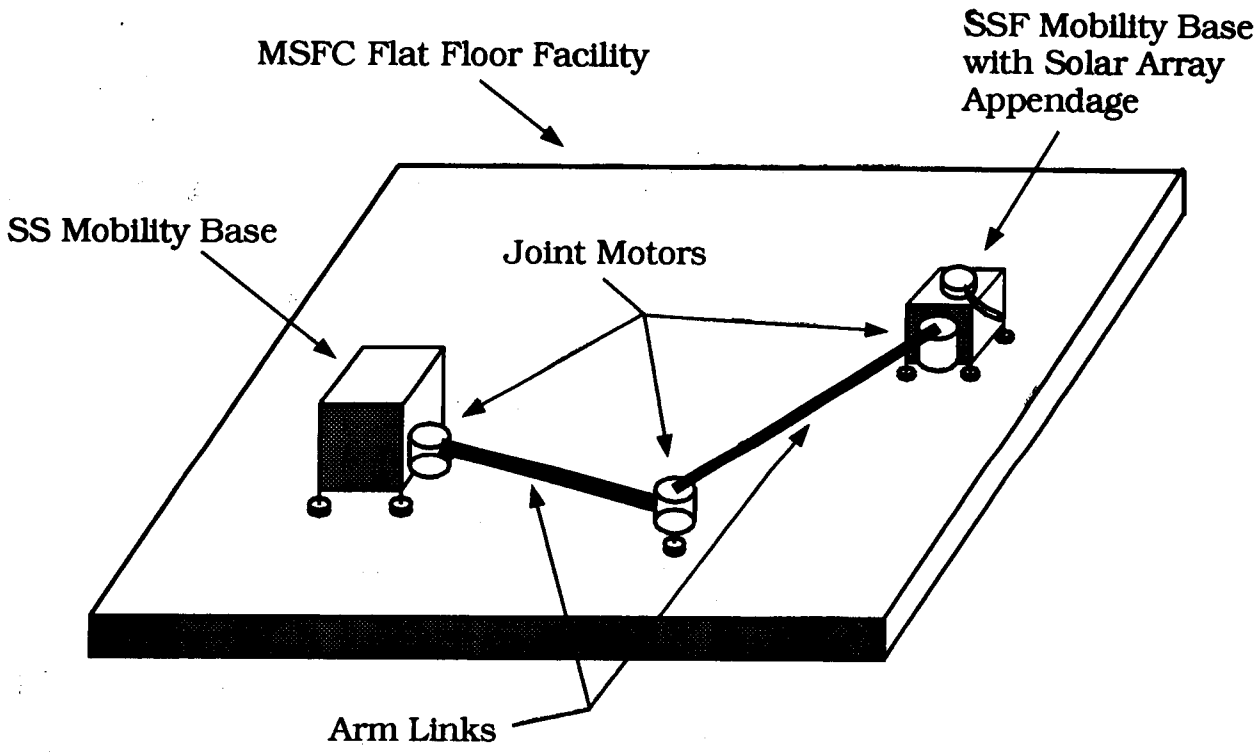
PRESENTATION OUTLINE

- Problem
- Testbed Overview
- Preliminary Design, Analytical Models, and Simulation Tools
- Initial Experimental Results and System Identification
- Concluding Remarks and Future Plans

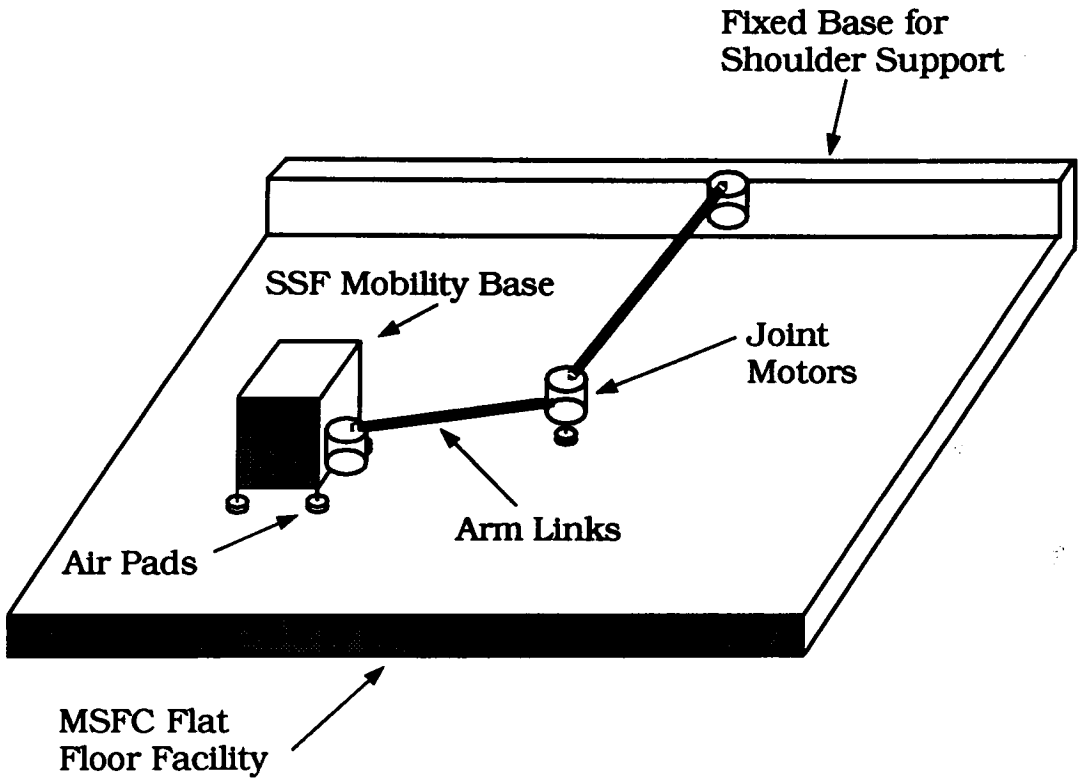
Space Station Berthing to the Space Shuttle



Multibody Controls Research Testbed
A Joint MSFC/LaRC Testbed



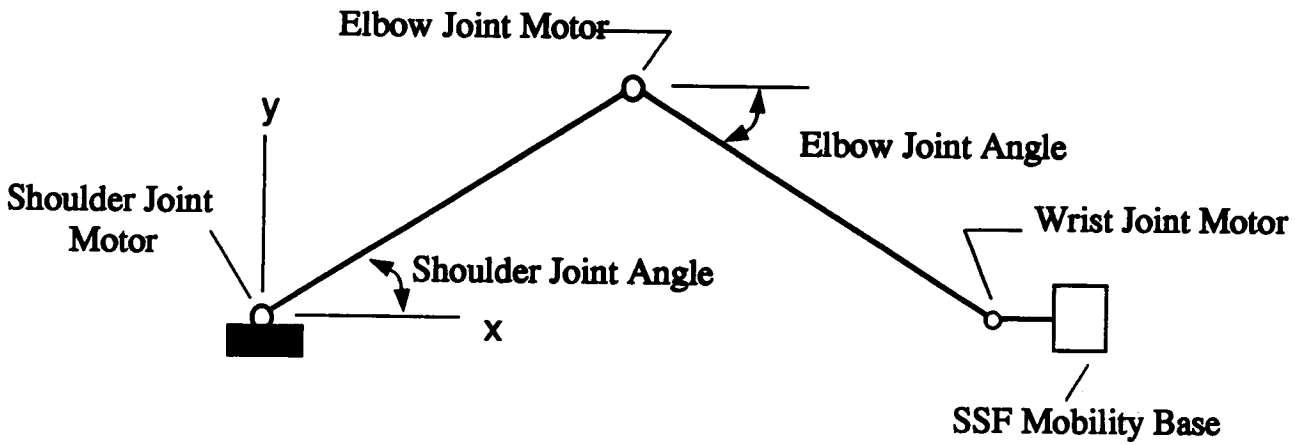
Interim Research Testbed



SSF Mobility Base Features

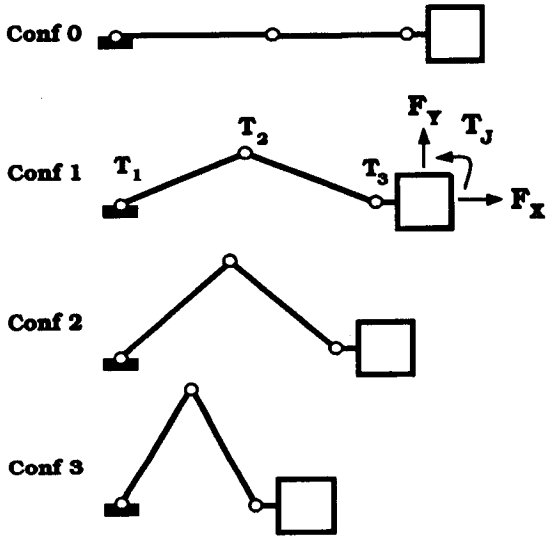
- Propulsion — 24 cold gas thrusters
- Sensors — 1 rate gyro, 2 accels, 2 laser scanners
- Attitude Control — 1 CMG
- Mechanical Support — 3 air bearings
- Power — 48 VDC from 10 NiCd Batteries
- Computer — 386 Based Microcomputer
- Communications — Radio

Nomenclature



Preliminary Design

Configurations Considered



Preliminary Design (Continued)

Structural Evaluation

- Design goal — .05 Hz for first bending mode
- NASTRAN analysis for structural evaluation
- Analysis for joints fixed and pinned

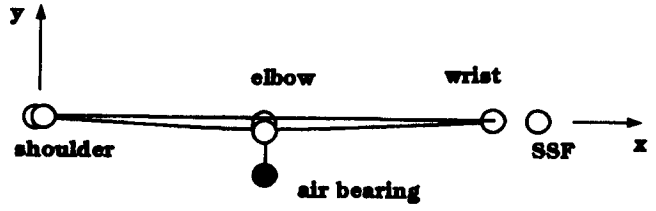
Controls Evaluation

- TREETOPS simulation for control effectivenesses — PD designs
- MatrixX simulation for control law evaluations — inverse kinematics designs

Results from Preliminary Design

- Arm links — 2.74 m, I beams
- Jets and CMG adequate for control research
- Joint motors adequate
- Problem uncovered in the elbow support design — fixed

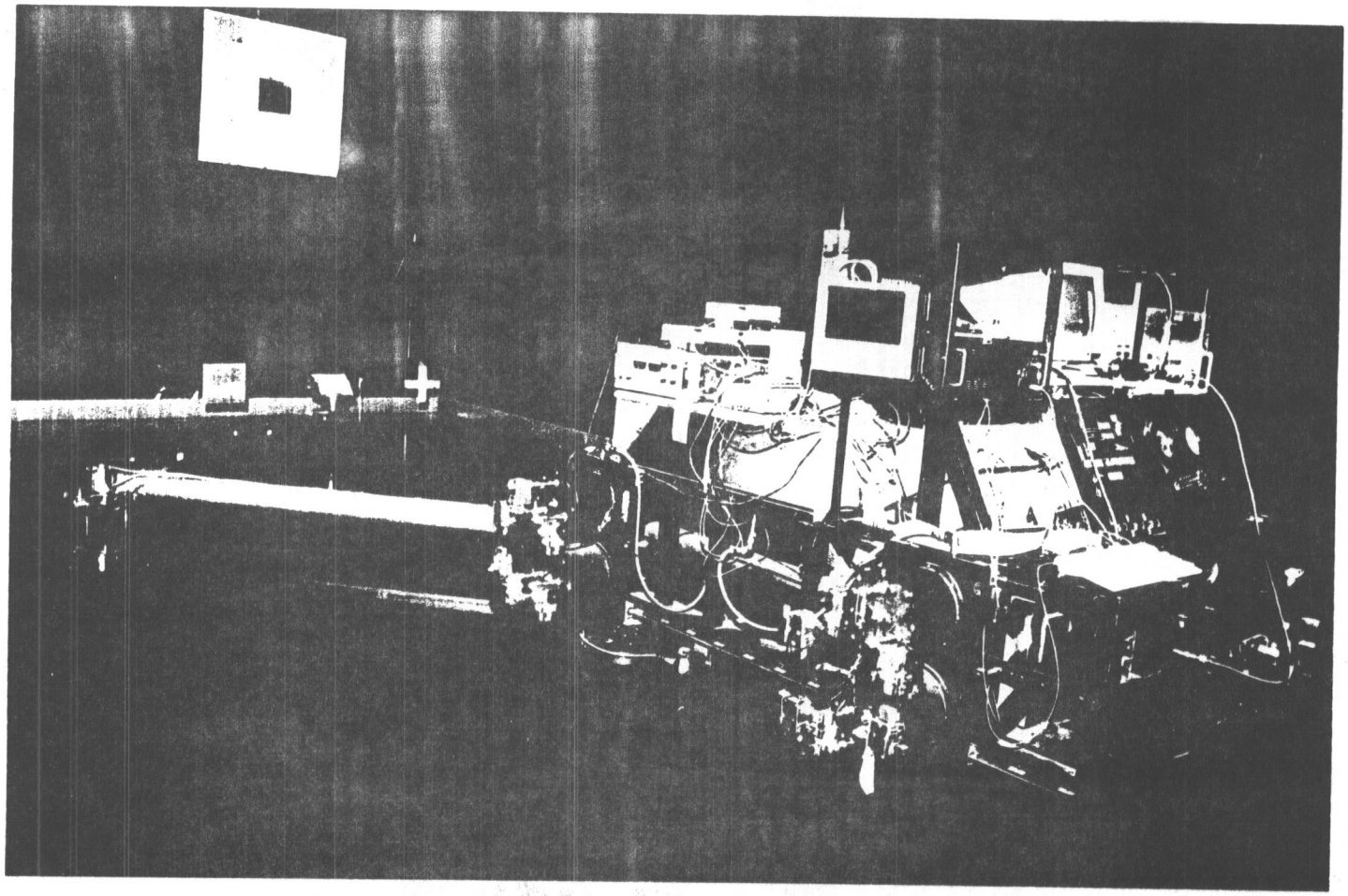
Configuration 0, 2.04 Hz. torsion mode



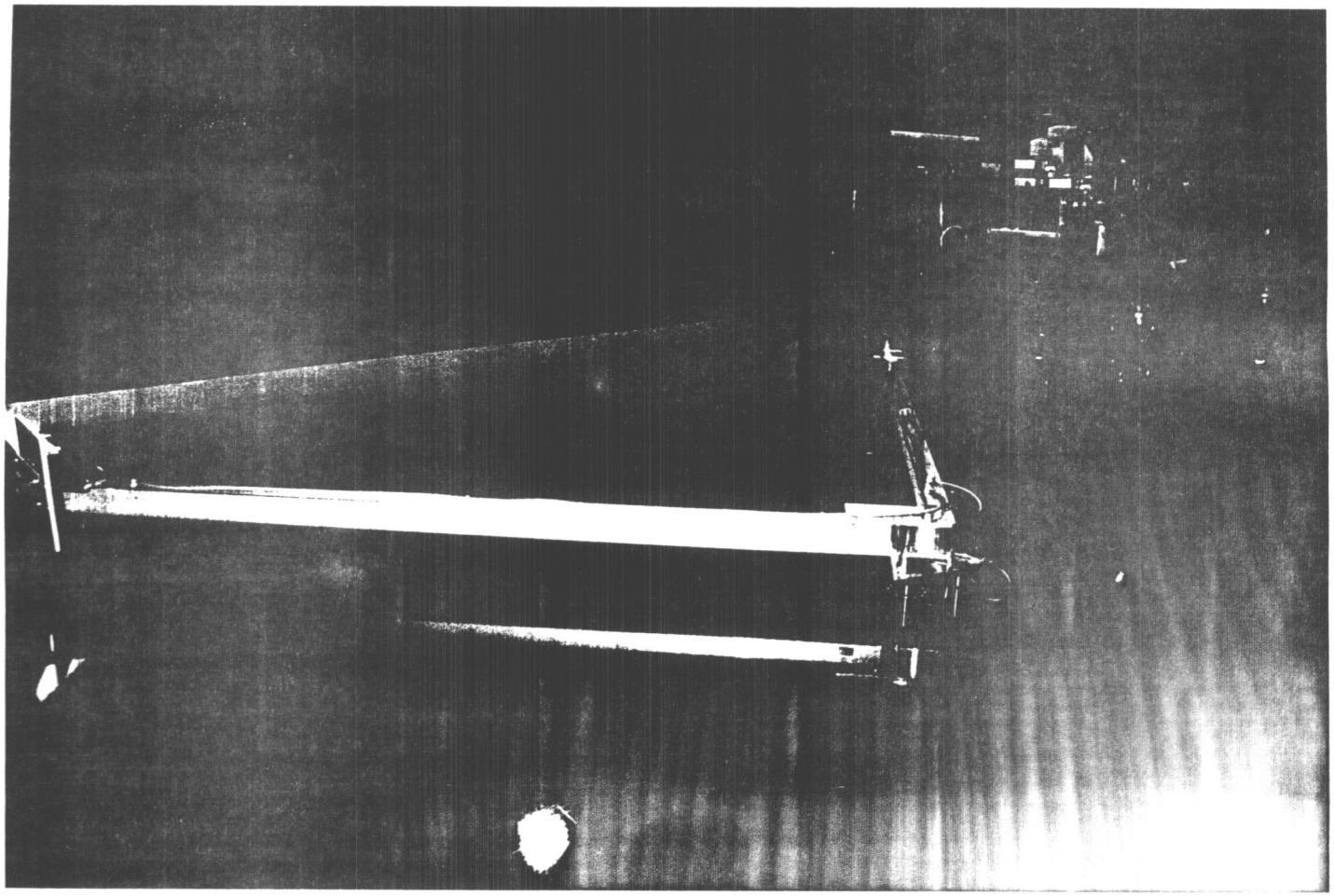
Cantilevered System Natural Frequencies

Model	Frequencies, Hz.						
Conf 0 fixed pinned	3 Rigid	.04 2.88	.048 4.29	2.25 5.99	12.03 12.03	18.38 18.73	24.1 24.43
Conf 1 fixed pinned	3 Rigid	.04 2.78	.24 4.12	.6 5.7	18.37 18.72	22.9 23.26	
Conf 2 fixed pinned	3 Rigid	.05 2.78	.18 4.11	.6 5.71	18.37 18.72	22.9 23.28	
Conf 3 fixed pinned	3 Rigid	.06 2.77	.17 4.11	.64 5.72	18.36 18.7	22.93 23.28	

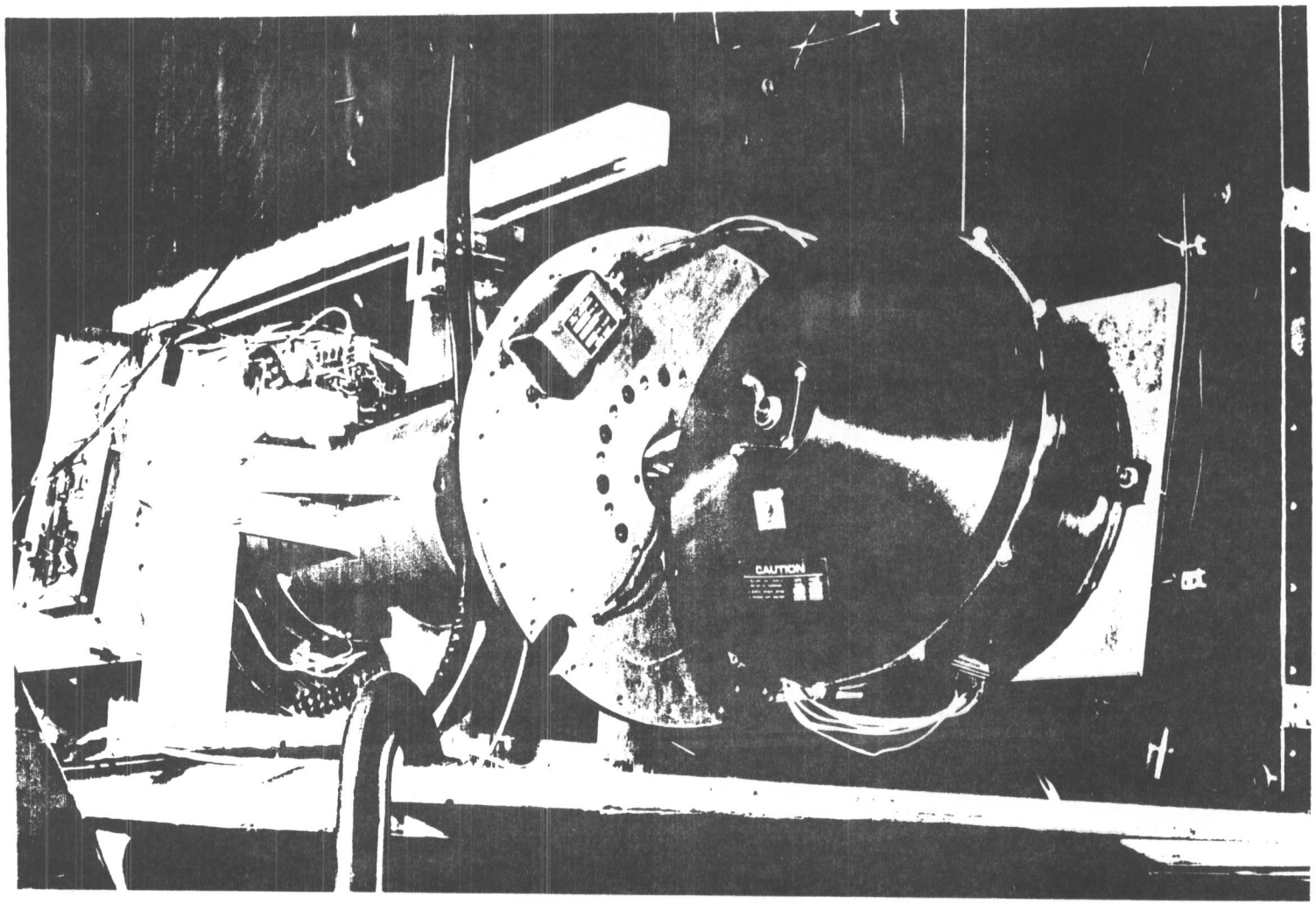
Photograph of the Testbed
View from Mobility Base



Photograph of the Testbed
View from Wall

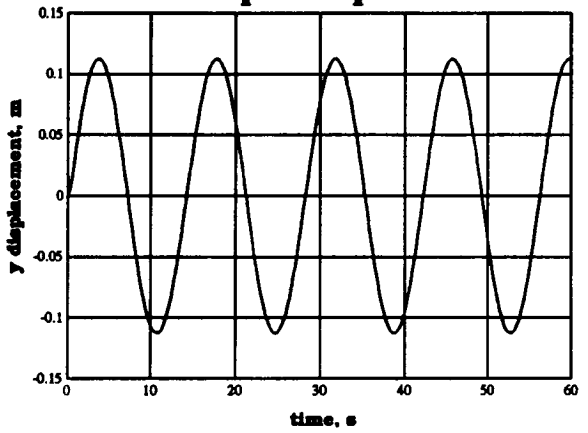


Photograph Showing the CMG

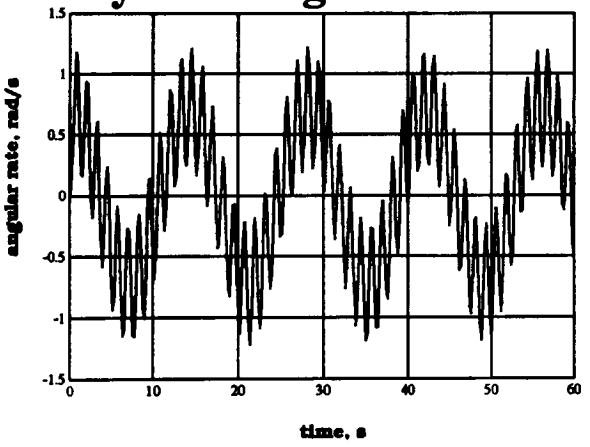


Open-Loop Response

Lateral Tip Displacement

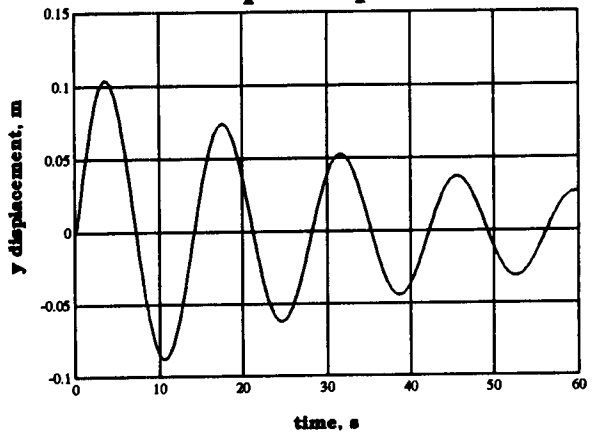


Payload Angular Rate

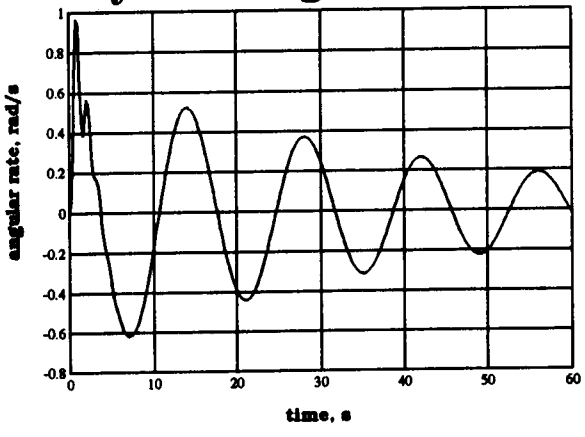


Closed-Loop Response

Lateral Tip Displacement

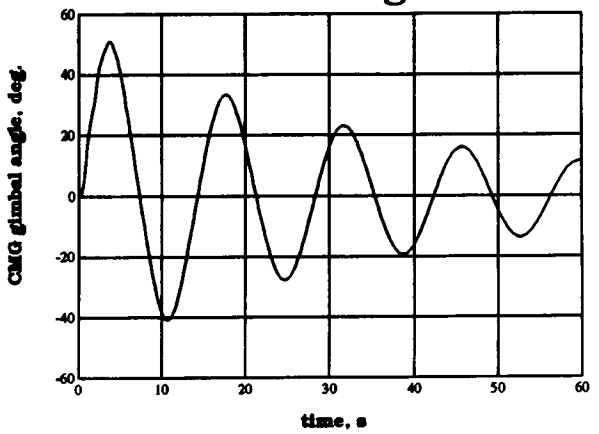


Payload Angular Rate

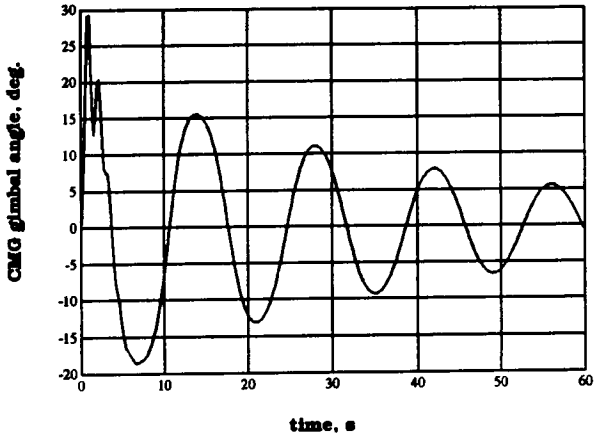


Closed-Loop CMG Requirement

Gimbal Angle



Gimbal Rate



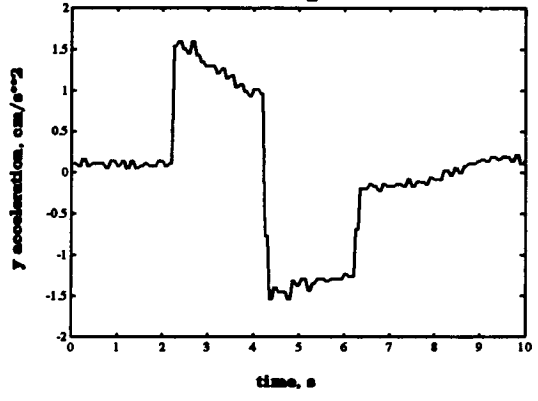
Initial Testing

- Configurations 0 — 4 tested in Dec.'92
- Pinned and emulated fixed joint conditions
- Excessive noise in sensor data when feedback used
- System ID conducted on useful data
- Retesting scheduled Jun 24,'93

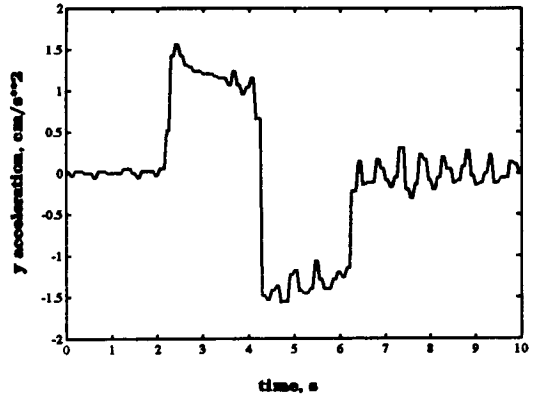
Y-Jet Firing Results for Conf. 1

Y-Accelerometer

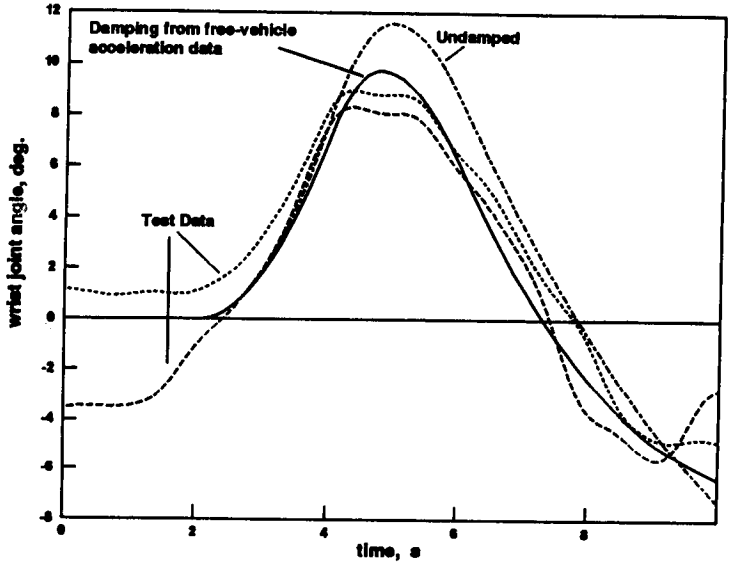
Free Response



Joint Motor Position Feedback



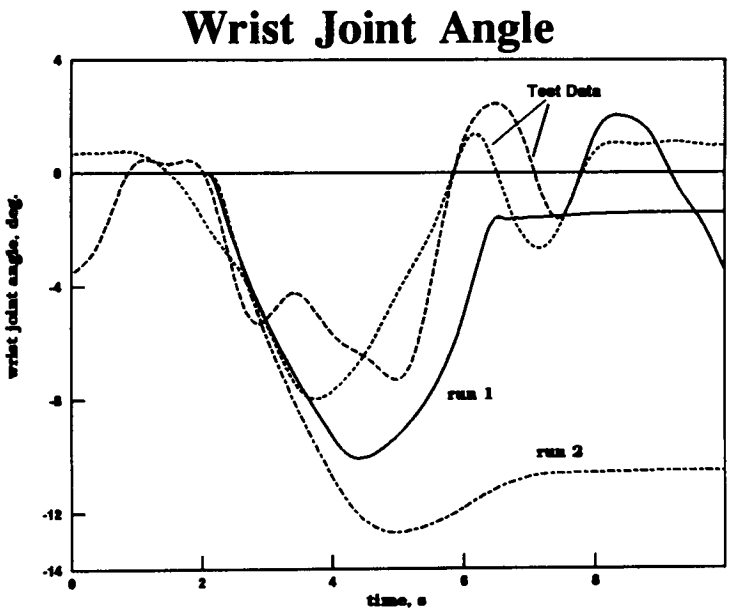
Wrist Joint Torque Input
Gear Efficiency = 50%



Wrist Joint Angle

Elbow Joint Torque Input

Effect of Gear Efficiency:
50% for run 1
75% for run 2



Concluding Remarks and Future Plans

- Design and Construction of the Testbed has been completed
- Joint Effort between MSFC and LaRC
- Characterization Testing scheduled for June 24
- Initial Experimental Results have been obtained
- Modal Testing has just been completed by MSFC
- Testbed planned to be operating in Aug.'93

Status Report of RMS Active Damping Augmentation

Mike Gilbert
NASA Langley Research Center
Hampton, Virginia

and

Martha E. Demeo
VIGYAN
Hampton, Virginia

STATUS REPORT OF RMS ACTIVE DAMPING AUGMENTATION

Michael G. Gilbert
Assistant Head
Spacecraft Dynamics Branch
NASA Langley Research Center
Hampton, VA.

Martha E. Demeo
Vigyan Inc.
Hampton, VA.

Workshop on Selected Topics in Robotics for Space Exploration
March 17-18, 1993

ACKNOWLEDGEMENTS

Charles Stark Draper Laboratory / Cambridge, MA
Janet Lepanto
Karl W. Flueckiger
Kevin M. Gift
Richard W. Metzinger
Lockheed Engineering & Sciences Company / Houston, TX
Mary C. Jensen
Jesse M. Moon
K. K. Rob
Susan B. Rogers
John M. Teel
Sam V. Veerasamy
NASA Johnson Space Center / Houston, TX
Elizabeth Bains
Ying M. Kuo
Ann E. Madison
NASA Langley Research Center / Hampton, VA
Michael A. Scott
Lockheed Engineering & Sciences Company / Hampton, VA
Georgette W. Watts

BRIEFING OUTLINE

Introduction

- Objectives
- Benefits
- Schedule

Summary of technical accomplishments

- Study parameters
- Astronaut evaluations

Fiscal 93 plans

- Heavy payload (SSF) studies
- Needed feasibility/cost data

ACTIVE DAMPING AUGMENTATION

Objective :

Improve RMS operational capability by reducing vibration decay time

Approach :

Apply Controls-Structures Integration (CSI) technology

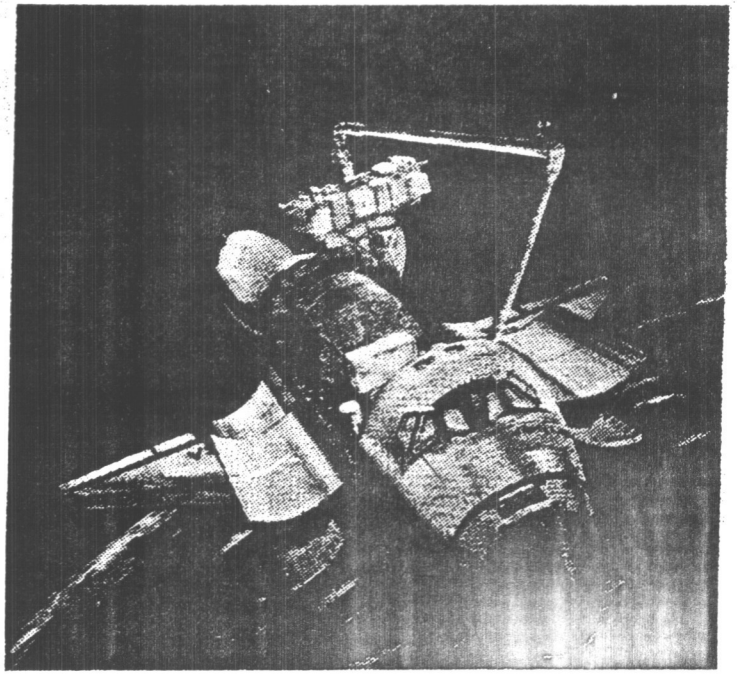
- sense the vibratory motion

- implement a feedback of the sensed motion to drive the joint servos and increase damping

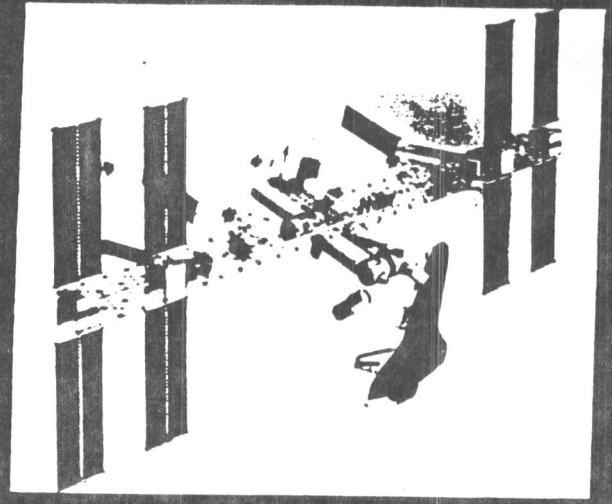
Joint LaRC / JSC program

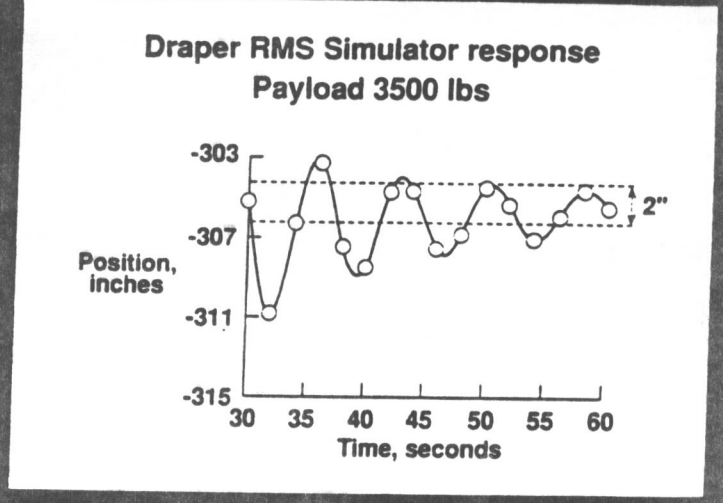
BENEFITS OF RMS ADA

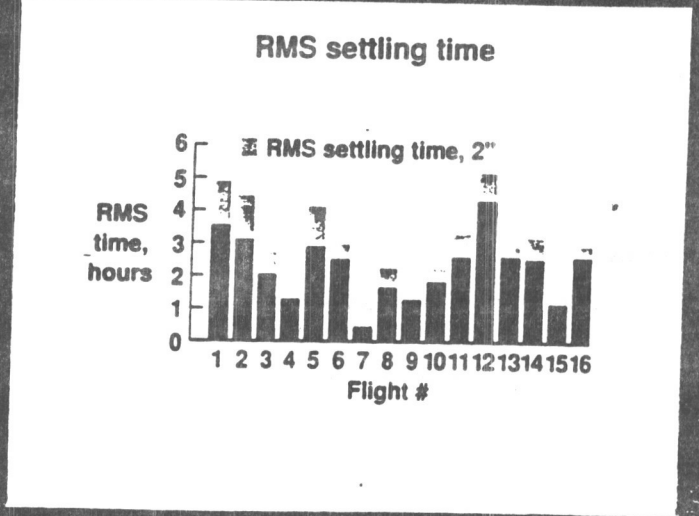
- **Reduced vibration decay times (improved operational timelines) following**
  - Commanded RMS maneuvers
  - Shuttle thruster firings
- **Reduced RMS loads**
- **Improved precise payload positioning capability**
- **Reduced RMS / Shuttle FCS interaction potential**
- **Reduced crew training time**

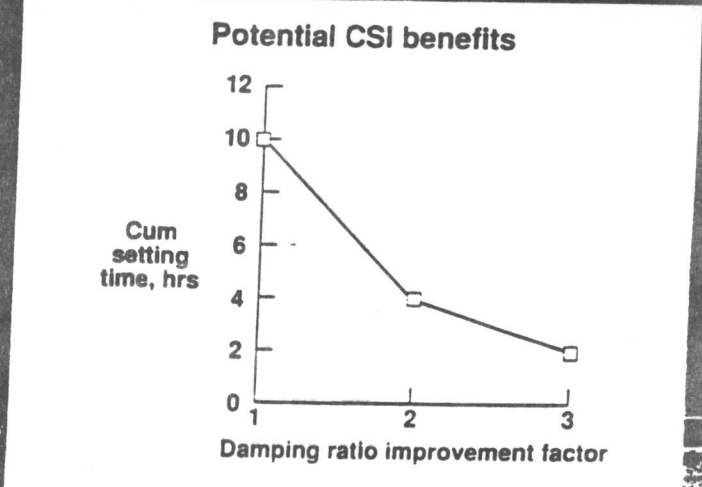


POTENTIAL SPACE STATION ASSEMBLY BENEFITS DUE TO CSI (Timeline)

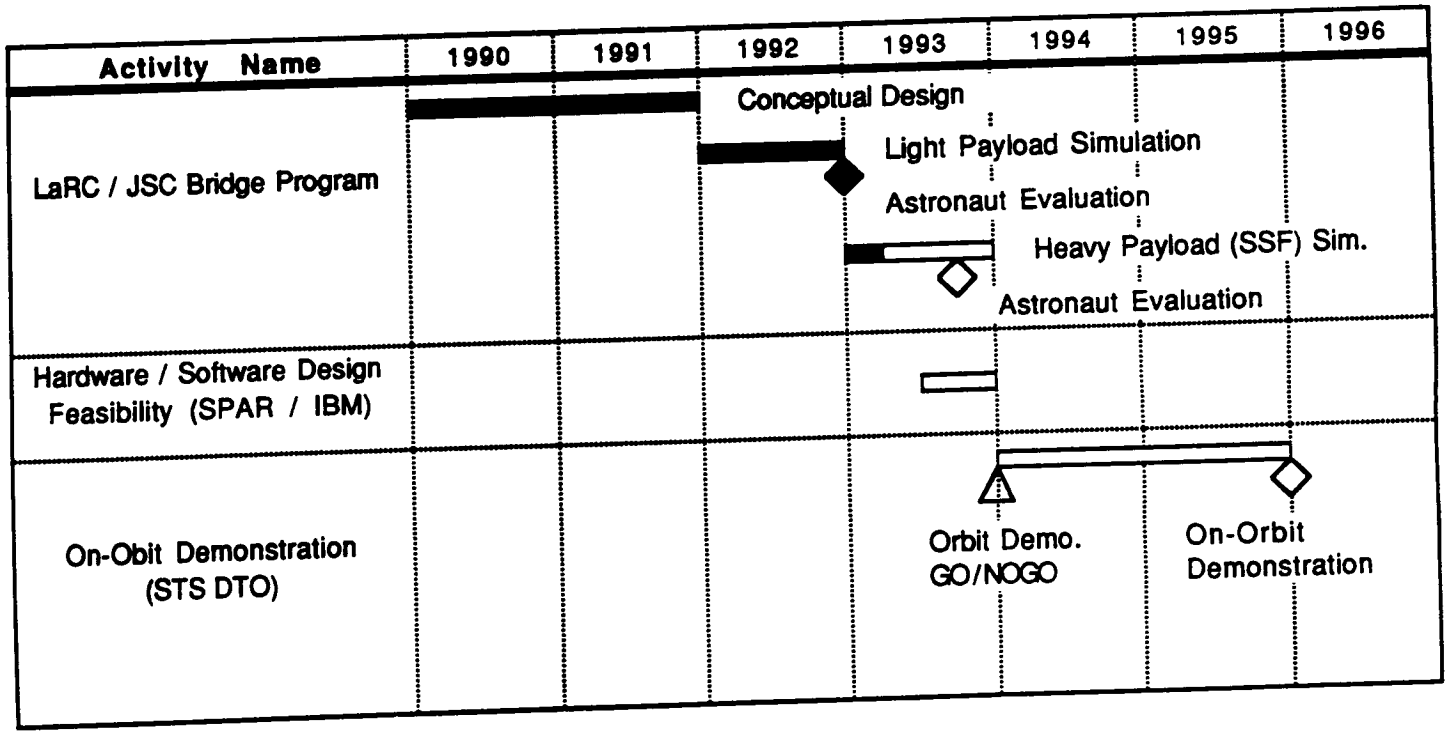






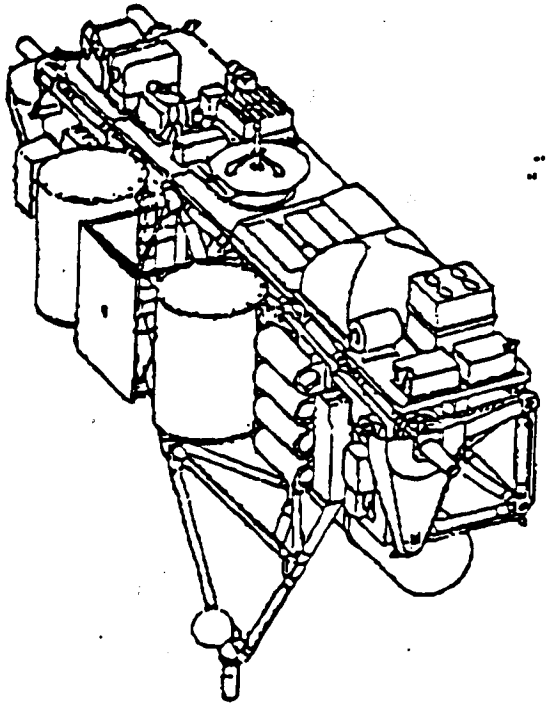


RMS ACTIVE DAMPING AUGMENTATION SCHEDULE

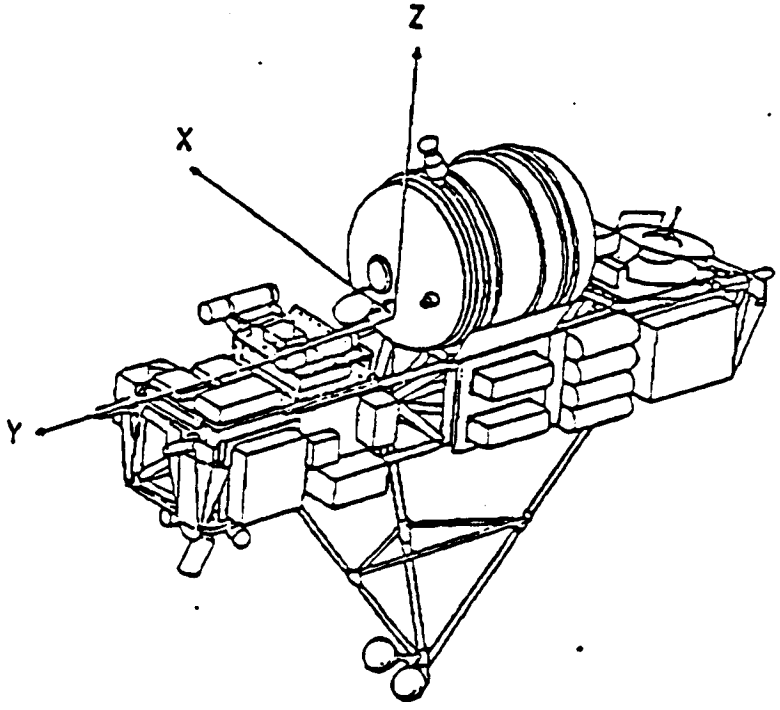


SIMULATED PAYLOAD REDEFINITION

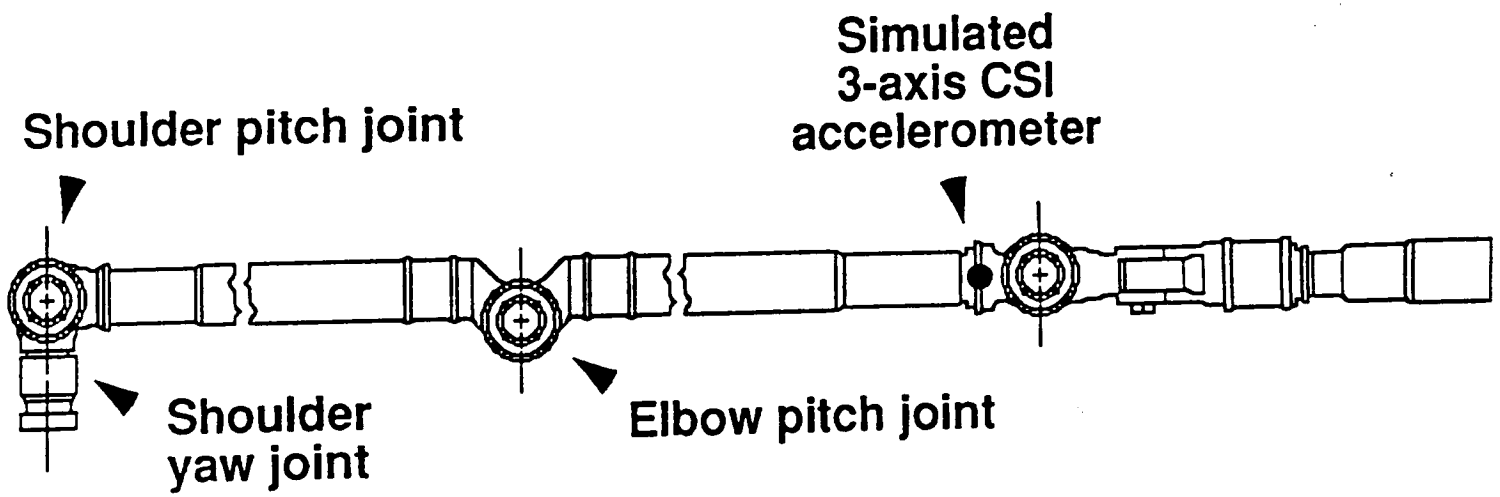
SPAS - 01 (STS - 7)



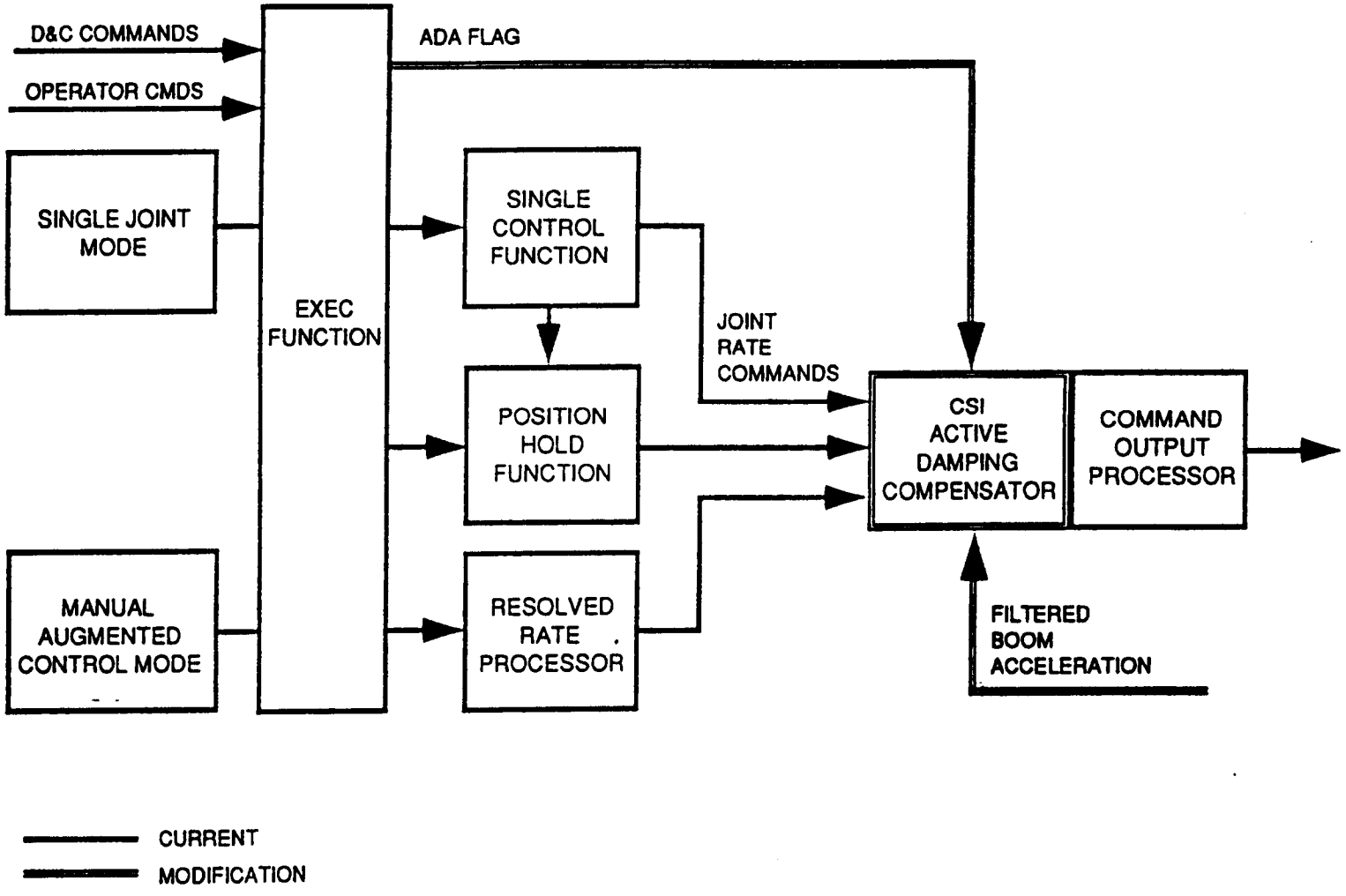
SPAS - 02 (IBSS, STS - 39)



SENSOR AND ACTUATOR DEFINITION



CSI CONTROLLER IMPLEMENTATION



ADA CONTROL LAW DESIGN

- **Exercised several control law design methodologies**
- **Used linear design models identified from simulated response data**
  - Consistent with on-orbit flight demonstration approach
  - Control laws checked on flight verified on nonlinear, non-real-time simulation
- **Control design objectives for astronaut evaluation**
  - 1st - Robustness to RMS configuration changes
  - 2nd - Damping performance

SES REAL-TIME SIMULATION

Plan:

Conduct man-in-the-loop simulation studies of active damping augmentation in the Shuttle Engineering Simulator (SES) at JSC

Goals:

- Obtain qualitative definition of performance improvement by astronaut operators
- Obtain supporting quantitative performance data

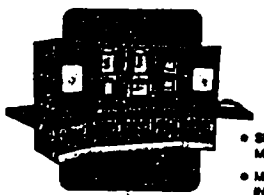
Objectives:

- Measure reductions in RMS vibration decay time following RMS maneuvers and Shuttle thruster firings
- Identify reductions in predicted RMS loads
- Insure no adverse interaction with Shuttle FCS

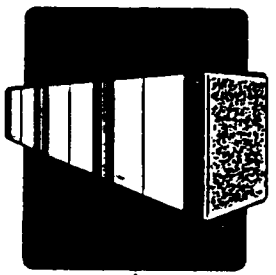
SHUTTLE ENGINEERING SIMULATOR - ON-ORBIT

SES ON-ORBIT FEATURES

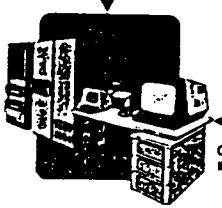
- THREE-BODY SIMULATION OF:
  - ORBITER
  - FREE-FLYING PAYLOAD
  - MANNED MANEUVERING UNIT
- ON-ORBIT OPERATIONS CAN INCLUDE ANY OR ALL OF THE FOLLOWING:
  - RENDEZVOUS
  - PROXIMITY OPERATIONS
  - PAYLOAD DEPLOYMENT AND RETRIEVAL WITH RMS
  - MMU OPERATIONS
- PAYLOAD MODELS DEVELOPED AS REQUIRED:
  - VISUALS
  - MASS PROPERTIES
  - PLUME IMPINGEMENT
  - AERO DRAG
  - RADAR SCATTER CHARACTERISTICS
  - CONTROL SYSTEM



- SIMULATION DATA MONITORING
- MALFUNCTION INSERTION
- SCENE MONITORS
- COCKPIT CRT MONITOR

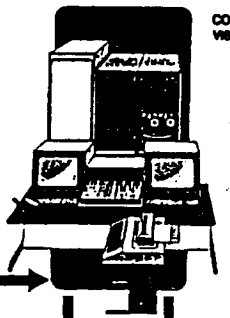


- EQUATIONS OF MOTION:
  - ORBITER
  - PAYLOAD
  - MMU
- ORBITER SUBSYSTEMS:
  - RADAR
  - IMU
  - ST
  - COAS
  - RCS
  - OMS
- GN&C FLIGHT SOFTWARE
- FLEX RMS
- ENVIRONMENT
- PLUME IMPINGEMENT
- MMU SUBSYSTEMS
- PAYLOAD CONTROL SYSTEM



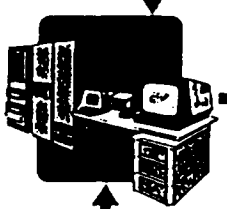
CONTROL CONSOLE INTERFACE

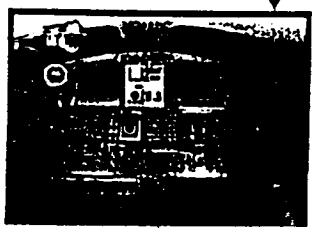
SHARED MEMORY



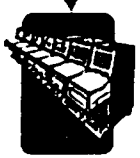
COMPUTER GENERATED VISUALS:
- ORBITER OVERHEAD VIEWS
- RMS AFT VIEW
- ORBITER CCTV'S
- MMU VIEW

- SIMULATION EXECUTIVE
- MAG TAPE DATA RECORDING
- STRIP-CHART RECORDING
- REAL-TIME PRINT
- SNAPSHOT PRINT





- CREW STATION INTERFACE
- COCKPIT CRT DISPLAYS
- SCENE INTERFACE



64-CHANNEL STRIP-CHART RECORDERS

CREW STATION INTERFACE

ADA SWITCH IN SES COCKPIT

STUDY STATUS

Phase 1 (completed April 3, 1992)

- SES familiarization
- Acceleration validation
- Collected data for system identification

Phase 2a (completed July 7-9, 1992)

- Compensator implementation logic validated
- Acceleration filter validated
- 5 baseline control sets evaluated
- Astronaut trainer evaluation (Ann E. Madison)

Phase 2b (completed September 22-24, 1992)

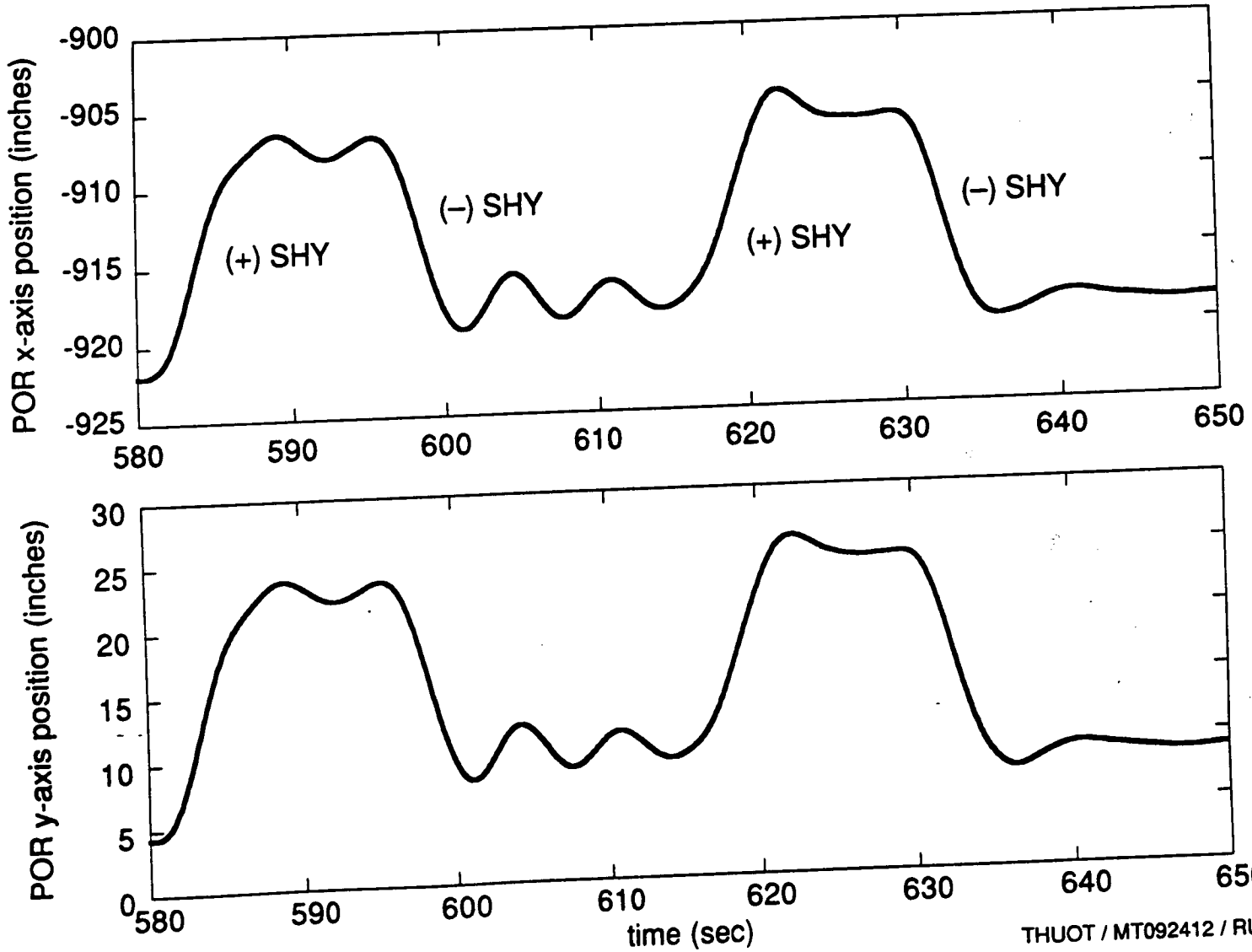
- Multi-point control law testing
- Astronaut evaluation of RMS ADA

ASTRONAUT EVALUATION

- Six astronauts briefed on ADA and participated in qualitative evaluation of ADA performance. (Thagard, Brown, Sherlock, Godwin, Reightler, and Thuot)
- Evaluated performance of a single ADA control law designed to work over a range of configurations.
- Examined ADA following Single and Manual RMS maneuvers and ADA disturbance rejection of Shuttle PRCS firings.
- Some astronauts were reluctant to deviate from conservative operating practices while others were more aggressive and willing to "push the envelope".
- Comments on ADA's potential operational benefit ranged from *"limited"* to *"a big improvement"*.
- Unanimous recommendation to **evaluate ADA with heavy payloads.**

Astronaut Evaluation:

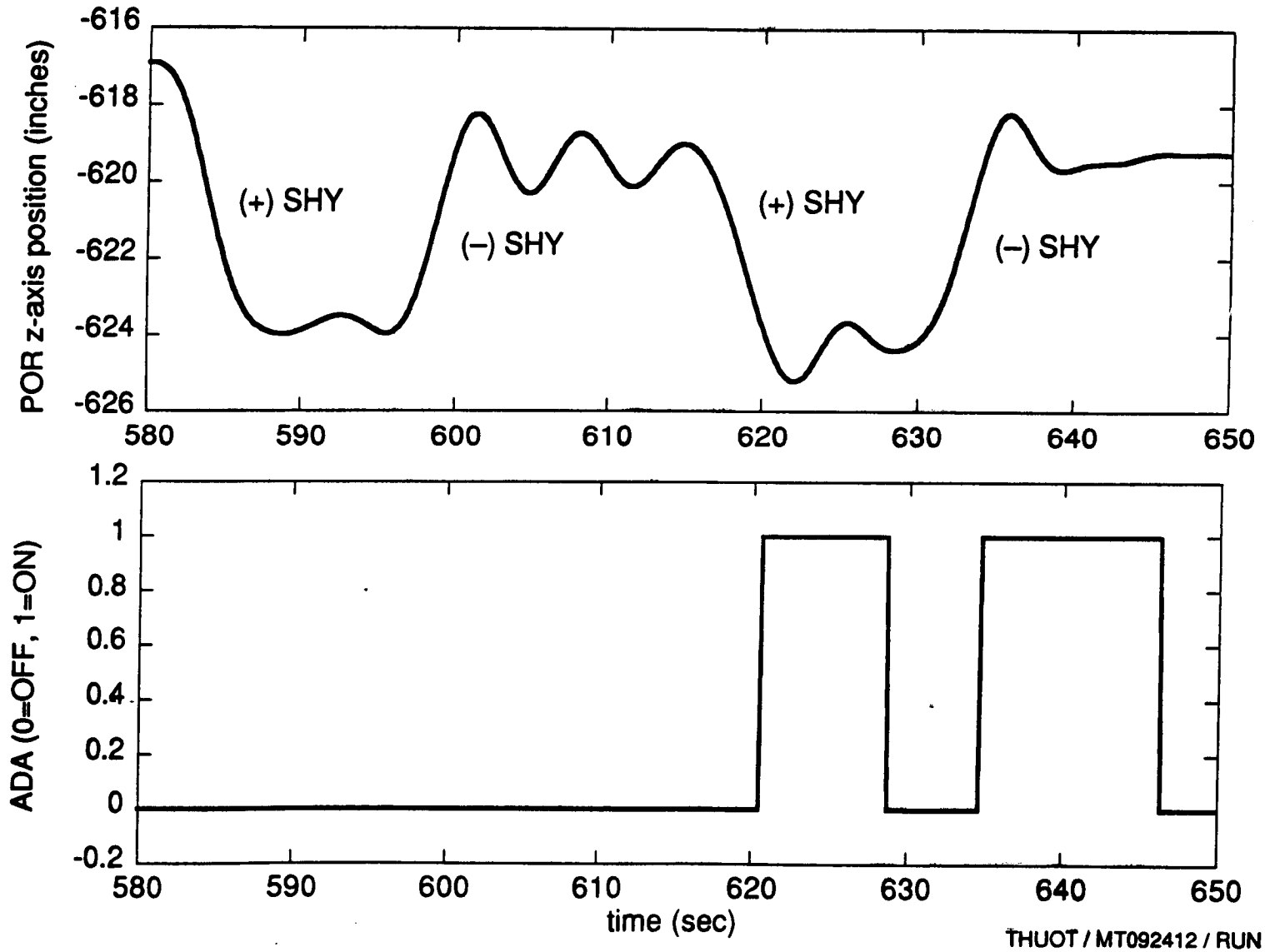
+/– Shoulder Yaw (SHY) Single Joint Maneuvers



THUOT / MT092412 / RUN 1 / MASC2

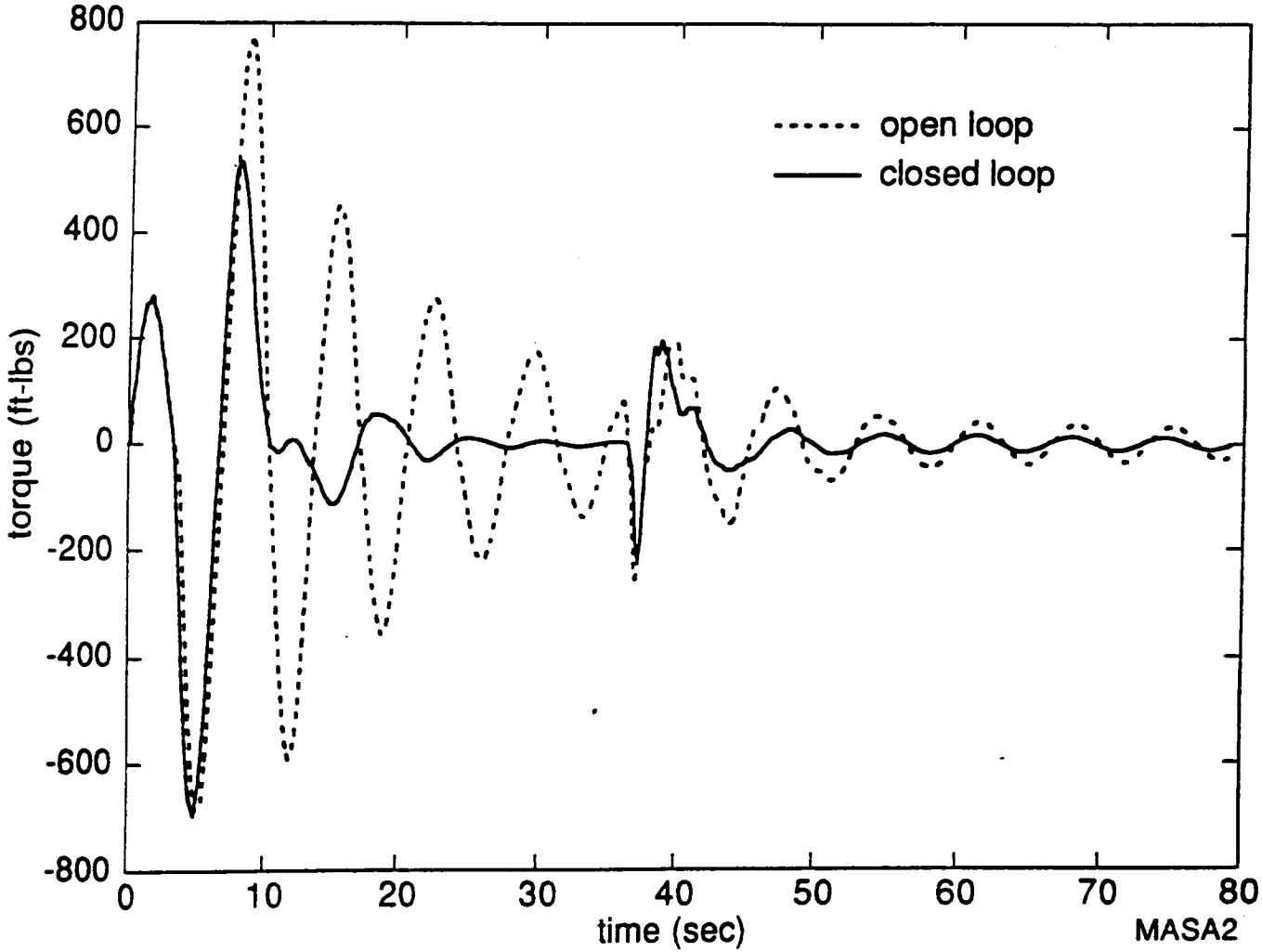
Astronaut Evalution: (cont'd)

+/– Shoulder Yaw (SHY) Single Joint Maneuvers



THUOT / MT092412 / RUN 1 / MASC2

PHASE 2a SES RESULTS: Joint Torque

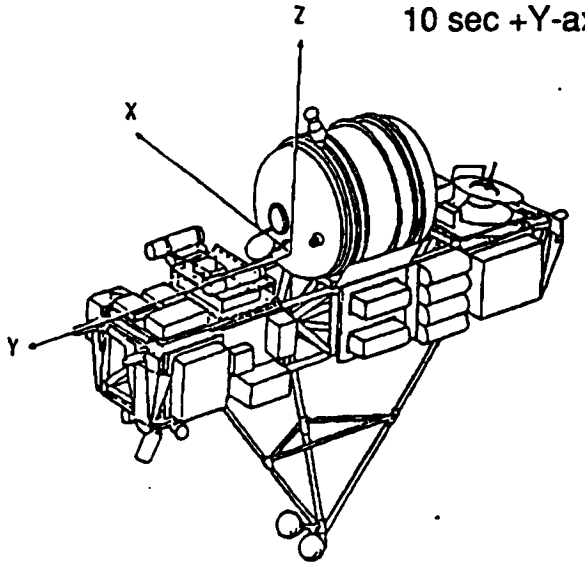


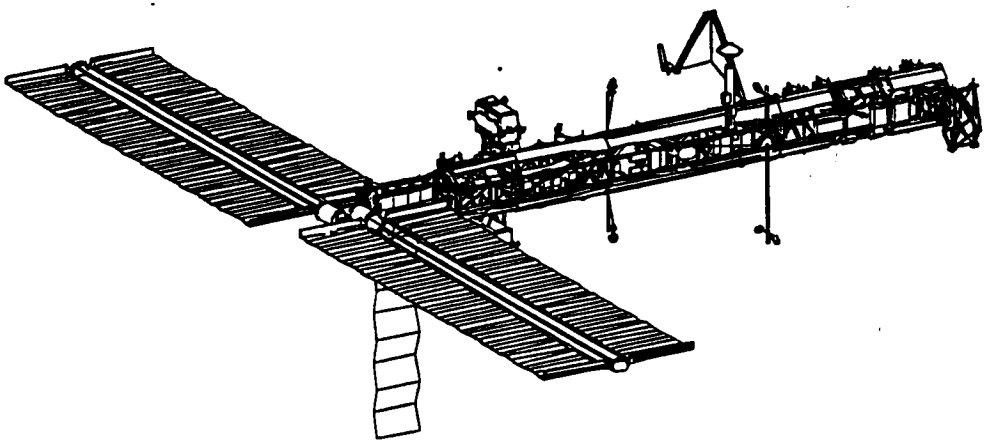
ASTRONAUT RECOMMENDED SSF ASSEMBLY STUDY

- **Unanimous concern**
  - may enable single-joint mode backup assembly operations
- **Study underway**
- **Includes RMS upgrades**
  - Position hold submode improvements
  - Wrist joint servo upgrades
  - POSH mode (no direct influence)
- **Continued CSDL support**
- **Uses SC-4 SSF payload**
- **Astronaut evaluation scheduled for June 1993**

Typical Response: SPAS-02 vs. SC-4

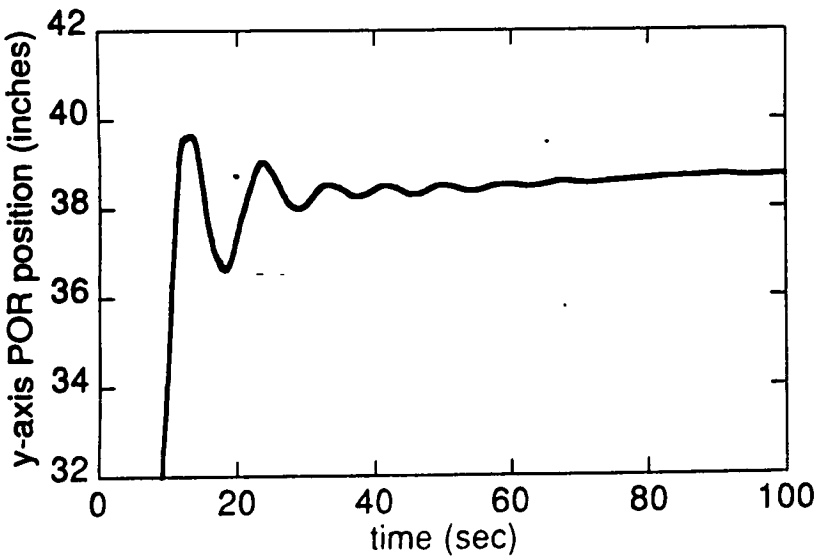
10 sec +Y-axis THC Manual Mode Maneuver

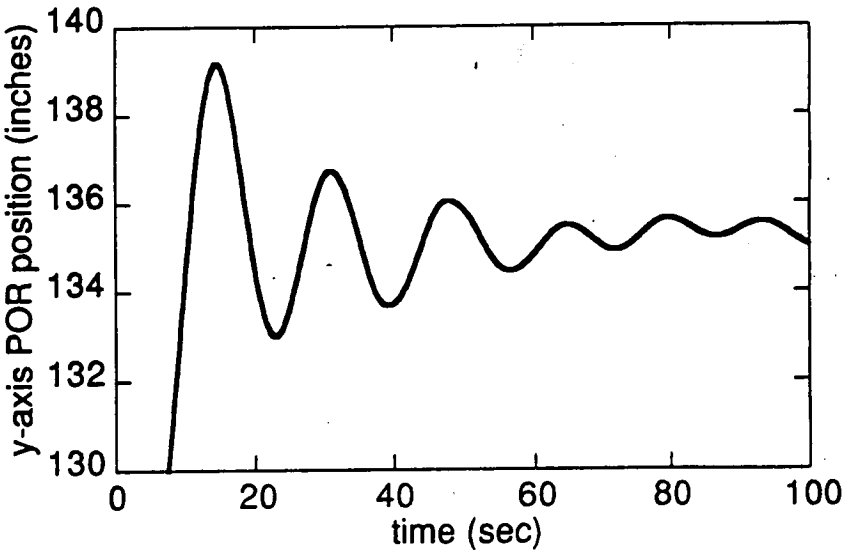




SPAS - 02 (IBSS, STS - 39)
3,990 lbs.

SC-4 Space Station
131,000 lbs.





INFORMATION NEEDED FOR FLIGHT DEMO DECISION

→ User interest / benefit

- Independent evaluation
- Accelerometer installation feasibility
- Flight software accommodation
- Cost estimates (flight demonstration)

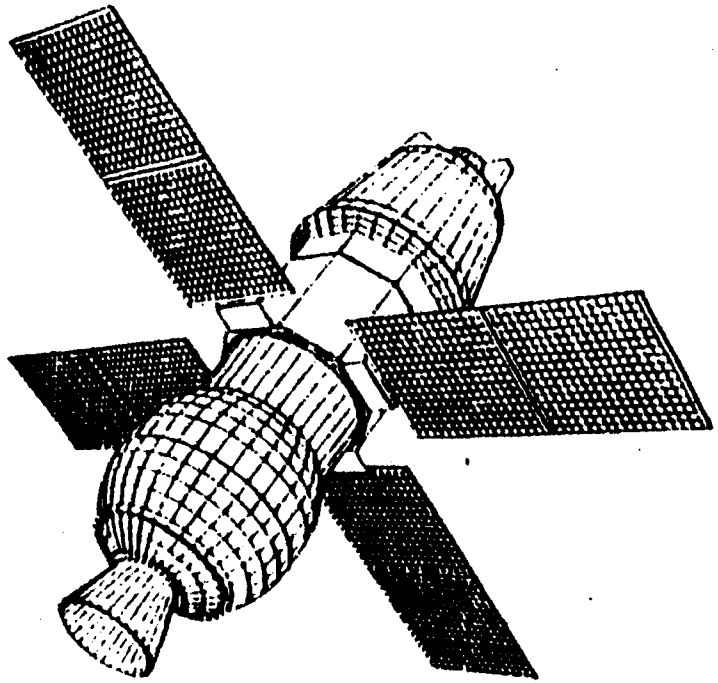
CONCLUDING REMARKS

- **Heavy payload system identification and control design is ongoing**
  - Expect linear models for control design by mid-April
  - Expect candidate control laws mid-May
- **Heavy payload SES testing is scheduled**
  - Late May: Control law validation testing
  - Late June: Astronaut performance evaluations
- **Briefed Director of SSF (R. Kohrs) on status 2/3/93**
  - Value for SSF assembly
- **Looking for flight demonstration feasibility/cost information**
  - IBM for GPC software implementation of ADA
  - SPAR for accelerometer installation and overall approach

Autonomous Rendezvous and Docking for Space Station Freedom

James L. Garrison, Jr. and Stephen J. Katzberg
NASA Langley Research Center
Hampton, Virginia

LANGLEY RESEARCH CENTER



PROPOSED AR&D INITIATIVE

AUTONOMOUS RENDEZVOUS AND DOCKING STUDIES FOR SPACE STATION *FREEDOM*

OVERVIEW OF WORK IN PROGRESS AND PLANS FOR 1993

March 17, 1993

James L. Garrison
Stephen J. Katzberg

NASA Langley Research Center

REQUIREMENTS FOR AR&D EXPERIENCE

Space Station:

"... reinitiate and complete efforts to develop operational scenarios and requirements for rendezvous, proximity operations, capture, and attached operations of transfer vehicles ***other than the Space Shuttle.***"

(R. Kohrs memo - Nov. 30, 1992)

Earth Sciences:

"Communications continue in good working order...GOES-5 is ***no longer controllable because of insufficient fuel.*** On January 28, 1989 a revised transmission schedule was implemented."

(APT Information Note 89-1 - March 13, 1989)

Technology Validation:

"... the primary recommendation of the committe [is] that NASA establish a national space facility for the development of space automation and robotics, one element of which is a ***telerobotic research platform in space.***"

(Annual Report of the Langley GN&C Technical Committe - September, 1992)

OPPORTUNITY FOR LOW COST AR&D FLIGHT EXPERIENCE

Langley researchers have proposed cooperation with the Space Automation and Robotics Center (SpARC) of the Environmental Research Institute of Michigan (ERIM) on their AR&D experiment planned as part of the COMET Program.

SpARC is a NASA Center for the Commercial Development of Space (CCDS).

COMET, the COMmercial Experiment Transporter is an orbital system operated by the University of Tennessee Space Institute (UTSI) exclusively for use of the CCDS's.

SpARC will make use of the first two COMET Service Modules, following completion of on-orbit experiments, to attempt AR&D.

This series of two missions (4/93 and 11/94) are already funded Conestoga launches out of Wallops Is.

Langley was asked to provide simulations and analysis of the mission, recommend algorithm enhacements, and perform ground-based hardware testing and simulation, if appropriate.

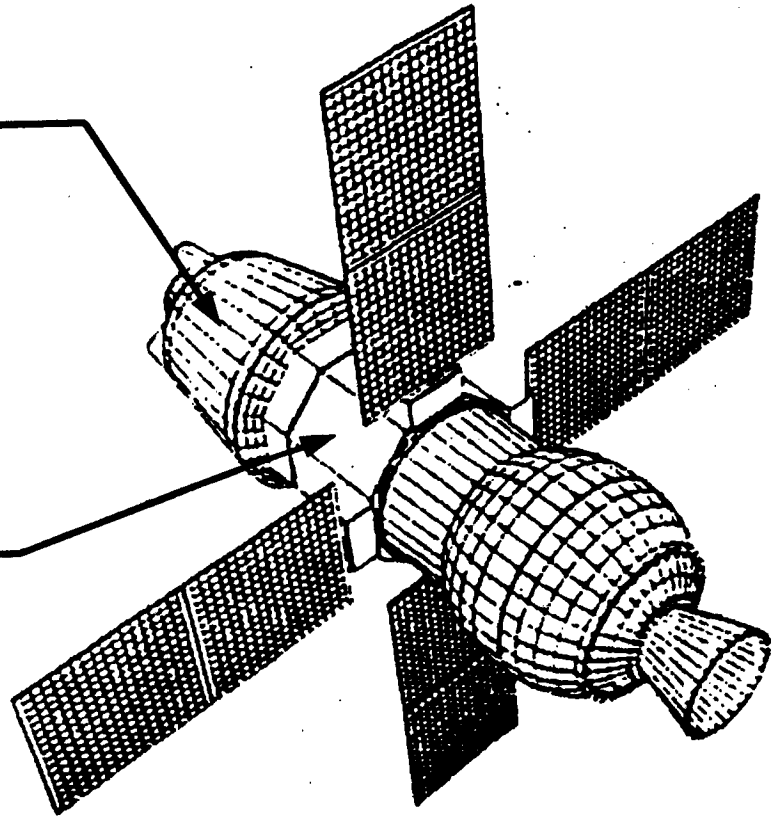
COMET SPACECRAFT PERFORMANCE

Recovery Module

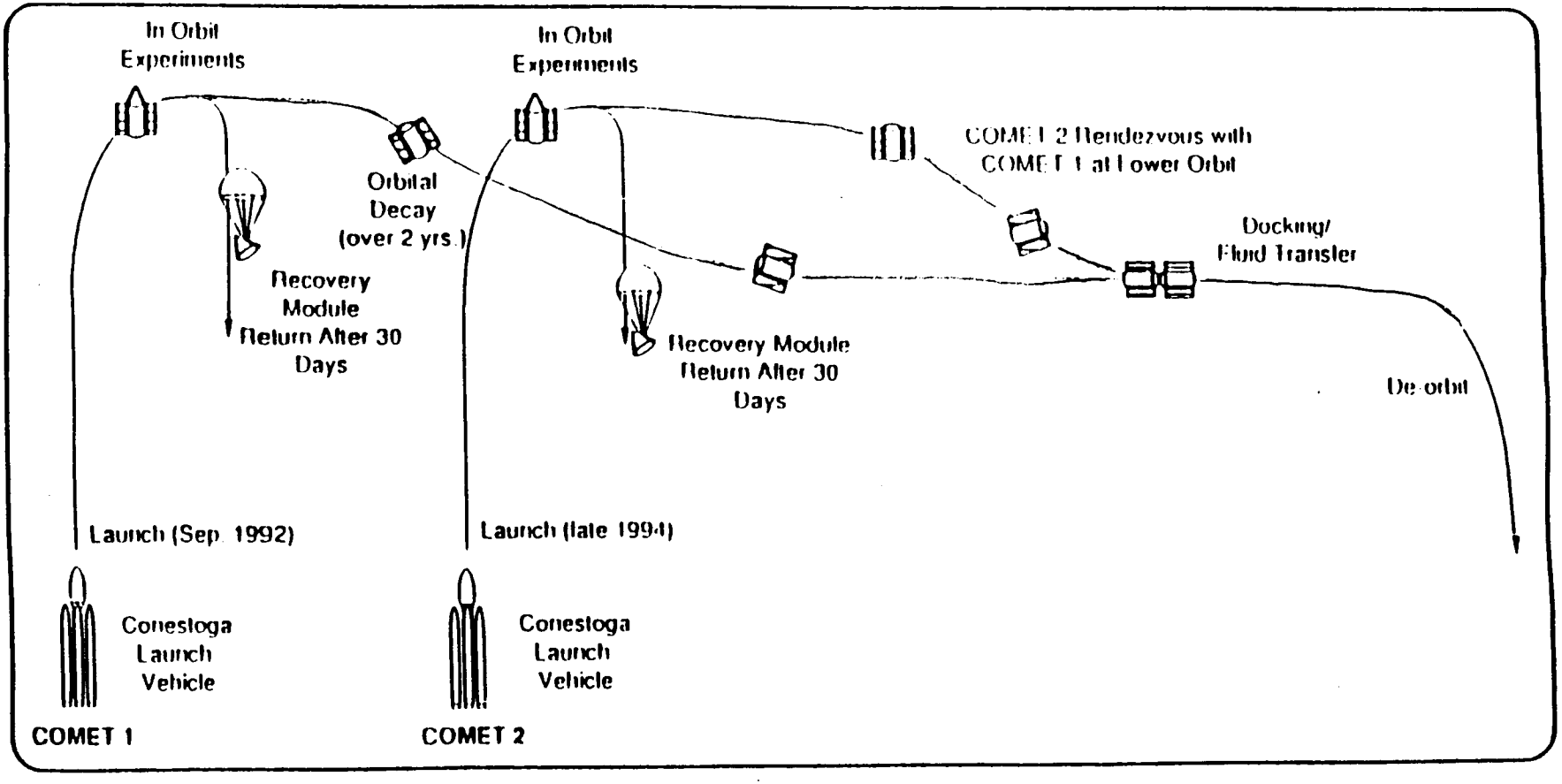
Mission Duration: 30 days
Payload Mass: 300 lbm
Payload Volume: 10 cu. ft.

Service Module

Mission Duration: 130+ days
Payload Mass:150 lbm
Payload Volume: 15 cu. ft.



AR&D MISSION PROFILE



ANALYSIS IN SUPPORT OF THE COMET AR&D MISSION

Independent Assesment of R-bar VS V-bar Approach

Controllability of Loosely Coupled Configuration

Approach Trajectory Optimization and Simulation

Contact Dynamics - Impact on Attitude Control of Target

Definition of AR&D Standards

ANALYTICAL MODELS IDENTIFIED FOR THREE REGIMES OF MISSION

Approach Trajectory

Loosely Coupled Configuration

Contact Dynamics

MODEL FOR APPROACH/PROXIMITY OPS.

RCS Control of Pursuit Vehicle

Translational Control Only

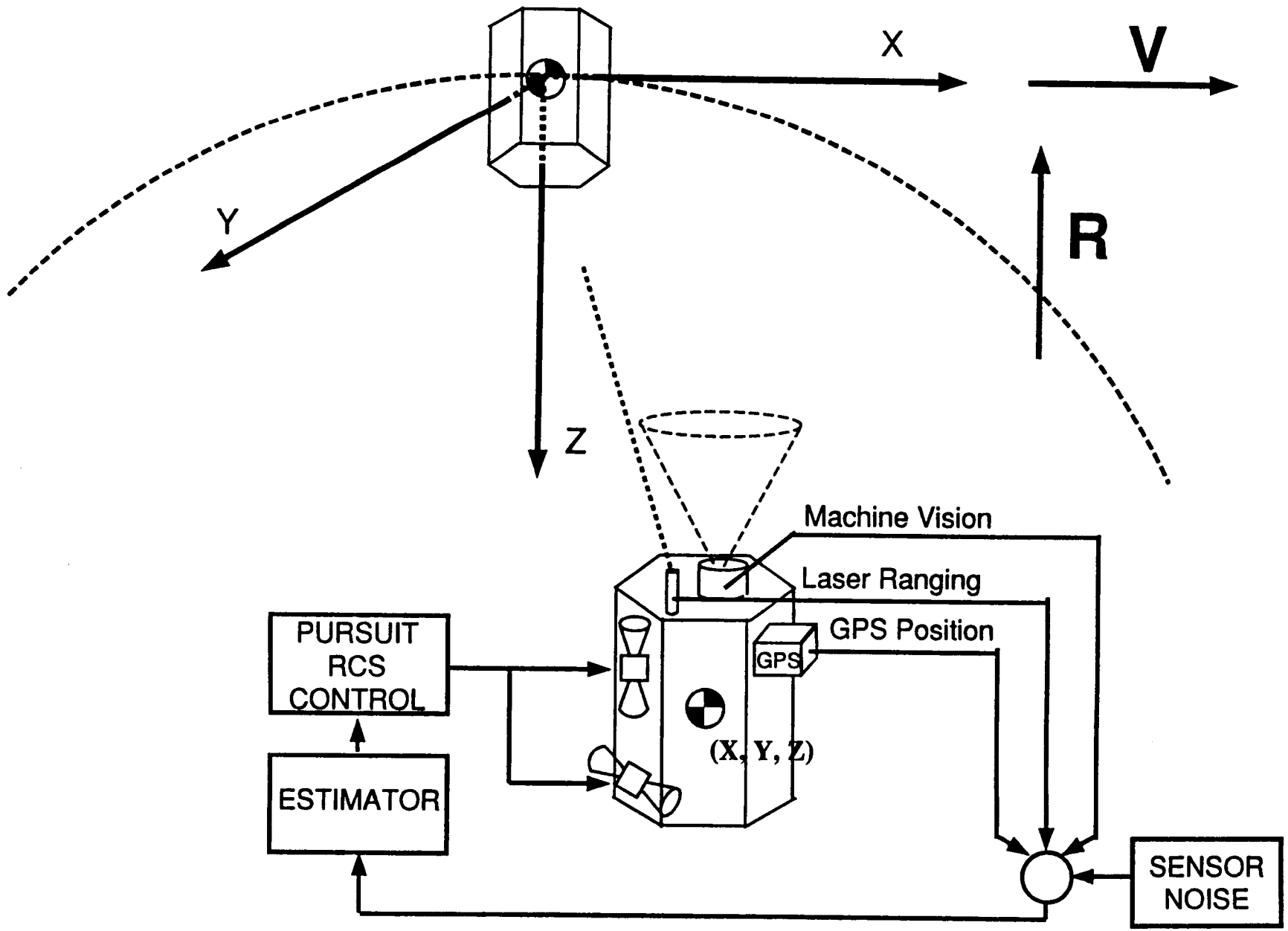
Dynamics Described in the Target Vehicle's Reference Frame by the "Euler-Hill" Equations:

$$x'' - 2\omega z' = A_x$$
$$y'' + \omega y = A_y$$
$$z'' + 2\omega x' - 2\omega z = A_z$$

4 States in Dynamic Model: Relative (X, Y, Z) Position of Pursuit Vehicle with respect to the Target and Propellant Expenditure

Enhancements:

- Sensor Noise
- Visibility of Laser Ranging and Machine Vision
- GPS Constellation Geometry
- Model of GPS Reciever


X
V
Y
R
Z
Machine Vision
Laser Ranging
GPS Position
GPS
PURSUIT
RCS
CONTROL
ESTIMATOR
(X, Y, Z)
SENSOR
NOISE

MODEL FOR LOOSELY COUPLED CONFIGURATION

Baseline COMET Attitude Controller on Target Vehicle

2 Sensors:

- Magnetometer
- Horizon Sensor

5 Actuators:

- 2 Reaction Wheels
- 3 Magnetic Torquers

Attitude Dynamics Only:

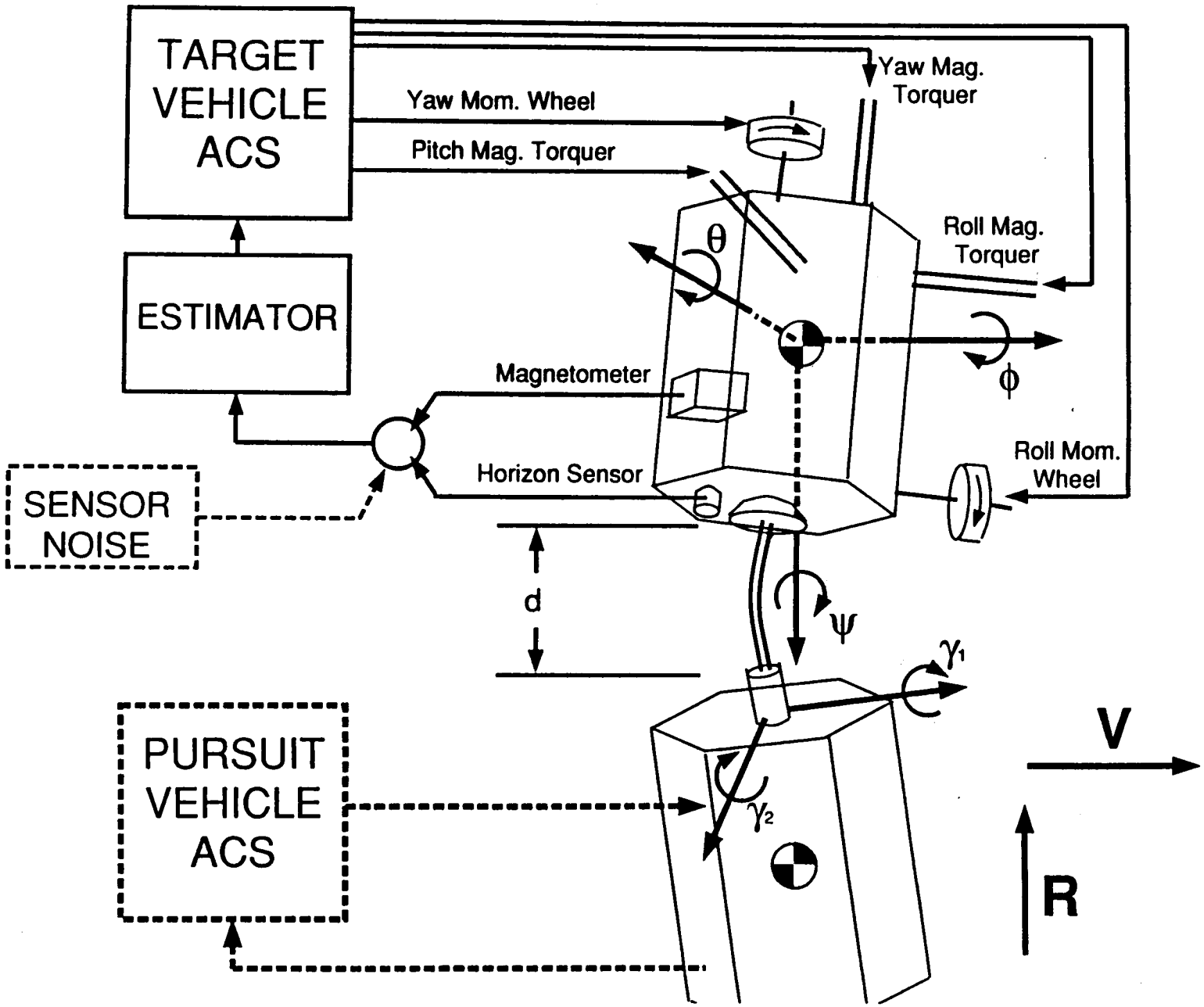
$$I\, d\omega/dt = -\omega \times I\omega + 3\omega^2 u \times Iu + M_{AERO} + M_{CONTROL} + M_{DOCK}$$

6 States in Dynamic Model:

- 3 Attitude of Target Vehicle
- 1 Length of Retraction of Probe
- 2 Bending of Probe

Enhancements:

- Sensor Model: Horizon sensor visibility, Magnetic Field Model
- Sensor Noise
- External (Random) Disturbances


TARGET VEHICLE ACS
ESTIMATOR
SENSOR NOISE
PURSUIT VEHICLE ACS
Yaw Mom. Wheel
Pitch Mag. Torquer
Yaw Mag. Torquer
Roll Mag. Torquer
Magnetometer
Horizon Sensor
Roll Mom. Wheel
θ
ϕ
ψ
γ1
γ2
d
V
R

MODEL FOR CONTACT DYNAMICS

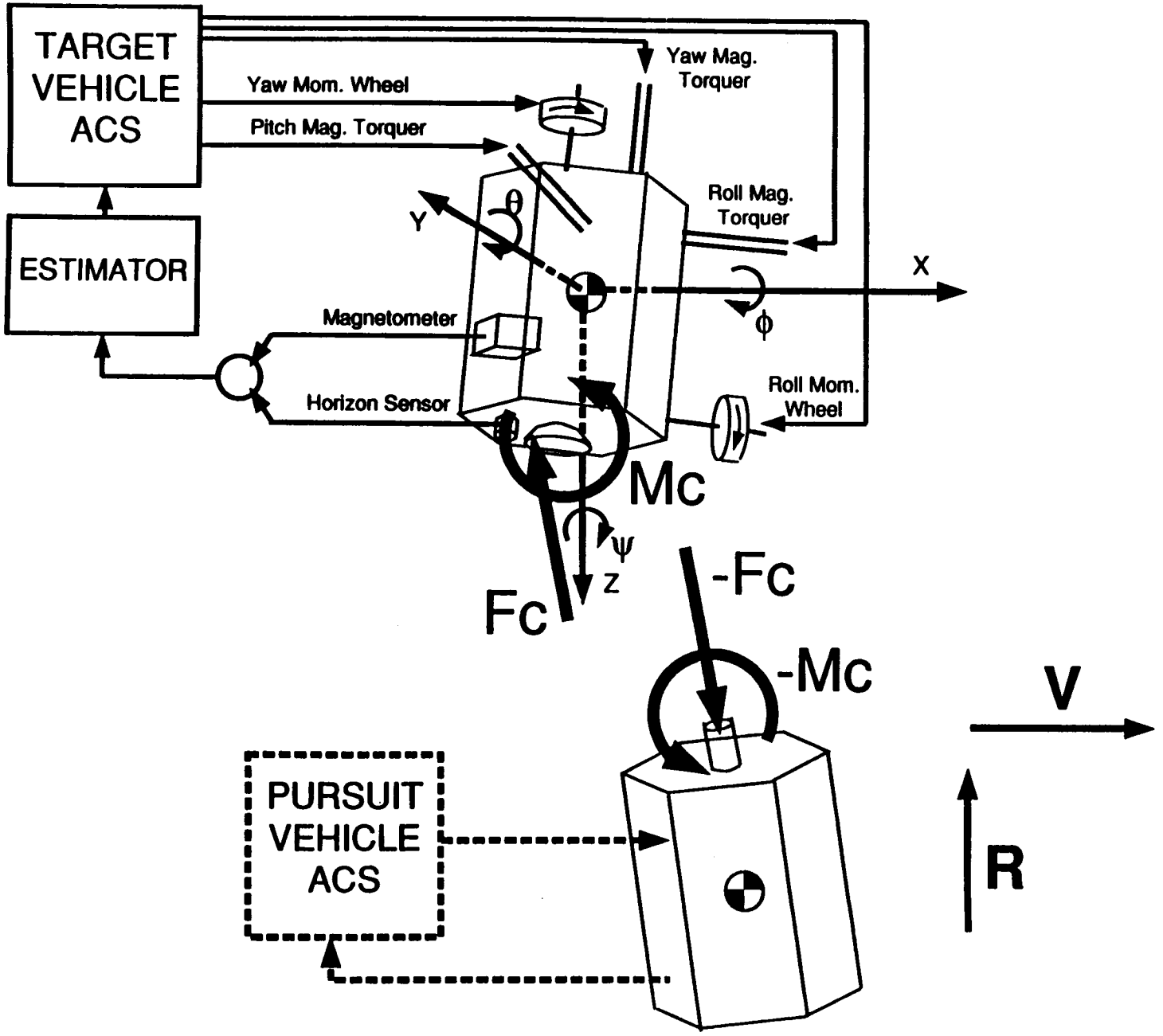
Attitude Control of Target Vehicle Alone

Same Spacecraft Model and ACS as for Coupled Control

Analytical Model for Contact Forces and Moments

Experimental Contact Force Estimates:

- Air Bearing Vehicles Tested by SpARC
- MSFC Mechanisms Laboratory
- Full Simulation with Two Air Bearing Vehicles on MSFS Flat Floor


TARGET VEHICLE ACS
ESTIMATOR
Yaw Mom. Wheel
Pitch Mag. Torquer
Yaw Mag. Torquer
Roll Mag. Torquer
Magnetometer
Horizon Sensor
Roll Mom. Wheel
Y
θ
X
ϕ
Mc
ψ
z
Fc
-Fc
-Mc
V
R
PURSUIT VEHICLE ACS

APPLICATION TO SPACE INFRASTRUCTURE

In-Space Demonstration of AR&D Technology

Validation of Analytical (and Ground Based Hardware ?) Models Against Actual Flight Data

Feasibility Studies for Use of Small Logistics Carriers

Definition of Uniform Standards and Common Interfaces for AR&D (along with AIAA)

CONCLUSIONS AND FUTURE PLANS

Need for Autonomous Rendezvous and Docking Technology Development Identified

Analytical Support Available

Potential for Ground-Based Testing and Simulation at MSFC

Opportunity for Flight Experience, Model Validation, and Data Exchange through Participation in COMET ARD Mission

REFERENCES

1. Montgomery, et. al., "Space Robotics - Recent Accomplishments and Opportunities for Future Research", NASA TM-107675, September, 1992.
2. APT Information Note 89-1, March 13, 1989.
3. Kohrs, Richard H., Letter to Deputy Director, Space Station Freedom Program and Operations, November 30, 1992.
4. Tobbe, P., and Naumann, C., "Automated Rendezvous and Capture at NASA / MSFC", Presented at the AIAA Space Programs and Technologies Conference, Huntsville, AL, March 24-27, 1992.
5. Bryan, T. C., Roe, F., and Coker, C., "Automated Rendezvous and Capture Development Infrastructure", Presented at the AIAA Space Programs and Technologies Conference, Huntsville, AL, March 24-27, 1992.
6. Commercial Experiment Transporter Customer's Operational Guide, Center for Space Transportation and Applied Research, March 23, 1992.
7. Apley, D., "Autonomous Rendezvous and Docking", presented to the Langley Research Center, November 10, 1992.
8. Kader, J. B., "Standards for Space Automation and Robotics", presented at the AIAA Space Programs and Technologies Conference, Huntsville, AL, March 24-27, 1992.
9. Leonard, C. L., and Bergmann, E. V., "A Survey of Rendezvous and Docking Issues and Developments", published in Advances in the Astronautical Sciences, Volume 69, April 24-27, 1989.
10. Govin, B., Lacombe, J. L., Fehse, W., "Guidance and Attitude Control During the Final Approach of an Autonomous Rendezvous Process", presented at the 34th Congress of the International Astronautical Federation, Budapest, Hungary, October 10-15, 1983.
11. Participating in Commercial Space Ventures, Office of Commercial Programs, May 1990.
12. Tchoryk, P., Apley, D. J., Jr., Conrad, D. J., and Dobbs, M. E., "Autonomous Rendezvous and Docking - A Commercial Approach to On-Orbit Technology Validation", Presented at the Autonomous Rendezvous and Capture Review, Williamsburg, VA, November 21, 1991.

Modeling and Experimental Verification of Single Event Upsets

T. N. Fogarty, J. O. Attia, and A. A. Kumar
Prairie View A&M University

and

T. S. Tang and J. S. Lindner
Texas A&I University

MODELING AND EXPERIMENTAL VERIFICATION OF SINGLE EVENT UPSETS

T.N.Fogarty, J.O.Attia and A.A.Kumar
Laboratory for Radiation Studies
Prairie View A&M University
and
T.S.Tang and J.S.Linder
Texas A&I University

Abstract

This paper reviews the research performed and the results obtained at the Laboratory for Radiation Studies, Prairie View A&M University and Texas A&I University, on the problem of Single Events Upsets, the various schemes employed to limit them and the effects they have on the reliability and fault-tolerance at the systems level, such as robotic systems.

Introduction

Random access memory (RAM) based on CMOS technology has gained wide acceptance in space applications [1][2]. It is known that CMOS Static RAMS show an upset sensitivity to single energetic heavy ions including gold, krypton and bromine which is called a single event upset (SEU). Immunity from SEU errors caused by protons or heavy ionizing particles is a requirement for reliable spaceborne integrated circuits. Computer simulation is an important mean to predict, analyze and verify the affects of SEUs on SRAMs. Figure 1a shows the circuit description of a six-transistor SRAM cell which has been used in SEU analyses and computer simulations. When the logic state is set such that node A is biased at Gnd and node B is at V_{dd}, the drain junctions of M1 and M4 are sensitive regions. If an ionizing particle hits the junction of M1, holes will be collected, resulting in a positive voltage spike at node A called the n-hit. Similarly if an ionizing particle hits the junction of M4, electrons will be collected, resulting in a negative voltage spike at node B, called the p-hit. If the voltage spikes are of sufficient amplitude and charge neutrality cannot be established fast enough through the 'ON' transistor, the flip-flop may regenerate and a bit error will occur [2].

Upset rate in CMOS RAMS can be reduced in two ways, either the charge collection capability of the memory can be degraded, or cell design can be altered to require greater critical charge for upset. Diminished coupling between the inverter pair of basic RAM cell decreases the probability of logic upset by slowing the feedback of gate voltage variations. If the gate voltage of the hit inverter remains stable during current impulse, the hit inverter will reestablish its prehit logic state. Maintenance of the hit inverter gate voltage at or near the prehit level is accomplished by maintenance of the voltage at the opposite information node. Thus the upset sensitivity is decreased by increasing the time constant of the feed back paths between the inverters by adding feedback resistors.

Experimental results obtained at the BNL Twin Tandem Van de Graaff Single Event Facility (Figure 1b) are presented for 16K, 64K and 256K Rad Hard Static RAMs(SRAM) [3][4]. Prior total dose radiation (1 $MRAD_{Si}$ of 2 MeV protons)

**We would like to acknowledge R.Kohler AT&T BL, P.M.Kibule & V.Zajic (BNL) formerly at Hampton University and our graduate students.*

***Partially supported by NASA-JSC-NAG-9-659, 9-331*

produces imprinting in the Rad Hard SRAM and significantly lowers the SEU LET threshold. In commercial, non-rad hard devices, one would expect a reduction in threshold LET at much lower doses. These results are compared to SPICE models of SRAM with Resistive Feedback to limit Single Event Upsets (SEU). Passive/Active resistance networks, switched capacitor networks for limiting SEU are modeled and compared. The MOS transistor active resistance feedback method of limiting SEU has the advantage that in the absence of Cosmic Ray induced charge, the operation of SRAM is not degraded by high resistance of the active network [6-17].

An extension of this work considers the distribution and dependency of these radiation induced defects to estimate the reliability and fault-tolerance of more complex systems such as A/D converters and CPUs (errors in instruction set and in active logic) [21][22].

BNL Twin Tandem Van de Graaff SEU Test

The beam energy and LET are monitored by four silicon surface-barrier detectors calibrated by an Am-241 alpha source, which makes the energy measurement independent of the information from the accelerator control room. The irradiation chamber contains an adjustable iris aperture and a mobile test board. The effective LET can be varied by rotating the board on the vertical axis to change the particle angle of arrival (Figure 1b). The effective LET is given by:

$$LET_{eff} = \frac{LET}{Cos\theta}$$

where theta is the angle of incidence of ions. A unique feature of the facility is laser optics for accurate device positioning into the beam. A neon laser is placed in the beam line, so that the beam spot can be simulated on the test board by visible light before the experiment. Evacuation and ventilation procedures are fully automated. Device positioning, beam diagnostic, data collection, and data management are computer controlled by user friendly menu driven software. The system specifications are summarized in Table 1.

Table 1. Brookhaven SEU Test Facility Specifications

Flux	10^1-10^6ions/cm^2/sec
Beam Uniformity	90-98% over 3 in. diameter
Beam Aperture diameter	0.1-1.4 in.
Test board work area	6 by 9 in.
Particle angle of Arrival	0°-73°
Effective LET in Si	1.4-280 MeV cm^2/mg
Cost	$520/h $420/h for exempt users

Radiation Hard Technology

The devices were fabricated by AT&T-Bell Laboratories CMOS Twin-Tub IV [4] "1.25um" Rad-Hard technology. As the density of SRAM increases from 16K to 64K and 256K, decrease in lateral dimensions results in lower critical charge due to lower gate and junction capacitances. In order to compensate for this increase in SEU susceptibility, the following process modifications were made;

1) Oxide thickness was reduced and doping density was raised to increase specific capacitances.
2) A twin-Tub process on thin epitaxial substrates reduced the charge collection by cutting off the funnel.
3) Feedback polysilicon resistors were used to reduce the SEU rate by

longer decoupling times for the two inverters.

In addition, radiation hardened gate oxide minimized threshold voltage shifts and transconductance degradation, radiation hardened field oxide eliminated parasitic leakage currents, and modified light doped drain (LDD) N-channel transistors minimized hot carrier effects.

A "2um" design rule 16K, and three "1.25um" design rule (a 64K , a 256K and a 32K x 8) were available for our experiments. The feedback resistor values at room temperature were 82, 109, 151, and 240 Kohms for the "2um" SRAM and 520 ND 670 Kohms for "1.25um" SRAMS. Other important processing and design parameters are given in Table 2.

Table 2. Processing and design parameters

	"2 um"	"1.25 um"
Channel width/length [um]		
NMOS	5.0/2.0	1.50/1.25
PMOS	4.25/2.0	1.75/1.50
Drain area [um^2]		
NMOS	50.3	8.9;7.0
PMOS	25.5	4.3;5.4
Gate oxide thickness [A]	215	240
Epitaxial thickness [um]	1.7	1.7
Surface doping density [$\#/cm^3$]		
N-substrate	$4x10^{16}$	$4x10^{16}$
P-well	$1.5x10^{17}$	$1.5x10^{17}$

SEU Test Experimental Set-up

Two heavy ion beams were employed in our experiments, ^{81}Br and ^{197}Au. Most of our measurements were performed at elevated temperatures 80-125°C. An individual temperature controller described in [5] was used for each DUT. The chip was plugged into a home-made DIP socket with some extra pins on each side, and thermally coupled with a power resistor from underneath with high-temperature epoxy or heat-sink grease. The temperatures were measured by a precision ACE-48006 thermistor attached with the same compound from the top. The required power was less than 10W for 20 pin SRAM chips, and the nominal power of the heating element was even smaller than that. In the experiment, temperature was controlled within $\pm$1°C.

The experiment was done using a MOSAID Memory Tester with 256K memory depth. Two testing modes are possible with this memory tester, static and dynamic. In static testing a bit map of errors is available but multiple hits are not recorded. In dynamic testing, the memory is checked for SEU during irradiation. If an error is found, the error counter is incremented by one and the error is corrected. Consequently, the error map is no longer available as it was in static testing. 16K, 64K & 256k CMOS SRAMs with feedback resistors performed satisfactorily in our test fixture down to 3.5 V. in both static and dynamic testing. However, preliminary measurements indicated a considerably higher SEU cross section in dynamic testing (Figure 2).

Because of cable capacitance (8 ft of flat cable between the memory tester, vacuum feedthrough, and the DUT), the constant memory reading/loading in dynamic testing resulted in 1 V noise on the power supply line. The lower average power supply voltage decreased the critical charge and increased SEU sensitivity. It is also seen in Fig.2 that dynamic testing performed at 5.0 V

exhibited the same critical LET as static testing at 4.5 V. By adding a fast tantalum capacitor 0.47 uF close to the DUT, the noise was reduced to 0.1V and the SEU cross section decreased. Unfortunately, we did not collect enough data to determine the critical LET. While dynamic testing is closer to device field operation, the measured SEU cross section can be substantially varied depending on the capacitive coupling between power supply and ground. To avoid this variation, all subsequent measurements were performed in the static mode. The power supply voltage was 5.0V and 4.5V for 16K and 64K SRAMs, respectively.

When a single memory cell is upset twice during the static testing, no error is recorded. If the same memory cell is upset one more time, only one error is recorded and so on. Clearly, the observed number of SEU is smaller that the actual number due to the possibility of multiple upsets of the same memory cell. Depending on the parity, the multiple upsets either escape observation entirely or are recorded as a single SEU. The following correction was applied to all experimental data obtained by static testing to account for multiple upsets:

$$P'_{SEU} = P_{SEU} - P^2_{SEU} + P^3_{SEU} - \ldots = \frac{P_{SEU}}{1-P_{SEU}}$$

where the probabilities p'_{SEU} and p_{SEU} are given by the observed and corrected numbers of SEU, respectively, divided by the memory size (16K or 64K). In order to keep both this correction and statistical uncertainties small, it is good practice to make the number of SEU approximately 10% of the memory size by accumulating an appropriate fluence.

SEU Test Results and Discussion

No failures of any of the tested 16K SRAM (TA670) were observed at room temperature up to the effective LET = 160 MeV cm^2/mg with any value of the feedback resistors in either static or dynamic testing. The same statement applies for testing at temperatures up to 50°C.

Data obtained for four 16K SRAMs with variable feedback resistors at 110°C with Br ions using MOSAID Memory Tester are shown in Figure 3a. Data obtained for 64K SRAMs at the same temperature and with the same ions using the MOSAID Memory Tester are displayed in Figure 3b. The critical LET (defined as LET at which the SEU cross section drops by a factor of 500 compared to its saturated value, i.e. approximately 10-5 cm^2 for our 16K SRAMs) increased with the increasing value of the feedback resistance as expected. Since both feedback resistors values and the power supply voltages were different for 16K and 64K SRAMs, we find little sense in comparing results for the two design rules. All feedback resistor values are given at room temperature. They can be calculated at elevated temperatures using a known temperature coefficient for polysilicon resistors. The coefficient was measured for a resistor on the test chip made by the same technology (see Figure 4). The dependence was found to be exponential, where k is the Boltzman constant and -E = 0.0899(10) eV the activation energy.

The SEU cross sections of the 16K SRAM with 82 Kohm feedback resistors were measured at three different temperatures, 90,110,125°C, and with two ion beams, Br and Au. Results obtained with the HP-8180/82 Data Generator/ Analyzer and with the MOSAID Memory Tester are shown in Figures 5a and 5b, respectively. A difference was observed in the critical LET for the two ions. The difference is, at least partially, caused by energy loss in the passivation layer which has an equivalent thickness of approximately 1 mg/cm^2 of Silicon. The LET vs. energy curves for *Br* and *Au* ions imported from Ziegler's Tables are plotted in Figure 6. A small energy loss shows 260 MeV ^{81}Br ions gradually approaching a maximum, thus the LET remains relatively constant. On the other hand, 310 MeV ^{197}Au ions have already passed the maximum and are in the region of steep decay, thus a significant LET reduction is

expected. For example, in our second experiment using the MOSAID Memory Tester, the critical LET measured at 110°C was found to be 63 and 80 MeV cm^2/mg for 260 MeV ^{81}Br and 310 ^{197}Au ions respectively, thus reducing the difference to 10 MeV cm^2/mg. Preliminary measurements showed that LET in Ziegler's Tables is underestimated by 8% for *Br* and by 4% for *Au* in the energy regions of interest. Such a correction would reduce the critical LET difference to 8 MeV cm^2/mg. The most probable explanation of the residual difference is based on charge collection effects, where the ion track of *Br* in the device sensitive region is longer and more diffuse than that for Au, thus there is less recombination and more charge is collected.

Using the error map, the SEU cross sections measured at high angles of particle arrival were corrected for the chip package shadow (up to 40% and 6% at 0-73° for 16K and 64K, respectively) but they still tapered off. The most drastic effect was observed for the 16K SRAM with 240 Kohm feedback resistors tested at 125°C with *Au* ions (see Figure 7). A sharp maximum occurred around the effective LET = 143 MeV cm^2/mg and the SEU cross section dropped to zero above 180 MeV cm^2/mg. The corresponding angle of incidence were 56° and 63°. After penetrating the passivation layer, the effective LET became 121 and 139 MeV cm^2/mg at these two angles. The calculation proves that the energy loss itself is insufficient for the explanation of the observed data. At high angles of incidence, the collected charge is probably shared by two or more neighboring nodes and becomes insufficient to upset either one.

After initial SEU testing, one of the 64K SRAMs was exposed to 2 MeV protons. A total dose of 1.3 MRad(Si) was accumulated, 0.65 MRad(Si) without operating bias and 0.65 MRad(Si) in the memory state "all 0". After proton irradiation, the SEU test was repeated in both "all 0" and "all 1" memory states. While SEU cross section in the state "all 1" did not differ from its pre-radiation value, the SEU cross section in the state "all 0" showed a slight increase (see Figure 8). The SRAM was found to prefer the state in which it was irradiated. Ionizing radiation induces bias dependent threshold voltage shifts and mobility degradation which cause a CMOS SRAM cell imbalance. Since the most sensitive strike location, for the present technology, is the OFF P-channel drain which is restored through an N-channel transistor, the cell imbalance is defined as the difference between N-channel threshold voltage shifts. As the 64K, 256K and 32K x 8 SRAMs utilize the same "1.25um" process and the same device geometry one would expect similar SEU response. The 64K and 32K x 8 do in fact show similar response (see Figure 9). However, a greater feedback delay time for the same LET threshold is apparent in the 256K. This can only be attributed to circuit pattern effects [4a].

Hardening Approaches for CMOS SRAM

Resistive Hardening

SPICE simulation of SRAM cells with feedback resistors between the inverters [12][13][4a] shows agreement with critical LET threshold considering the decrease in delay as the polysilicon resistor value decreases with temperature increase thus decreasing LET threshold. Cell write delay times are shown in Figures 10 and 11 for the "1.25um" and "2um" processes respectively. The SRAM cells with feedback resistors are shown in Figure 12a. For the "2um" process SRAM disturbed by a 1 mA exponential pulse, the results at room temperature and 87°C are summarized in Table 3.

Table 3. 16K SRAM, 1mA Exponential Pulse

Temp.	Resistance	Max. Wid.to Q_c
27°C	82K	0.8nsec
87°C	46K	0.3nsec

Integrating the current pulse one finds an estimate of the disturbing charge

necessary to reach critical charge and cause an SEU in the SRAM cells at room and elevated temperatures. Clearly less charge is needed to cause an SEU at elevated temperatures.

Capacitive Hardening
In the MOSIS "2um" process the specific resistance of the polysilicon resistors is much lower and simulation of the distributed RC feedback shows that the resistor value may be decreased for equivalent LET threshold. However because of area considerations most commercial vendors will use higher specific resistance and shorter polysilicon resistor length negating the need for this correction [14].

Simulation of the insertion of a capacitance between the drain to gate nodes shows equivalent LET threshold and superior speed when compared with the feedback resistor approach. In this approach the capacitance is not in the write path of the cell. The write time is increased for this approach as the inserted capacitance increases the node capacitance. But this increased time is much smaller than the resistive hardening concept (see Figure 12b)[15].

The critical charge is 4.8 pC for a capacitance of 0.1pF for a rectangular current pulse of 3ns width and 1.6mA amplitude. For a 100 Angstrom oxide thickness the area for this capacitance is nearly equal to the area of an NMOS transistor. Speed performance becomes a problem as larger scales of integration are required. We conclude that the increased area for radiation-hardening can be sacrificed for better speed performance.

Active Hardening
A CMOS Transmission Gate (TG) exhibits a nonlinear current-voltage characteristic when it conducts; hence called nonlinear active resistor. The TG resistance strongly depends on its terminal voltage: increasing rapidly as the terminal voltage increases. This phenomenon can be utilized to increase SEU immunity of a SRAM cell. A SEU hardened CMOS SRAM cell using TGs as feedback resistors is shown in Figure 12c. In this cell, the inverter pair is decoupled by two TGs whose p-channel and n-channel transistors are respectively gated by the ground and power source. These transistors provide the resistance needed for increasing critical charge of the cell and also introduce additional capacitance to the sensitive nodes and feedback paths of the cell, which can effectively increase SEU immunity of the cell [16].

Operation of the cell can be described briefly as following. When the cell operates normally, the resistance of the two TGs is very low since the voltages across the gate terminals are very small. The cell is essentially an unhardened one. When one of the sensitive nodes is hit by an ionizing particle, electrical charges are collected at the hit node, causing a sudden voltage increase or decrease at the hit node while the voltages at other nodes are relatively unaffected. In response to the voltage increase across the terminals of the TG connected to the hit node, the resistance of the TG becomes very high. The high feedback resistance protects the stored cell data from SEU.

Effectiveness of the new SEU-hardening technique was studied numerically. The current induced by a particle hit was simulated by an exponential pulse. The rise and fall time constants of the exponential pulse were set equal to 0.01ns and 0.25ns respectively.

Simulations showed that the new technique improved the SEU immunity of a SRAM cell effectively. With TGs whose channel length was 12um and width was 1.2um, a CMOS SRAM cell did not upset from a current pulse with amplitude of 10mA and width of 1.4ns, as seen in Figure 13a. An estimate for this TG channel resistance is about 100 Kohm. Prevention of upset from the same current pulse required a passive feedback resistor of 205 Kohm. Figure 13b shows the simulation result. We believed that it was the combination of distributed channel resistance and parasitic capacitance, mainly gate capacitance, making

TGs more effective than lumped resistance in increasing critical charge of the cell. This observation agreed with our findings in [14] where distributed RC feedback was shown to be better than lumped RC and pure resistive feedback.

Switched Capacitor SRAM

It can be shown that if clock frequency is high enough, the combination of switches and capacitor can replace a resistor that is dependent only the clock frequency and capacitor [18][19][20].

The switched capacitor network employed in this work is single-phase grounded switched capacitor shown in Figure 14a. It consists of two switches and a capacitor. For SPICE simulations, the equivalent circuit that can be used for the single phase switched capacitor network is shown in Figure 14b.

The switched capacitor network was implemented using MOS technology. The circuit is shown in Figure 15. PSPICE simulations were performed. Samples of the output are shown in Figures 16,17 and 18.
The rise time, fall time, propagation delay, and time shift (T-shift) [i.e. the time required for the output to occur when the input signal is applied]. Table 1 shows the switching times for SRAM with no feedback resistors, switched capacitor SRAM and SRAM with feedback resistors of 40K, 80K, and 150K.

From Table 4, it can be seen that the rise time, fall time, propagation delay and T-shift of the switched SRAM and SRAM without feedback resistance are small compared to those of SRAM with feedback resistors. In addition, switching times of the switched capacitor SRAM are similar to those of SRAM without feedback resistors.

Table 4. Switching Times of SRAMs

Feedback Value	Fall Time * E-9	Rise Time * E-9	Prop. dly * E-9	T-Shift * E-9
SRAM w/out feedback	0.4087	0.380	0.395	0.765
SC SRAM	0.280	1.029	0.654	0.769
40K	1.70	1.581	1.64	2.296
80K	3.15	3.121	3.135	4.019
150K	5.73	5.503	5.616	6.972

Systems Level Analysis

This section addresses the problem of reliability and fault-tolerance from a systems point of view. Due to space considerations we will only provide a broad description of our approach and model. We view the system in a hierarchical fashion, viz., the system is considered as a collection of functional units, each functional unit made up of several functional subunits, and so on to the primitive elements (Figure 19). The black circles in the figure refer to "irreducible modules" -- subsystems that cannot be further subdivided into smaller systems. The white circles refer to "reducible modules."

At any given level of description, the system is seen to have a set of modules of each type. For instance, a private branch exchange system (PBX) consists of several "sequential" and "parallel" subsystems (see Figure 20). Another example is the Intel 80386 microprocessor consists of nine logical units: bus interface unit, prefetch unit, instruction decode unit, execution unit, control unit, data unit, protection test unit, segmentation unit and paging unit. The execution unit is further divided into ALU and the 32-bit register files. Each of these in turn can be broken down into gates and then to

transistors. Two important aspects of our approach are the inclusion of:

1. "correlational dependence'' of failure rates; for instance, the occurrence of one failure may accelerate the occurrence of another; and
2. explicit modeling of failure rates in terms of the geometrical, circuit and material parameters.

Each such unit is then modeled in terms of rate equations that would allow a computation of the MTTF. The reliability at any given level will be calculated by a generalized Markov model. The knowledge of MTTF for each subsystem may then be used in two ways: (i) from the parametric dependence of the MTTF, to choose different material combinations to maximize the MTTF; and (ii) to replicate those subsystems with a lower MTTF, so that the entire system may degrade gracefully. This approach is obviously better than assigning an average MTTF for the whole system.

Formulation

The general approach in terms of a non-Markovian model is given elsewhere [21-23]. In the following we discuss a simplified version, the so-called memoryless or Markovian approximation. In this limit, which is the most common limit used in the literature [24-25]. The corresponding equations are:

$$\frac{dP_i}{dt} = -\alpha_{i,i+1}P_i + \beta_{i+1,i}P_{i+1} + \alpha_{i-1,i}P_{i-1} - \beta_{i,i-1}P_i - \gamma_{i,f}P_i,$$

for i = 1,2,...,N-1, and

$$\frac{dP_0}{dt} = -\alpha_{0,1}P_0 + \beta_{1,0}P_1 - \gamma_{0,f}P_0, \quad \frac{dP_N}{dt} = \alpha_{N-1,N}P_{N-1}, \quad \frac{dP_f}{dt} = \sum_{i=0}^{N-1}\gamma_{i,f}P_i$$

where P_i is the probability that the subsystem is in the state S_i, $a_{i,j}$ is the transition (failure) rate from state S_i to state S_j, $b_{j,i}$ is the transition (repair) rate from state S_j to state S_i and $g_{i,f}$ is the transition (failure) rate from state S_i to the fatal state S_f. S_f denotes the "fatal failure state."

Solution

The above equations may be solved iteratively to yield the reliability $R(t)$ and the $MTTF$:

$$R(t) = \sum_{i=0}^{N-1} P_i(t)$$

$$R(t) = L^{-1}[R(s)] = L^{-1}\left[\sum_{0}^{N-1} P_i(s)\right]$$

$$MTTF = \xrightarrow{\lim s \to 0} \sum P_i(s)$$

(s is the LaPlace Transform variable).

Sample Application

Let us apply this general solution to the case of a system with triple-modular- redundancy (TMR, see Fig. 21). That is, N=3. In this case, units

A, B and C are identical units, while unit D may be an identical or a different unit. The system will function so long one of the former three units are functioning, but the moment unit D fails, the whole system fails. The failure of D is what is termed fatal or common-cause failure. [An electrical realization of this system is a typical buffer between two large circuits. A, B and C could be simple inverting buffers (INRBs) while D may be a super-inverting buffer (INRBS).] The MTTF for this case is given by,

$$MTTF = \frac{\Delta_1\Delta_2 - \beta_{2,1}\alpha_{1,2} + \alpha_{0,1}\Delta_2 + \alpha_{0,1}\alpha_{1,2}}{\Delta_0\Delta_1\Delta_2 - \Delta_0\alpha_{1,2}\beta_{2,1} - \Delta_2\alpha_{0,1}\beta_{1,0}}$$

Special Cases

Case	MTTF	Comment
$\alpha_{ij} = \alpha,\ \beta = 0,\ \gamma = 0$	$3/\alpha$	Three times the survival rate of a single component (or module), as should be expected
$\alpha = \gamma$	$7/8\alpha$	Less than the survival rate of a single component, i.e., inclusion of redundancy does not necessarily augment the reliability of the system in the presence of fatal faults
$\alpha_{0,1} = \alpha, \alpha_{1,2} = k\alpha, \alpha_{2,3} = k^2\alpha$ $\gamma_{i,\phi} = \alpha, \forall i$	$\frac{(k+1)^3 - k^2 - k + 1}{2(k+1)\left(k^2+1\right)}\left(\frac{1}{\alpha}\right)$	k=1, MTTF=7/8α, previous result k>1, the MTTF is much smaller. k=2, MTTF=11/15α

Failure Rate Models

Failure Phenomenon	Failure Rate
Electromigration	$\alpha \propto AJ^n \exp\left(-E_a / k_B T\right)$
Time-Dependent Dielectric Breakdown	$\alpha = AQ_{bd}^{-1}E_c^2 \exp\left(-B/E_c\right)$
Thermal Breakdown	$\alpha = \frac{P_f}{\left(\pi K \rho C_p\right)^{1/2} A_j \left(T_c - T_0\right)}$

Similar rates may be constructed for single-event-upsets, stuck-at-faults, etc.

Summary and Conclusions

1. No SEU of AT&T RAD-HARD SRAM was observed at Room Temperature.

2. Critical LET decreased with increasing temperature and/or decreasing value of feedback resistors.

3. Critical LET was different for Bromine and Gold ions because of Zeigler Curve effects and the heavy ion track structure.

4. Imprint of the memory pattern after 1.3 $MRad_{Si}$ TID of protons was consistent with the threshold voltage shift of NMOS transistors. This decrease in critical LET threshold is expected to be significant for non-RAD-Hard devices at much lower prior total dose radiation.

5. SPICE simulation of SRAM cells with feedback resistors between the inverters shows agreement with critical LET threshold considering the decrease in delay as the polysilicon resistor value decreases with temperature increase thus decreasing LET threshold.

6. In the MOSIS "2um" process the specific resistance of the polysilicon resistors is much lower and simulation of the distributed RC feedback shows that the resistor value may be decreased for equivalent LET threshold. However because of area considerations most commercial vendors will use higher specific resistance and shorter polysilicon resistor length negating the need for this correction.

7. Simulation of the insertion of a capacitance between the drain to gate nodes shows equivalent LET threshold and superior speed when compared with the feedback resistor approach. However this approach is not area conservative.

8. The active resistor(TG) SEU-hardening technique for CMOS SRAMs has been identified. This technique is effective in improving SEU immunity and needs no modifications of the fabrication process.The new technique shows low resistance except when necessary to limit cosmic ray induced charge.

9. The switched capacitor SRAM, implemented using MOS technology, has characteristics similar to those of CMOS SRAM without feedback resistors. The switching times of the switched capacitor SRAM are comparable to those of SRAM without feedback resistors. In addition, it was found that the switching times of the switched capacitor SRAM are superior to those of SRAM with feedback resistors. This work shows that switched capacitor SRAM is a viable alternative to SRAM with feedback resistors for SEU immunity.

10. An attempt to formulate a unified framework to compute the reliability of a system in the presence of fatal faults and redundant elements. In particular it is showed that, the inclusion of redundancy does not necessarily enhance the reliability of the system. Though the discussion has been in the framework of electronic systems, our formulation may also be used to describe distributed and parallel processing systems. In such cases, "failure" may be interpreted as "non-availability" of a processor (perhaps owing to its being accessed during its computation cycle) and MTTF might be an estimate of the length of computational time required for a given computation.

References

[1] Kerns, Shafer, Rockett Jr., Pfidmore, Berndt and Van Vonna, Proceedings IEEE, Vol.76, No.11, Nov.1988, pp. 1470-1509.

[2] J.L.Andrews, J.E.Schroeder & B.L. Gingerich, W.A. Kolasinski & R.Koga, S.E. Diehl, "Single Event Error Immune Cmos Ram". IEEE Trans. on Nucl. Sci. Vol. NS-29, No. 6, December 1982.

[3] T.N.Fogarty, C.Herman, K.Diogu, F.Wang, "Simulation of Natural Space Radiation Effects on VLSI Technology", Electrochem.Soc.Ext.Abstracts F87, #624,pp.880-881 (1987).

[4] V. Zajic, K. Kloesel, D. Ngo, P. M. Kibuule, A. Oladipupo, T. N. Fogarty, R. A. Kohler, E. G. Stassinopulos, "Single Event Upset and Total Dose Radiation Effects on Rad-Hard SRAMs","Journal of Electronic Materials, Vol 19, No. 7, 1990.

[4a] Private Communication R.A.Kohler AT&T BL Dec. 26 1986.

[5] W.A.Kolasinski, R.Koga, E.Scnauss, and JDuffey, "The Effect of Elevated Temperatures on Latchup and Bit Errors in CMOS Devices", IEEE Trans.Nucl.Sci.NS-33,pp.1605-1609(1986).

[6] Serafin Castro, "A Study on The Effects Of a Single-Event Upset And Radiation Hardening." M.S Thesis, Texas A&I University, Kingsville, TX, December 1991.

[7] James C. Pickel and James T. Blanford, Jr., " Cosmic-Ray-Induced Errors In MOS Devices," IEEE Transactions on Nuclear Science, Vol. NS-27, No. 2, April 1980.

[8] G. C. Messenger, " Collection of Charge On Junction Nodes From Ion Tracks," IEEE Transactions on Nuclear Science, Vol. NS-29, No. 6, December 1982.

[9] S.E. Diehl, A. Ochoa, Jr. and P.V. Dressendorfer, R. Koga and W.A.Kolasinski " Error Analysis and Prevention of Cosmic Ion-Induced Soft Errors in Static CMOS RAM", " IEEE Transactions on Nuclear Science, Vol. NS-29, No.6, December 1982.

[10] R.C. Jaeger, R. M. Fox and S. E. Dieh, "Analytic Expressions for the CriticalCharge In CMOS Static RAM Cells," IEEE Transactions on Nuclear Science, Vol. NS-30, No. 6, December 1983.

[11] Richard L. Johnson, Jr. and Sherra E. Diehl, " An Improved Single Event Resistive-Hardening Technique for Cmos Static Rams," IEEE Transactions on Nuclear Science, Vol. NS-33, No. 6, December 1986.

[12] Private Communication P.M. Kibole to T.N. Fogarty Hampton University Dec. 1989.

[13] Y.Suneel, S.Castro, T.Tang, J.S.Linder, Texas A&I University & T.N.Fogarty, PV A&M LRS "Simulation Study of Single Event Upset Immune CMOS SRAM". Proceedings PV A&M CEA Symposium, March 15,1993

[14] T.Tang, Y.Suneel, J.S.Linder, Teas A&I University, "Single Event Upset Immune CMOS SRAM With Distributed RC Feedback". Proceedings PV A&M CEA Symposium, March 15,1993

[15] J.S.Linder, S.H.Khan, T.Tang, Texas A&I University, "Capacitive Radiation Hardening Approach for SEUs of SRAM Cell", Proceedings PV A&M CEA Symposium March 15,1993.

[16] T.Tang, I.P.Tadptri, J.S.Linder, Texas A&I University, "An Active Resistive-Hardening Technique for CMOS SRAM". Proceedings PV A&M CEA Symposium, March 15,1993.

[17] F.Eshete, J.O.Attia, Prairie View A&M University, "Radiation Hardening of CMOS SRAM Using Switched Capacitor Networks". Proceedings PV A&M CEA Symposium, March 15,1993.

[18] R.L.Gieger, P.E.Allen, and N.R.Strader "VLSI Design Techniques for Analogue and Digital Circuits" McGraw-Hill, Inc., 1990.

[19] C.F.Kurth and G.S.Moschytz "Nodal Analysis of Switching Capacitor Networks" IEEE Trans. on Circuits and Systems, Vol. CAS-26, No.2, Feb.1979.

[20] B.D.Nelin "Analysis of Switched-Capacitor Networks Using General Purpose Simulation Programs" IEEE Trans. on Circuits and Systems, Vol. CAS-30, No.1, Jan.1983.

[21] A.A.Kumar, "VLSI Space Systems Reliability," Proceedings of the NASA Radiation Conference, Hampton, March 1991.

[22] A. A. Kumar, "Reliability of Large Scale Systems," Proceedings of the E&A'93 Symposium, Prairie View A&M University, Texas, March 15-16, 1993.

[23] A.A. Kumar, Models of Disordered Systems, book under preparation.

[24] W.K. Chung, "A k-Out-Of-N:G Redundant System with Dependent Failure Rates and Common-Cause Failures," Microelectron. Reliab., vol. 28, p. 201, 1988.

[25] W.K. Chung, "Reliabilty Analysis of Repairable and Non-Repairable Systems with Common-Cause Failures," Microelectron. Reliab., vol. 29, p. 545, 1989.

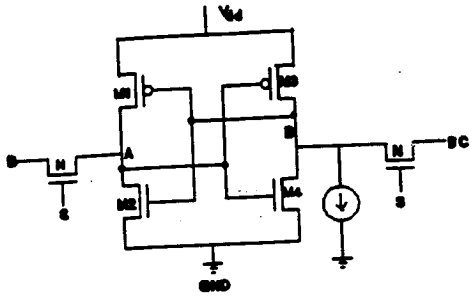


Figure 1a. A SRAM cell .

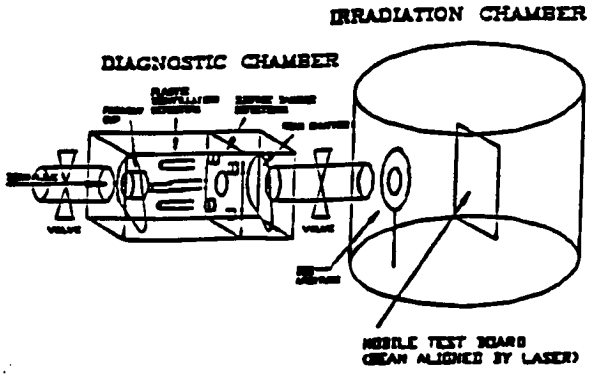


Figure 1b. Brookhaven SEU Test Facility

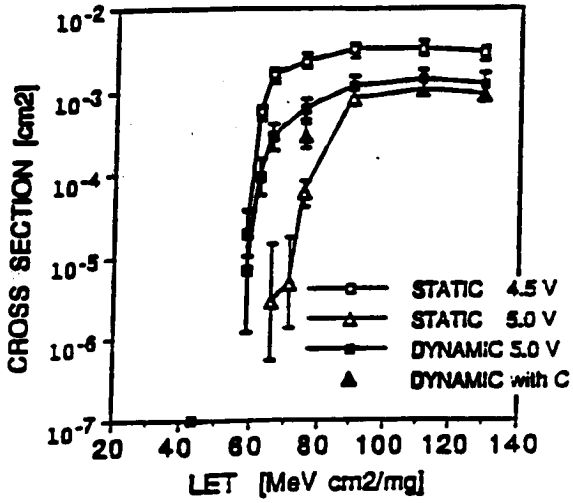


Figure 2. Effect of power supply voltage on SEU of 16K SRAM with 82 Kohm feedback resistor (90°C, Br ions). Zero SEU are displayed on the LET axis.

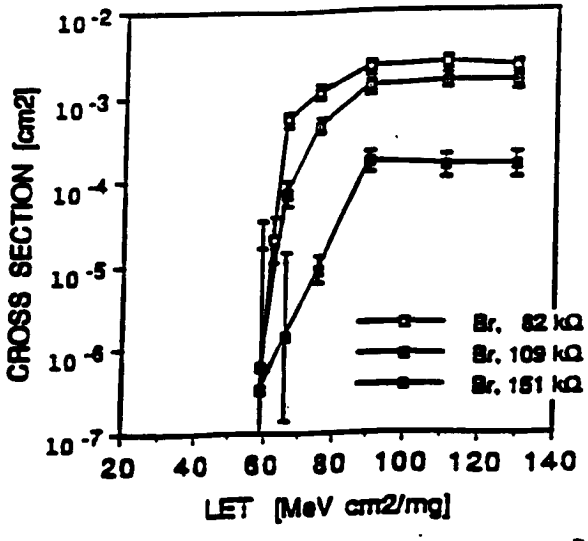


Figure 3a. SEU of 16K SRAMs with variable feedback resistors (110°C, Br ions). Data from MOSAID Memory Tester.

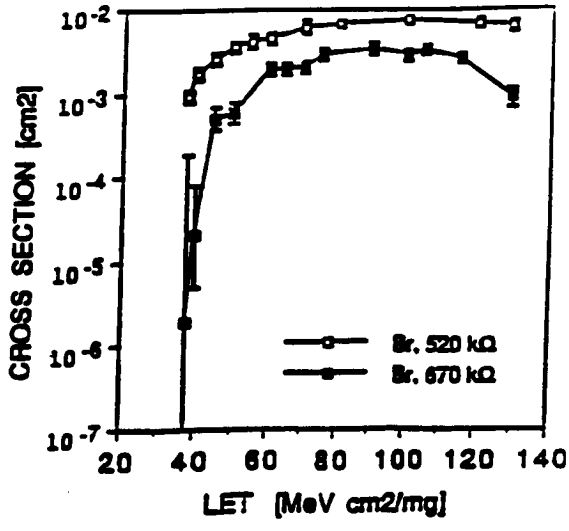


Figure 3b. SEU of 64K SRAMs with variable feedback resistors (110°C, Br ions). Data from MOSAID Memory Tester.

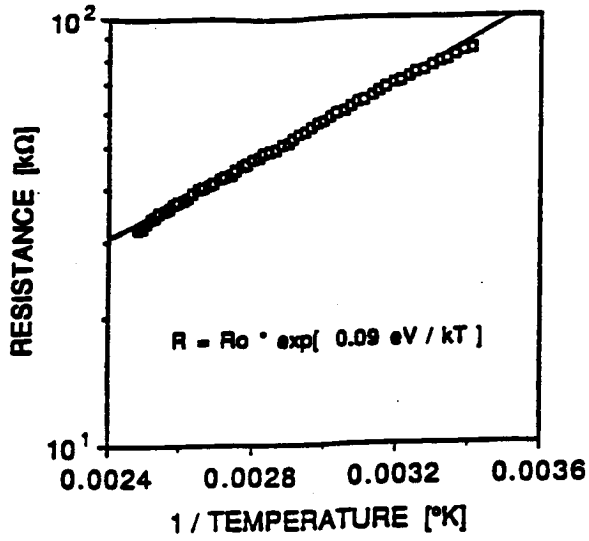


Figure 4. Temperature dependence of a polysilicon resistor.

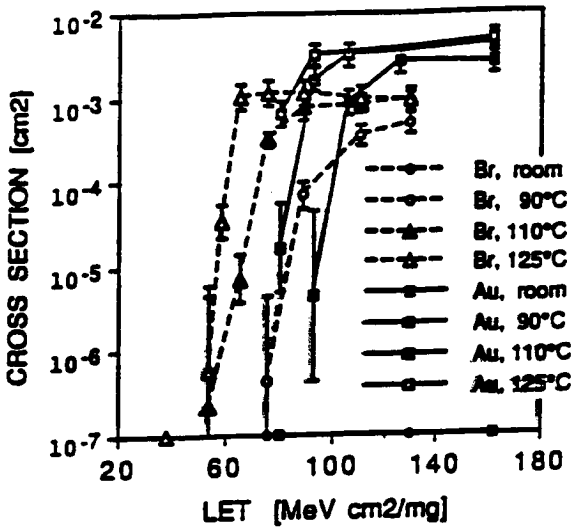


Figure 5a. SEU of 16K SRAM with 82 Kohm feedback resistors at elevated temperatures for Br and Au ions. Data from HP Data/Generator/Analyzer.

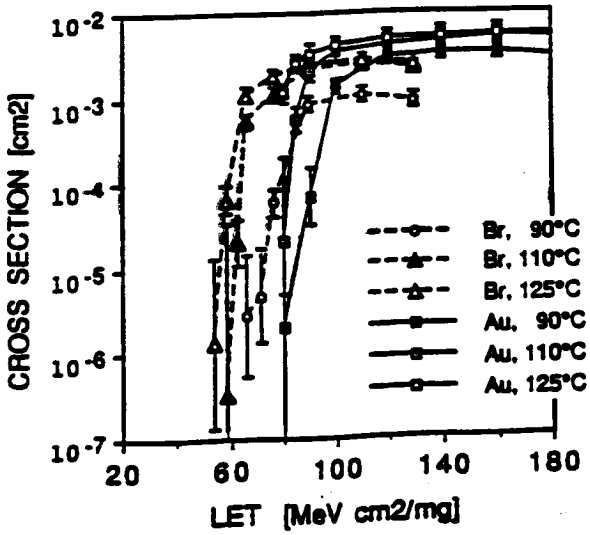


Figure 5b. SEU of 16K SRAM with 82 Kohm feedback resistors at elevated temperatures for Br and Au ions. Data from MOSAID Memory Tester.

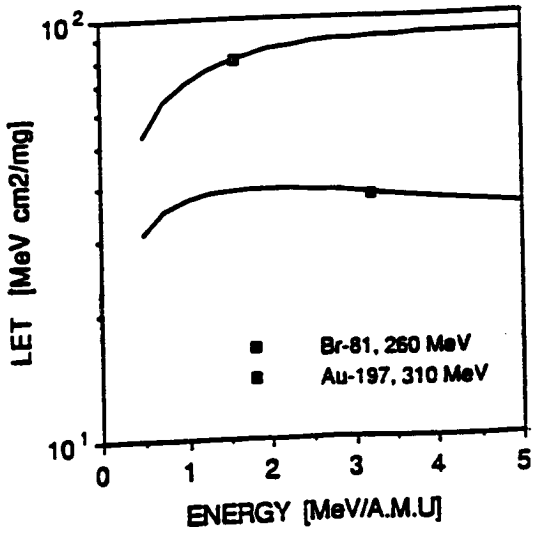


Figure 6. LET vs energy for ^{81}Br and ^{197}Au (from Ziegler's Tables).

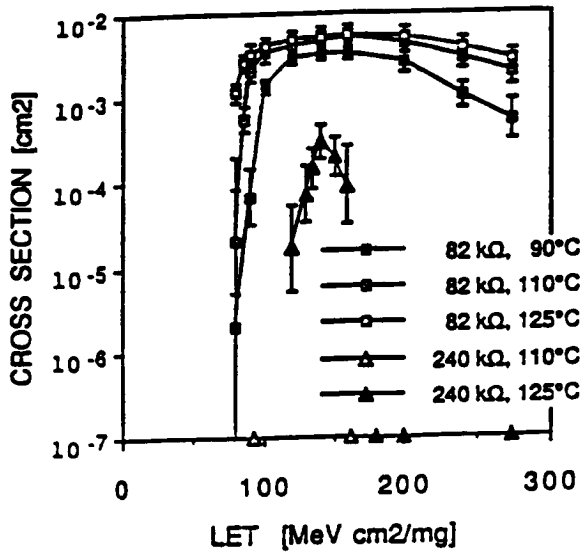


Figure 7. Decrease of SEU of 16K SRAMs at high angles of incidence for Au ions.

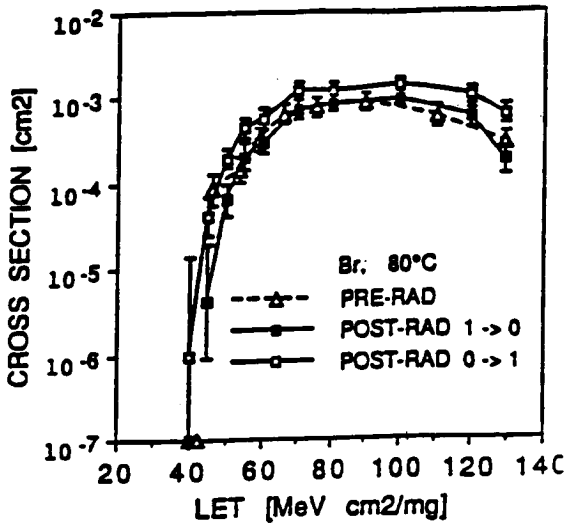


Figure 8. SEU of 64K SRAM with 520 Kohm feedback resistors before and after proton irradiation (80°C, Br ions).

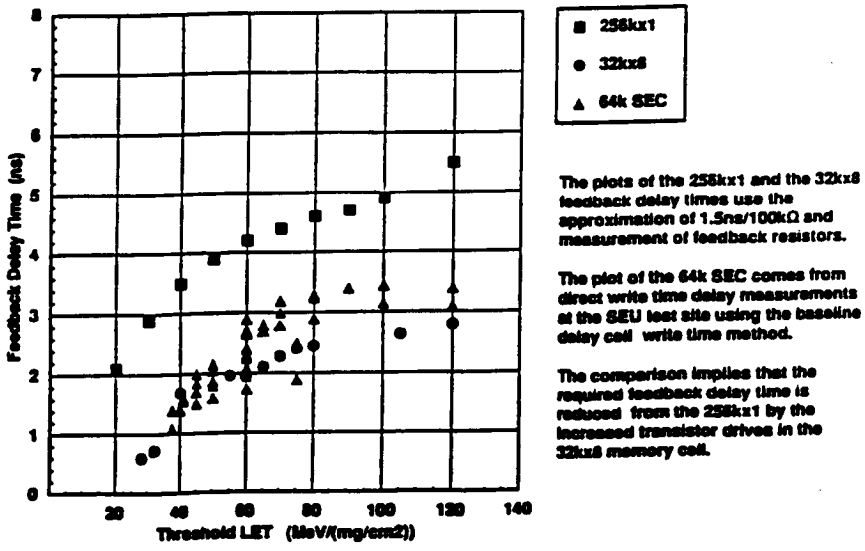


Figure 9. Comparison of Fundamental SEU Response of 32K x 8 and 256K x 1 SRAMs.

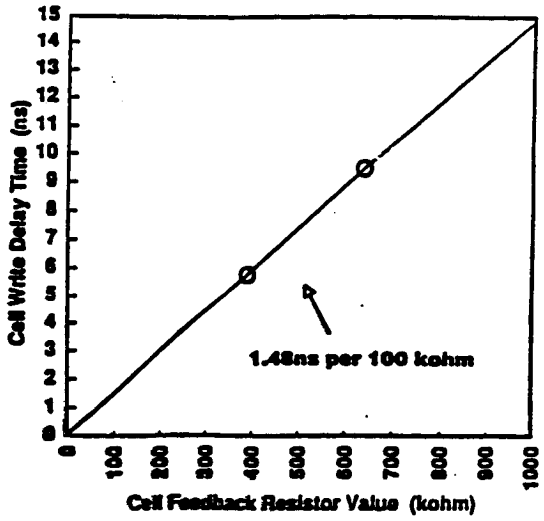


Figure 10. Feedback delay for 32K x 8 ("1.25um" process) cell produced by Cell Feedback Resistance as determined from write time method.

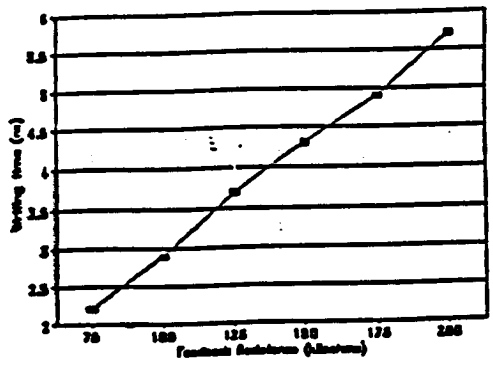


Figure 11. Write time vs. Feedback Resistance for "2um" process.

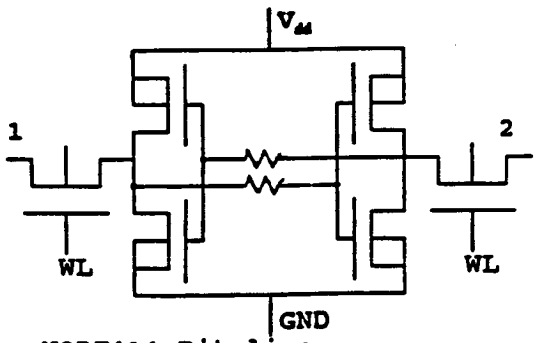


Figure 12a. SRAM Cell with feedback resistors.

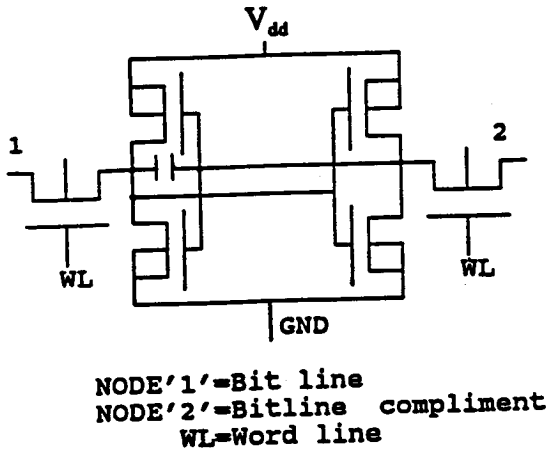


Figure 12b Schematic diagram of a capacitive-hardened SRAM.

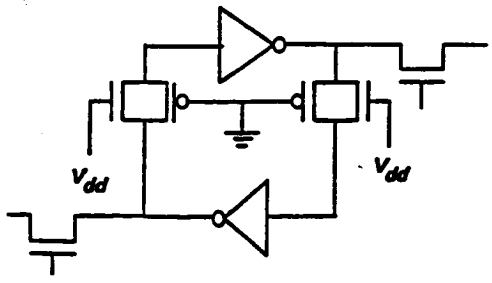


Figure 12c. A CMOS SRAM Cell with TGs as feedback resistors.

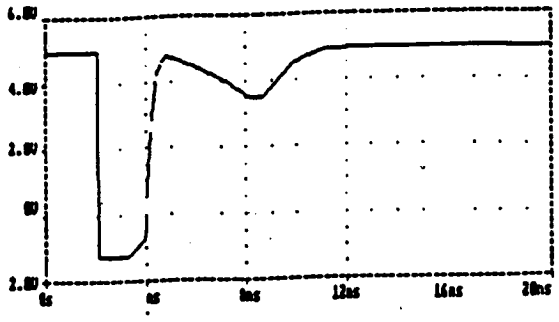


Figure 13a. Simulation Result of SRAM with TGs as feedback resistors.

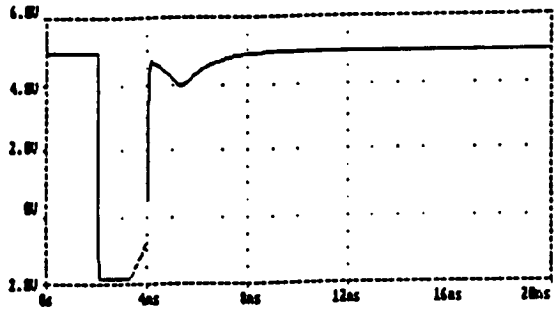


Figure 13b. Simulation result of SRAM with passive feedback resistors.

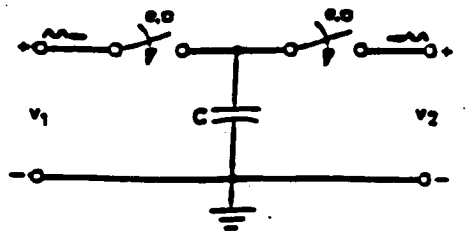


Figure 14a. Single phase grounded switched capacitor.

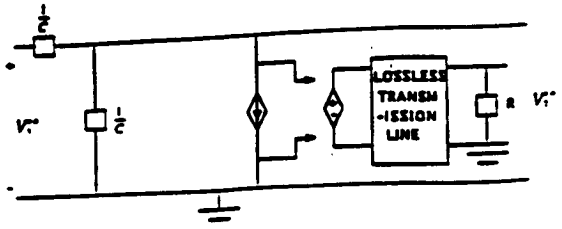


Figure 14b. Continuous-time domain equivalent circuit with implementation of lossless transmission line.

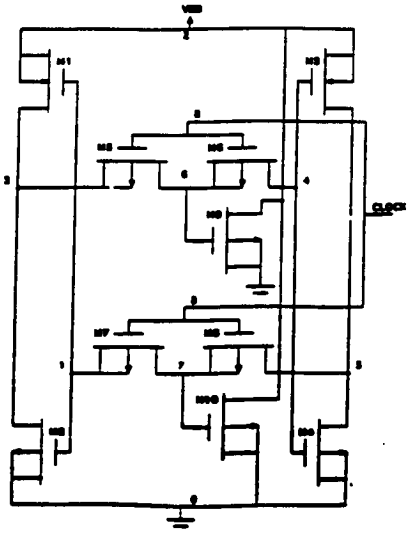


Figure 15. Switched Capacitor SRAM.

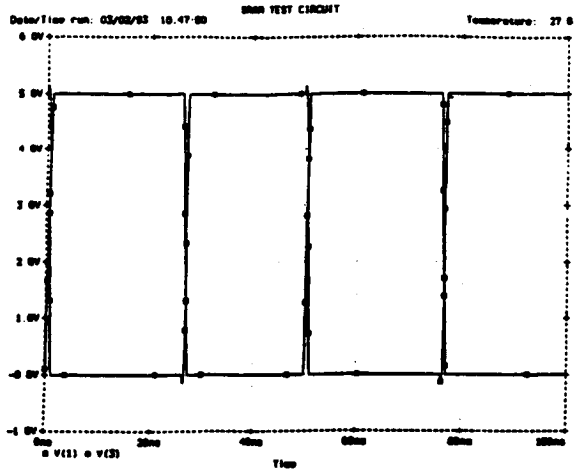


Figure 16. Transient analysis of CMOS SRAM without feedback resistor. V(1) is input and V(3) is output.

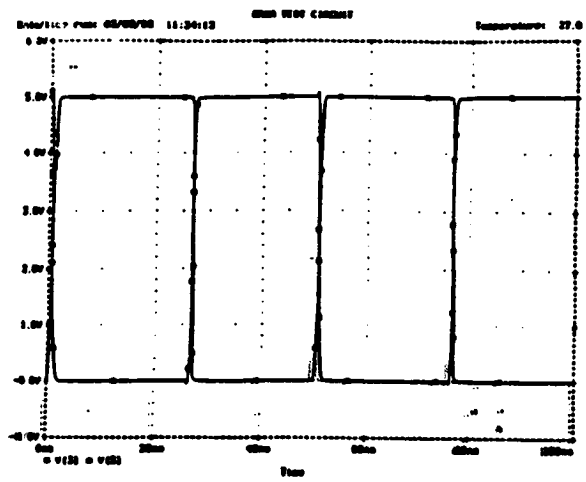


Figure 17. Transient analysis of switched capacitor SRAM. V(3) is input and V(5) is output.

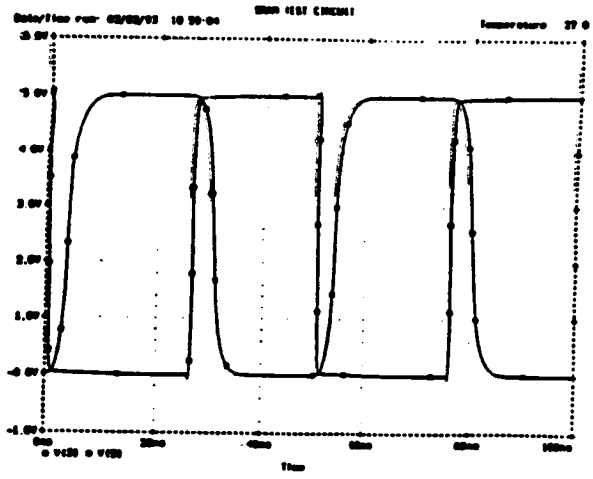


Figure 18. Transient analysis of CMOS SRAM with feedback resistor of 80 Kohms. V(3) is input and V(5) output.

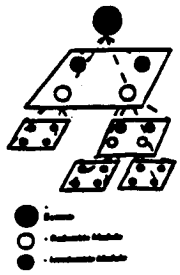


Figure 19. Hierarchical View of a system.

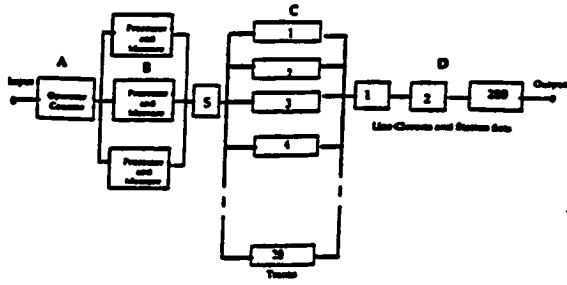


Figure 20. A private branch exchange system (PBX).

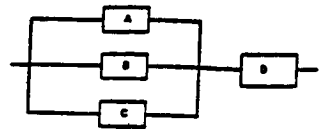


Figure 21 TMR System.

Simulation Modeling for Long Duration Spacecraft Control Systems

Mark A. Boyd
NASA Ames Research Center
Moffett Field, California

and

Salvatore J. Bavuso
NASA Langley Research Center
Hampton, Virginia

Simulation Modeling for Long Duration Spacecraft Control Systems

Mark A. Boyd □ NASA Ames Research Center □ Moffett Field
Salvatore J. Bavuso □ NASA Langley Research Center □ Hampton

Key Words: Fault Tolerant Hypercube Multiprocessors, Fault Trees, Weibull Decreasing Failure Rates, Cold Spares, HARP (Hybrid Automated Reliability Predictor), Simulation for Reliability Model Evaluation

Summary and Conclusions

We describe the use of simulation and contrast it to analytical solution techniques for evaluation of analytical reliability models. We also discuss the role importance sampling plays in simulation of models of this type. We next describe the simulator tool we use for our analysis. Finally, we demonstrate the use of the simulator tool by applying it to evaluate the reliability of a fault tolerant hypercube multiprocessor intended for spacecraft designed for long duration missions. We use the reliability analysis to highlight the advantages and disadvantages offered by simulation over analytical solution of Markovian and non-Markovian reliability models.

1. INTRODUCTION

Recent work in the development of reliability analysis tools has produced a number of software packages that allow complex system behavior to be expressed with analytical models. The systems to which these modeling methods are applied often are complex fault tolerant computing systems designed for very high reliability. However, these systems can exhibit certain types of system behavior that require analytical models for which feasible analytical (numerical) solution techniques are not currently available. In these situations the existing analytical modeling framework may be enhanced to allow simulation of the analytical model (i.e. a fault tree or Markov model) as a replacement solution method to the traditional analytical solution techniques for the model. This is the approach that we follow in this paper.

The very large number of trials needed to obtain statistically significant results historically has been a significant problem for the use of simulation to model complex, highly reliable fault tolerant systems. Recent efforts to overcome this problem have produced new modeling tools capable of obtaining acceptable results with a reasonable number of trials through the use of a variance reduction technique called *importance sampling*. New modeling tools which incorporate this technique have been designed to be compatible with the Hybrid Automated Reliability Predictor (HARP) modeling tool[9], which is itself a component of the HiRel package of reliability modeling tools[1]. HARP solves the same types of models as the simulator, but uses analytical (numerical) solution techniques instead of simulation.

As is often the case, the development of the new modeling tool we describe here was driven by the needs of a specific reliability analysis project: the use of hypercube multiprocessors for highly reliable guidance, navigation, and control (G,N,& C) systems for long duration manned spacecraft. We are interested in exploring the use of a fault tolerant hypercube architecture that can use either hot or cold spares. It is clear from preliminary studies that the use of hot and cold spares with the traditional constant failure rate model will not meet the high reliability requirement for long duration space missions without onboard repair[11, 12, 19]. Recently acquired empirical data provide convincing evidence that *decreasing* failure rates are common in spacecraft applications[10]. For these reasons, we want to be able to include decreasing failure rates in our reliability analysis. The inclusion of decreasing failure rates with cold spares requires the use of a non-Markovian reliability model which is substantially more difficult to solve analytically than a Markovian model that assumes constant failure rates. Given the current state of the art, analytical solution of such non-Markovian models generally is tractable only for very small simple models, whereas the model of the above hypercube system is very large. The cumulative effect of all of these factors led us to the use of simulation modeling.

In this paper we summarize the use of simulation as a modeling method and describe how it can be applied to the evaluation of analytical system models. We compare evaluation of analytical models by simulation to evaluation by analytical solution techniques and describe the role of importance sampling in our implementation of simulation. We next describe the simulator itself and the process of specifying a model for use with it. We then illustrate the use of the simulator by applying it to a hy-

percube architecture proposed for a G,N,& C system for long duration spacecraft. We explore the effect of assuming decreasing failure rates for active and cold processors within the hypercube instead of constant failure rates, and demonstrate the advantages that simulation provides over analytical solution methods for such system models.

2. SIMULATION MODELING FOR RELIABILITY PREDICTION

The usual method of using simulation to evaluate reliability and performance of systems involves building a computer model of the system, generating events of interest (i.e. component failures), and observing the response of the model to the generated events. The timing and types of events are generated using probability distributions which are assumed to govern event occurrence. Values are sampled from the appropriate probability distributions and are used to specify which type of event occurs next and when that occurrence will be. A sequence of events is generated in this manner until either the mission time expires or the system fails. Such a sequence of events provides one instance of how the system would be expected to behave in the environment characterized by the governing probability distributions and is referred to as a "history" or "trial". The model is evaluated at the end of a trial to determine measures of interest such as whether the system is still operating (reliability) or how much work was accomplished (performance), etc. This process is then repeated numerous times to obtain average values for the measures of interest and accompanying sample standard deviations. From probability theory it follows that as the number of trials increases, the average value obtained in the simulation approaches more closely the actual value that characterizes the long run behavior of the system as expressed by the model. The standard deviation, which is a measure of the expected closeness of the simulation average to the actual value, is proportional to $\frac{1}{\sqrt{n}}$ (where n is the number of trials)[18]. Hence obtaining a highly accurate value for a measure of interest may require a very large number of trials.

2.1 Analytic Solution Methods vs. Simulation

An alternate approach to reliability evaluation involves building an analytic (mathematical) model to express the relevant behavior of the system. A number of different analytical model types are in widespread use. One very successful analytical model type is the Markov chain and its generalizations (non-Markovian discrete state models). These models express system behavior by identifying a number of distinct states in which the system may be. The system can be in only one state at a time, and from time to time makes a transition from one state to another. The distribution of the time the system spends in individual states and the characteristics of the transition rates between states differentiate Markovian and non-Markovian models[13]. Analytical models are usually solved using either direct or numerical methods, so often they can give answers with greater accuracy than simulation methods for a comparable amount of computational effort. However, analytical solution methods suffer from requiring much more memory storage for data structures than simulation methods. As a result, models that become too large to be accommodated by analytical solution methods might still be within reach of simulation techniques. In addition, increasing behavioral complexity in analytical models requires analytical solution techniques with increasing computational requirements. Hence to solve a model of sufficient complexity, an analytical solution method could require more (rather than less) execution time than a simulation method for a comparable level of accuracy in the output. In cases like these where the limitations of analytical solution methods are exceeded, simulation provides a useful alternate approach.

The drawback to building a computer simulation model of a system under study is that constructing the model and validating it is often a complex, time consuming, and error-prone process. An alternative is to apply simulation not to a model of the system itself, but to an analytical model of the system such as a Markovian or non-Markovian model. With this approach there is of course the problem of constructing the analytical model. However, this tends to be easier than constructing a system-level simulation model. Also, the topic of analytical model construction has been addressed by a number of researchers in the past several years and tools have been created to assist in model construction (see [3] and [4] for brief surveys of tools for automated Markov model construction). The approach we have chosen for the current study applies simulation to Markovian and non-Markovian models of the hypercube multiprocessor system. This allows us to capitalize on previous work performed by the authors on the hypercube system using Markovian models[5] and permits us to extend the scope of that work.

2.2 Simulation for Evaluation of Markovian and non-Markovian Models

Markovian and non-Markovian discrete-state models can be evaluated by simulation in the following way. Each trial represents a single traversal path among the states of the model. The common beginning point for all trials is at an initial state in which all system components are assumed to be operating correctly. Upon entry into each state, the process is begun for determining the time of transition out of the current state and which state the system goes to next. The time to next transition is sampled from a probability distribution that depends upon

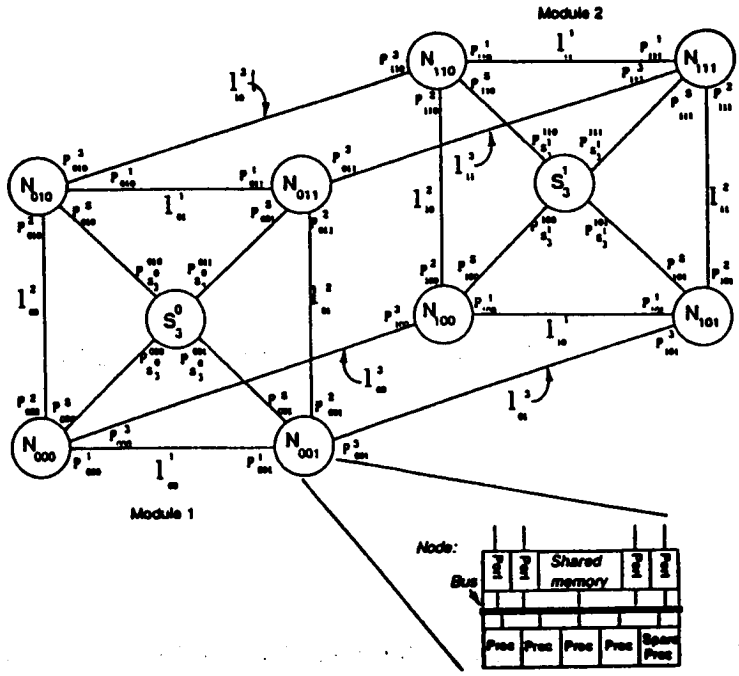


Figure 1: Hypercube Multiprocessor System

the failure rates of the components still active. If the failure rates of all components are constant, the model is a Markovian model. If the component failure rates are all functions of mission time (i.e. non-constant), the model is a non-homogeneous Markov model. If the component failure rates are individually functions of more than one time variable (i.e. there is more than one "clock" in the system upon which component failure rates may depend), the model is a non-Markovian model. We use all three types of models in the present study. Once the time to next transition has been determined, a sampling from a second distribution is done to determine which of the remaining operating components will experience the failure that is the cause of the transition out of the state. The determination of the sampling distributions is described in [14] and [16]. We note that this formulation of the simulation process can accommodate the use of Fault/Error Handling Models (FEHMs) to implement *behavioral decomposition* for incorporating imperfect fault coverage as is done in HARP[15]. Although that capability was available, we did not consider imperfect fault coverage in the present study. During each trial successive inter-state transitions are generated until either the mission time is exceeded or the system fails, causing the trial to end. The system unreliability is then estimated from the proportion of trials during which the system failed before the mission time was reached.

2.3 Importance Sampling

A major characteristic of highly reliable systems is that system failure events are extremely rare. This means that a large majority of the trials are likely to end by the mission time expiring rather than through a system failure. Since system failures are the events of interest, a very large total number of trials must be run before a sufficient number of system failures occur to provide a meaningful estimate from the proportion of failure trials to total trials (i.e. an estimate of the system unreliability). A variance reduction technique called *importance sampling* may be employed to reduce the total number of trials required. An excellent introduction to importance sampling may be found in [6]. The basic idea behind importance sampling is to select an alternate distribution from which to sample which has much higher probability density than the original distribution in the regions of interest where the original distribution's density is very small. Parity to sampling from the original distribution is maintained by weighting the observations sampled from the new distribution to reflect the relative difference in density magnitude between the two distributions. For example, if the density of the new distribution is four times greater than the density of the original distribution in a certain region, then a failure event observed in that region by sampling from the new distribution is counted as only $\frac{1}{4}$ of a failure. The importance sampling techniques implemented in the simulator we used for this study, called *forced transitions* and *failure biasing*, are described in [14]. Both have the effect of emphasizing component failure events in order to increase the number of trial terminations due to system

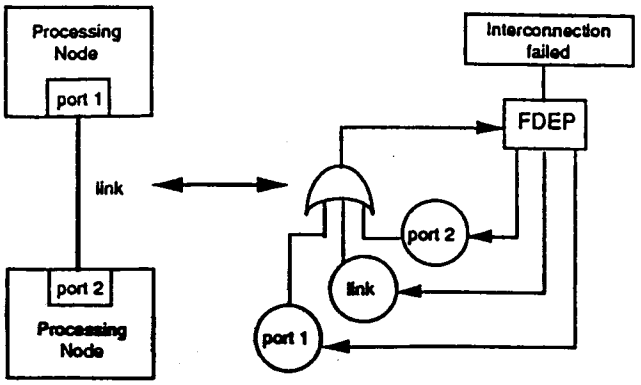


Figure 2: Modeling Functional Dependencies due to Processing Node Interconnections

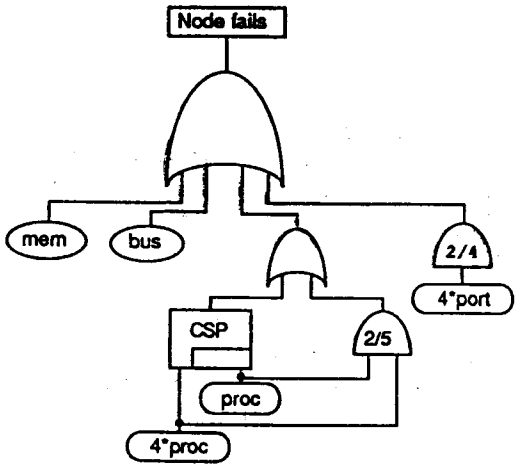


Figure 3: Fault tree model of Architecture 1 Processing Node with Cold Spares

failure, hence reducing the total number of trials needed in order to accumulate a sufficient number of system failure terminations to provide an acceptable estimate of the system unreliability.

2.4 Simulator Description

The original version of the simulator we used for our analysis was designed by Lewis[17] and implemented at Northwestern University. It required a system model to be described as a set of components arranged in groups. Each group could optionally have cold spares, and could have either a constant or a Weibull increasing failure rate. Each group could also have a Fault/Error Handling Model (FEHM) associated with it to allow the use of behavioral decomposition as is done in HARP. System failure criteria were specified in the form of a set of component cut sets which the analyst had to derive from a combinatorial model of the system (for example, a fault tree). For our study, we modified this simulator to enable it to use decreasing as well as increasing Weibull failure rates, and to allow it to accept the input model in the form of a *dynamic fault tree* (see below) rather than as a set of component cut sets. The resulting simulator program accepts its input model in the same form as the HARP program, and accepts input files with the same format as HARP. In addition, it is capable of evaluating all models that HARP is capable of evaluating, making it completely compatible with HARP. This is an important advantage because it allows the reliability analyst to develop his/her system model once and then input it to whatever evaluation program is most appropriate depending on the characteristics of the model and the programs. It also allows a comparative evaluation of the performance of the two programs by applying them both to the same model(s).

3. SYSTEM MODEL

The hypercube multiprocessor system and the model of it that we use in this study are described in [3] and [5] under the name of Architecture 1. We give a brief description of it here. The architecture is shown in figure 1. It consists of a 3-dimensional hypercube configured as two fault-tolerant 2-dimensional modules, each with a spare processing node. The processing nodes themselves are multiprocessors containing four active processors and a spare processor. The spare processor can be either a hot or cold spare. The structure of the processing nodes is also shown in figure 1. Each processing node communicates with other processing nodes in the system through four ports. For the system to be operational all eight processing nodes must be operational and must all be able to communicate with each other. Therefore, the system will be considered failed if any processing node fails and a spare processing node is unable to take over or if any two nodes in the hypercube are unable to communicate with each other.

Although the form of the analytical model that is actually evaluated is a Markovian/non-Markovian discrete-state model, it is specified by the reliability analyst in the form of a *dynamic fault tree*[3, 8]. When simulation is not used for model evaluation, the dynamic fault tree can be converted into a Markov chain which can then be solved numerically for state probabilities. When simulation is used for model evaluation, the discrete-state structure of the underlying Markovian model is inherent in the simulation process and the dynamic fault tree is used only to determine whether a state which has been entered is a failure state.

A dynamic fault tree is a generalized fault tree model in which the traditional set of combinatorial fault tree gates is extended to include several non-standard gates that are designed to express *sequence dependent* behavior. Sequence dependent behavior is behavior that depends in

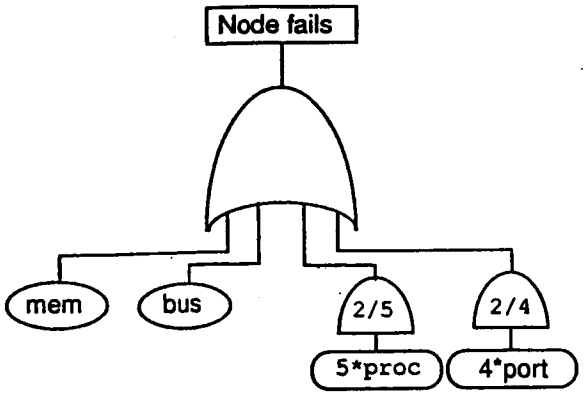


Figure 4: Fault tree model of Architecture 1 Processing Node with Hot Spares

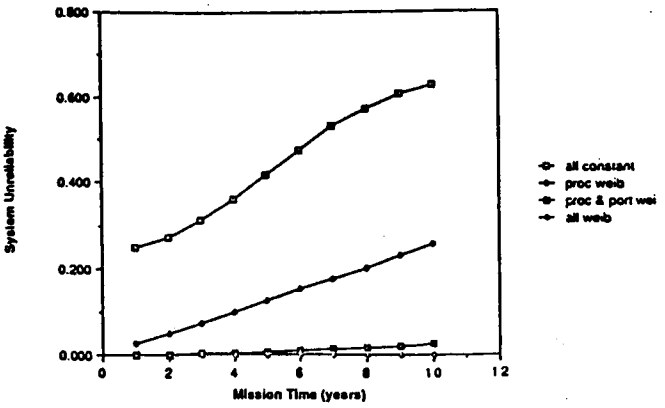


Figure 5: Effect of Weibull DFRs on System Unreliability (Hot Spares)

Component	Initial constant failure rate
Shared Memory	3.477×10^{-7}
Intra-node bus	1.147×10^{-7}
Processor	1.990×10^{-6}

Table 1: Initial Constant Hazard Rates (failures/hour) for Components in Processing Nodes

some way on the order in which events occur. The hypercube system under study exhibits two instances of this type of behavior: functional dependencies (the failure of one component causes one or more other components to either fail or become unavailable) and cold spares (a cold spare cannot fail while it is "cold"; it can fail only after it has been activated to substitute for a failed active component). The functional dependencies appear in the interconnections between the processing nodes; specifically, if either the internode link or one of the two ports on either side of the internode link fails, the remaining two components (link and/or port(s)) become useless to the remaining operation of the system and hence may be considered to be effectively failed themselves. These functional dependencies are modeled with *functional dependency gates*, as shown in figure 2. Cold spares are used within the processing nodes and are modeled using a *cold spare gate*, an example of which appears in figure 3.

Figure 4 models a processing node when the spare processor is hot (i.e. active and running from mission start just like the four initially active processors). The 2-out-of-4 gate for which the four ports are inputs reflects the effect of the message routing protocol[5]. Figure 3 models the processing node when the spare processor is cold. Diagrams of fault trees modeling the full architecture were omitted from this paper due to lack of space. The interested reader may find them in [3].

4. ANALYSIS RESULTS

We evaluated the system model for the cases where all components had constant failure rates with hot or cold spare processors (time homogeneous Markov models), various components had Weibull DFRs with hot spare processors (non-homogeneous Markov model), and various components had Weibull DFRs with cold spare processors (non-Markovian model). For this paper our primary purpose is to illustrate the use of simulation to evaluate the models and contrast it with analytical solution techniques. Therefore we will use here only selected results from our analysis to compare the advantages and disadvantages of simulation vs. analytical solution methods. A more complete reliability analysis of the hypercube system is found in [2]. Our primary analysis goal was to determine whether assuming Weibull decreasing failure rates (DFRs) for components instead of constant failure rates would result in a sufficient improvement in predicted system reliability to conclude that the architecture was adequate to successfully complete a 10 year mission. Results using constant failure rates[2] indicated that the proposed architecture would be inadequate, with the probability of system failure exceeding 60% after 10 years. Initial attempts to evaluate the model with HARP (which uses analytical solution techniques) were not successful due to the large size of the model. The dynamic fault tree model of the system contains 70 basic events (110 components total), and 175 fault tree nodes (basic events + gates). It produces a Markov model with many thousands of states. Furthermore, when Weibull DFRs are assumed together with cold spares, the size and complexity of the resulting non-Markovian model is well beyond the capability of any analytical solver tool that exists today, both in terms of memory and execution time required for its solution. In contrast, our simulator was able to evaluate the model with none of the problems experienced by HARP. Components with decreasing failure rates were assumed to have an initial failure rate λ_{exp} given in table 1 which declines monotonically over the mission time according to the Weibull failure rate expres-

Mission Time (Years)	All Components Constant FRs	Processors Weibull DFRs	Processors and Ports Weibull DFRs	All Components Weibull DFRs
1	$.249 \pm .016$	$.0250 \pm .0031$	$.000519 \pm .00022$	$.000255 \pm .00013$
2	$.271 \pm .016$	$.0489 \pm .0048$	$.00147 \pm .00031$	$.000361 \pm .00015$
3	$.312 \pm .017$	$.0738 \pm .0065$	$.00286 \pm .00044$	$.000439 \pm .00017$
4	$.361 \pm .018$	$.0988 \pm .0091$	$.00481 \pm .00078$	$.000504 \pm .00019$
5	$.419 \pm .018$	$.126 \pm .014$	$.00729 \pm .0013$	$.000550 \pm .00020$
6	$.475 \pm .018$	$.152 \pm .017$	$.0102 \pm .0018$	$.000638 \pm .00031$
7	$.530 \pm .018$	$.176 \pm .019$	$.0135 \pm .0024$	$.000673 \pm .00033$
8	$.576 \pm .017$	$.202 \pm .023$	$.0173 \pm .0037$	$.000718 \pm .00036$
9	$.609 \pm .016$	$.231 \pm .031$	$.0208 \pm .0045$	$.000766 \pm .00041$
10	$.631 \pm .013$	$.257 \pm .036$	$.0257 \pm .0091$	$.000777 \pm .00041$

Table 2: Effect of Weibull DFRs on System Unreliability (Hot Spares)

sion:

$$\lambda_{weib}(t) = \lambda_{exp}\alpha t^{\alpha-1} \quad (1)$$

where α is the Weibull shape parameter[20] which is assumed to have the value $\alpha = 0.5$. All components not having DFRs were assumed to have constant failure rates given in table 1. Table 2 and figure 5 show the effect of assuming Weibull DFRs for various subsets of components. The results reported in table 2 are averaged over 10 runs of 10000 trials per run. The effect of assuming Weibull DFRs for increasing numbers of the components clearly results in decreasing system unreliability. The result of assuming Weibull DFRs for all components is a difference of about three orders of magnitude in the system unreliability (from 0.631 ± 0.013 when all components have constant FRs down to about $0.777 \times 10^{-3} \pm 0.41 \times 10^{-3}$ when all components have Weibull DFRs).

The above discussion illustrates the advantage that simulation can have over analytical techniques: simulation may be able to evaluate models that are beyond the reach of analytical techniques both in terms of memory and execution time. Furthermore, if only ballpark evaluations are desired, simulation may be able to produce the required results relatively quickly. Figure 6 contrasts the reliability predictions for the hypercube with hot spares assuming constant failure rates and Weibull DFRs for all components. The results are averaged over 10 runs, with each run consisting of only 1000 trials requiring approximately 4 minutes or less of clock time. With only 1000 trials per run, the standard deviations are relatively large. Nevertheless, the outcome of the comparison is clearly apparent.

However, simulation does have an important disadvantage compared to analytical solution techniques. If the accuracy of the evaluation is important, then the execution time required by simulation to achieve the required accuracy increases rapidly and can quickly become uncompetitive with that required by analytical solution techniques (provided the model is small enough for analytical solution techniques to be used). Table 3, which shows the reliability of a single processing node in the hypercube and the execution time required to obtain it, contrasts the values obtained using HARP to values obtained using the simulator with varying numbers of trials per run. Increases in the accuracy of the reliability estimate, as measured by the decreasing size of the standard deviation, require very significant increases in the execution time. The table clearly shows that it is better to use the analytical solver than the simulator, both in terms of execution time and accuracy of the reliability prediction. This result holds in general, and experience has shown that it is usually preferable to use an analytical solver whenever feasible rather than a simulator to evaluate a reliability model. In particular, whenever accuracy in results is important we feel that the use of a simulator generally should be a last resort to be pursued after analytical modeling techniques have been found to be infeasible.

5. SUMMARY

We have described a reliability analysis study which was performed to determine whether assuming of Weibull decreasing failure rates (DFRs) for components of a fault tolerant hypercube would significantly improve the 10 year system reliability estimate over that obtained assuming constant failure rates. Our results show that a substantial improvement in system reliability does result from assuming Weibull DFRs, indicating that a candidate architecture that would otherwise be considered inadequate instead could provide acceptable reliability after all. We also contrasted the use of simulation and analytical solution techniques to evaluate Markovian and non-Markovian reliability models. Observations made from our analysis indicate that analytical solution techniques are preferable whenever the model is small enough and when accuracy of the answer is important. Conversely,

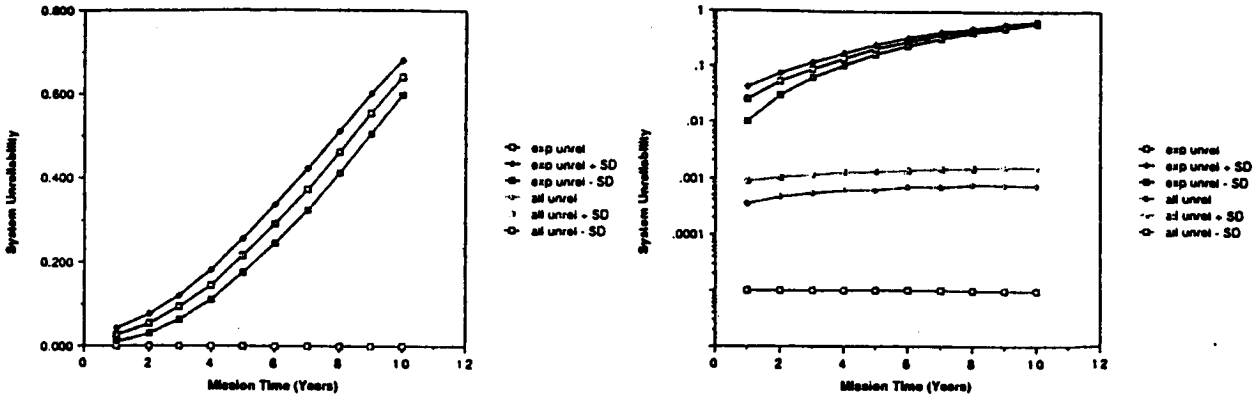


Figure 6: Ballpark Evaluation of the Effect of Weibull DFRs on System Unreliability (Hot Spares)

Solver	Reliability Estimate	CPU time required
HARP	0.04468	6.4 sec
Simulator, 10^3 trials/run	$.04374 \pm .0023$	29.2 sec
Simulator, 10^4 trials/run	$.04455 \pm .00073$	4 min 53.8 sec
Simulator, 10^5 trials/run	$.04462 \pm .00023$	48 min 26.9 sec
Simulator, 10^6 trials/run	$.04463 \pm .000073$	8 hrs 0 min 2.5 sec

Table 3: Processing Node Model Evaluation Accuracy vs. Execution Time

simulation is preferred whenever approximate ballpark answers for a large model are sufficient, or when the model is too large or exhibits system behavior too complex to be accommodated by analytical solution techniques. Finally, we have described a simulator tool for evaluating Markov and non-Markovian reliability models which is compatible with the HARP (analytical) reliability evaluation program and is part of the HiRel package of reliability evaluation tools. There is a great advantage to having analytical and simulation tools be compatible with each other in this way (i.e., both using the same input models and files, and both providing the same analysis capability) because it allows the reliability analyst a great deal of flexibility in conducting the analysis. Solution methods may be mixed and matched and applied in the most appropriate way to a single system model depending on the type and scope of the desired results.

References

[1] Salvatore J. Bavuso and Joanne Bechta Dugan. HiRel: Reliability/availabilty integrated workstation tool. In *Proceedings of the Reliability and Maintainability Symposium*, pages 491–500, January 21-23 1992.

[2] M. A. Boyd and Salvatore J. Bavuso. Modeling a highly reliable fault-tolerant guidance, navigation, and control system for long duration manned spacecraft. In *AIAA/IEEE Digital Avionics Systems Conference, Seattle, WA*, October 1992.

[3] Mark A. Boyd. *Dynamic Fault Tree Models: Techniques for Analysis of Advanced Fault Tolerant Computer Systems.* PhD thesis, Department of Computer Science, Duke University, 1990.

[4] Mark A. Boyd. *What Markov Modeling Can Do For You: An Introduction.* 1993 Reliability and Maintainability Symposium, Tutorial Notes, January 1993.

[5] Mark A. Boyd and Jesus O. Tuazon. Fault tree models for fault tolerant hypercube multiprocessors. In *Proceedings of the Reliability and Maintainability Symposium*, January 1991.

[6] Charles E. Clark. Importance sampling in Monte Carlo analyses. *Operations Research*, 9:603-620, September-October 1961.

[7] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Transactions on Computers*, pages 547–553, May 1987.

[8] Joanne Bechta Dugan, Salvatore Bavuso, and Mark Boyd. Fault trees and sequence dependencies. In *Proceedings of the Reliability and Maintainability Symposium*, pages 286–293, January, 1990.

[9] Joanne Bechta Dugan, K. S. Trivedi, Mark K. Smotherman, and Robert M. Geist. The hybrid automated reliability predictor. *AIAA Journal of Guidance, Control and Dynamics*, 9(3):319–331, May-June 1986.

[10] Herbert Hecht and Eugene Florentino. Reliability assessment of spacecraft electronics. In *Proceedings of the Reliability and Maintainability Symposium*, pages 341–346, January 1987.

[11] M. L. Johnson. Long duration mission reliability via multiprocessing. In *American Astronautical Society, 17th Annual Meeting the Outer Solar System*, June 1971.

[12] Jong Kim and et al. Chita R. Das. Reliability evaluation of hypercube multicomputers. *IEEE Transactions on Reliability, Special Issue on Parallel/Distributed Computing Networks*, 1988.

[13] H. S. Larson and B. O. Shubert. *Probabilistic Models in Engineering Science*, volume II. John Wiley & Sons, NY, 1979.

[14] E. E. Lewis and F. Boehm. Monte Carlo simulation of Markov unreliability models. *Nuclear Engineering and Design*, 77:49–62, 1984.

[15] E. E. Lewis, F. Boehm, C. Kirsch, and B. Kelkhoff. Monte Carlo simulation of copmplex system mission reliability. In *Proceedings of the 1989 Winter Simulation Conference*, pages 497–504, Washington, D.C., December 4-6 1989.

[16] E. E. Lewis and T. Zhuguo. Monte Carlo reliability modeling by inhomogeneous Markov processes. *Reliability Engineering*, 16:277–296, 1986.

[17] M. E. Platt, E. E. Lewis, and F. Boehm. General Monte Carlo reliability simulation cold including common mode failures and HARP Fault/Error-Handling. Technical report, The Technological Institute of Northwestern University, January 1991.

[18] R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, NY, 1981.

[19] Lei Tien and et al. Chita R. Das. Reliability evaluation of butterfly multiprocessors. In *ACM SIGMETRICS*, 1989.

[20] K. S. Trivedi. *Probability and Statistics with Reliability, Queueing and Computer Science Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

Biographies

Mark A. Boyd
NASA Ames Research Center
Mail Stop 269-4
Moffett Field, CA 94035 USA
(415) 604-3678
FAX: (415) 604-4036

Mark A. Boyd is a research scientist in the Information Sciences Division at NASA Ames Research Center. He was awarded a BA in Chemistry from Duke University in 1979, an MA in Computer Science from Duke University in 1986, and a Ph.D. in Computer Science from Duke University in 1991. His research interests include mathematical modeling of fault tolerant computing systems and the development of dependability modeling tools. He is a member of the IEEE, the ACM, and the MAA.

Salvatore J. Bavuso
NASA Langley Research Center
Mail Stop 478
Hampton, VA 23681 USA
(804) 864-6189
FAX: (804) 864-7891

Salvatore J. Bavuso is a senior researcher at NASA Langley Research Center in Hampton, VA. He received a BS degree in mathematics from the Florida State University in 1964 and a MS degree in applied mathematics from the North Carolina State University at Raleigh in 1971. He has been instrumental in the development of advanced reliability modeling technology for over a decade and is the NASA project manager for the HARP and CAREIII programs.

Technology Drivers for Flight Telerobotic System Software

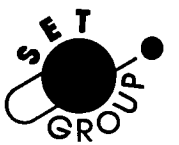
Robert Labaugh
SET Group
Denver, Colorado

Technology Drivers for Flight Telerobotic System Software

Robert J. LaBaugh
SET Group
Denver, Colorado

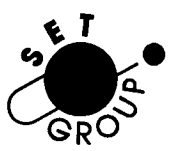
Selected Topics In
Robotics For Space Exploration

3/18/93 LaRC



Introduction

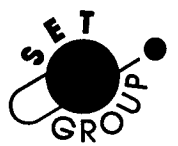
- Major Software Drivers in a Flight Manipulator System
  - Control Algorithms
  - Distributed Hardware Architecture
    - Bus Loading
    - Margin/Performance Requirements (10ms/20ms)
  - Data Management
    - Telemetry/Data Recording
    - Operator Interface
  - Safety
  - Fix It in Software



Flight Software Lines of Code

- Estimated at 40K Ada Statements
- Approximately 22K in Development Library at Start of Technology Capture Effort

Function	Lines of Code	Percentage
Control Algorithms	5.5K	13.8%
Operator Interface	10.0K	25.0%
Safety	5K	12.5%
Data Management	7K	17.5%
Misc. Hardware Control	5.5K	13.8%
Common Utilities	5K	12.5%
ROM	2K	5.0%



Flight Computer Architecture

- Distributed 80386-80387s
  - 8 Controllers
    - Custom Design
    - 512K Bytes RAM
    - Joint Controllers Embedded in Arm
    - 4x4 in. Surface Mount Boards
  - Space Station Standard Data Processor
    - 3 CPUs with 4M Bytes RAM each
- MIL-STD-1553B Busses Connecting CPUs
  - Workstation Bus
  - Telerobot Bus
- PGSC Used for Display and Initial Program Load

System Safety

- Critical Items Required To Be Two Fault Tolerant
  - One Path Outside of Computer System
  - Other Two in Independent Systems
- FTS Safety Requirements
  - Safe Return Of Orbiter
    - Doors Must Be Able To Close
    - System Must Be Safe For Landing (Caged)
  - Inadvertent Release of Hardware
    - Manipulator Grasp of Object
    - Object Caging Mechanism
  - Correct Operation of Manipulator
    - No Unplanned Contact with Environment
    - Planned Contact at Safe Forces and Torques

Safety Critical Parameters

Parameter	Monitored By	Hazard Mitigated
Cartesian Position	TRCC/TRRC	Unplanned Contact
Cartesian Velocity	TRCC/TRRC	Unplanned Contact
Cartesian Force (6 DOF)	Joint Controllers (H/W)	Excessive Force
Joint Position	Joint Controllers (H/W)/TRRC	Unplanned Contact
Joint Velocity	Joint Controllers/TRRC	Unplanned Contact
Joint Torque	Joint Controllers (H/W)	Unplanned Contact, Excessive Force
Joint Motor Current	Joint Controllers (H/W)	Unplanned Contact, Excessive Force
End Effector Gripping Force	Joint Controllers (H/W)	Excessive Force
End Effector Grip Current	Joint Controllers (H/W)	Excessive Force
Joint Position Variance	Joint Controllers	Unplanned Contact
FTT-A versus FTT-B Variance	TRCC	Excessive Force
Actuator/EE Temperature	TRCC	Failure to Stow
Processor Temperature	TRCC	Failure to Stow, Unplanned Contact
Processor Health	TRCC/TRRC/PM	Unplanned Contact

System Safety – Software Functions

- Cartesian Safety
  - Position/Boundary Management
    - Check Arm Position versus Environment
  - Velocity Limits
  - Force Applied to Environment
- Manipulator Joint Safety
  - Position versus Joint Stops
  - Consistency of Three Position Sensors
- Communications
  - Heartbeat Between Critical Computers
  - Checksum of All Messages
- Temperatures

System Safety – Software Functions (cont.)

- Operational Checks
  - Tighter Bounds than Safety Limits
  - Violation Results in Limited Value or Soft Stop
- Safety Checks
  - Violation Results in Emergency Shutdown
- Hardware Checks Can Also Produce Emergency Shutdown (ESD)
  - Need to Report Sensor Which Caused ESD
- Ada Run-time Checks Not Sufficient for Detection of Problems
  - Corruption of Code
  - Execution of Non-code
- DDC-I Use of 80386 Protected Mode
  - Code in Read Only Segment
  - Access Outside of Segment Trapped by Hardware

Fix It in Software

- Coarse Encoder Calibration Curves
  - Position Dependent Error
  - Varied with Temperature
  - Varied with Time
- Augmented Damping
  - 1000Hz
  - Multiple Digital Filters
- FTT Decoupling
- Safety
  - Force Limiting
  - Third Instance of Collision Avoidance
- Power Switch Control
- Power/Thermal Problem

Early Lunar Rover Mission Studies

V. P. Gillespie
NASA Langley Research Center
Hampton, Virginia

NASA

Early Lunar Rover Mission Studies

Vernon P. Gillespie
Langley Research Center

Presented to: Robotics for Space Exploration Workshop
March 18, 1993

NASA

Early Lunar Rover Mission Studies Chronology

- August 1991 - Project Conceived
- October 25, 1991 - Presentation of Concept to Dr. Griffin
- November 1991 - February 1992 Concept Refined
- February 12, 1992 - Revised Concept Presented to Dr. Griffin
- March 1992 - Study Team Formed
- April 1992 - Possible "Trafficability Problem" Identified
- April 1992 - Concept Presented to LPI Workshop
- June 1992 - Dynamic Model Designed and Fabrication Initiated
- September 1992 - Concept Presented to International Planetary Mobil Vehicles Conference
- October 1992 - Dynamic Model Tested
- November 1992 - JSC put Artemis on hold
- December 1992 - Study Completed

NASA

Study Project Description

- Launch 1 -60kg mini rover on the Artemis lander to the Moon
- Rover shall be capable of site certification near the landing site and long traverses to obtain compositional and selenophysical profiles of the surface
- **NASA will operate rover from selected universities to support educational purposes**
- The first launch shall be within 3 years of funding availability
- **Hardware design, analysis, fabrication, test, and operations to be accomplished in-house to train young government personnel**
- **Project to be managed by LaRC; supported by Wallops, Sandia and CERL/WES**

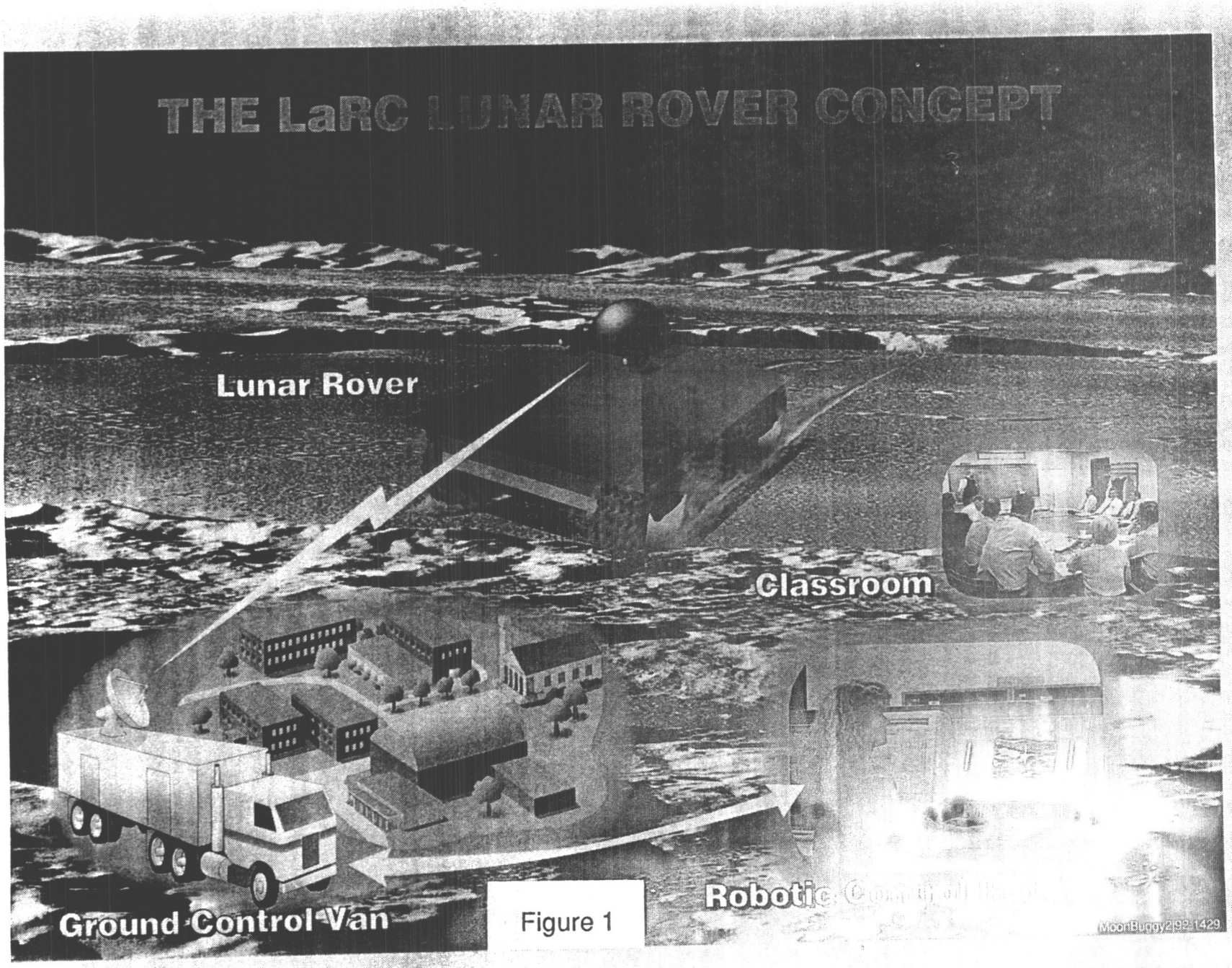
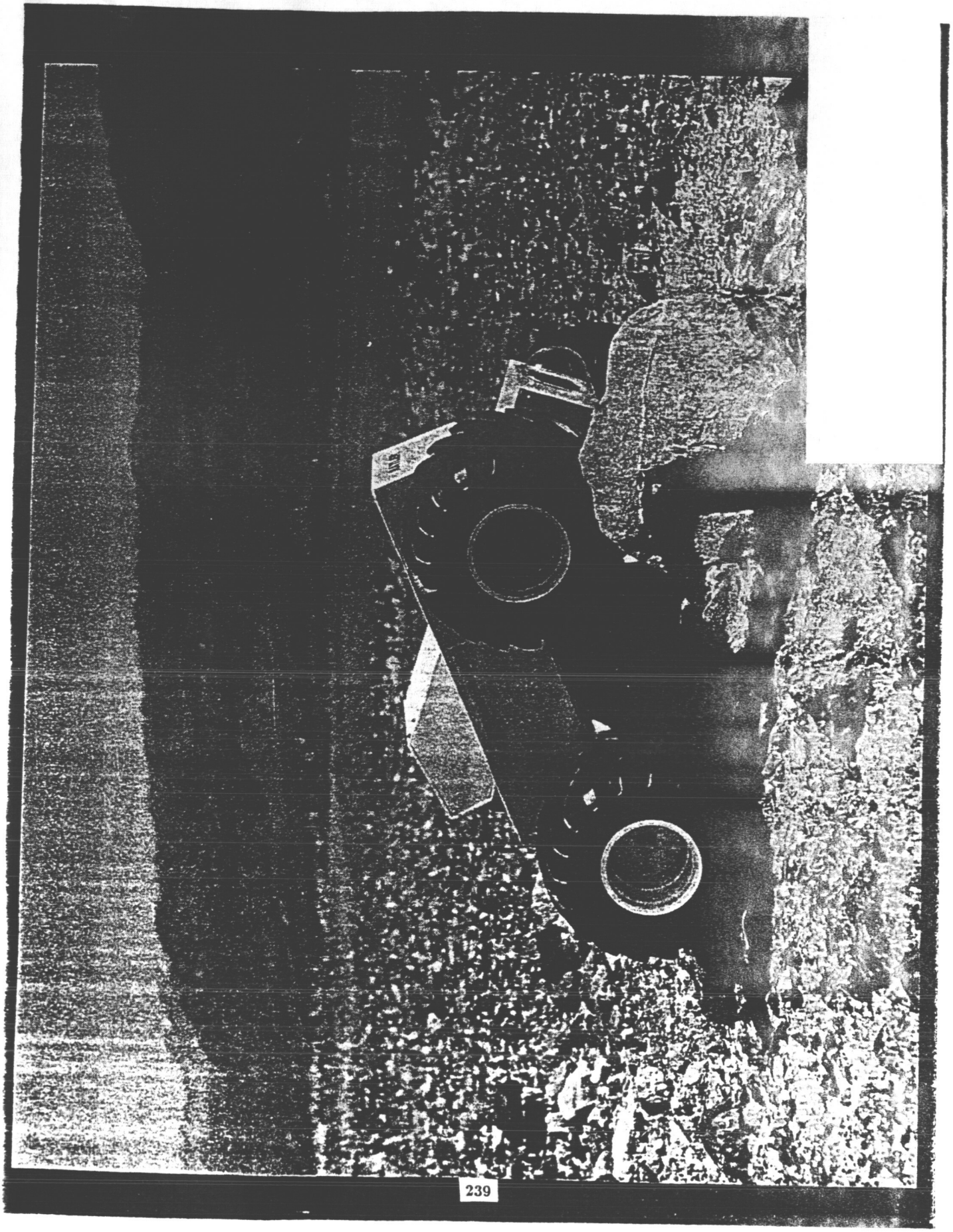
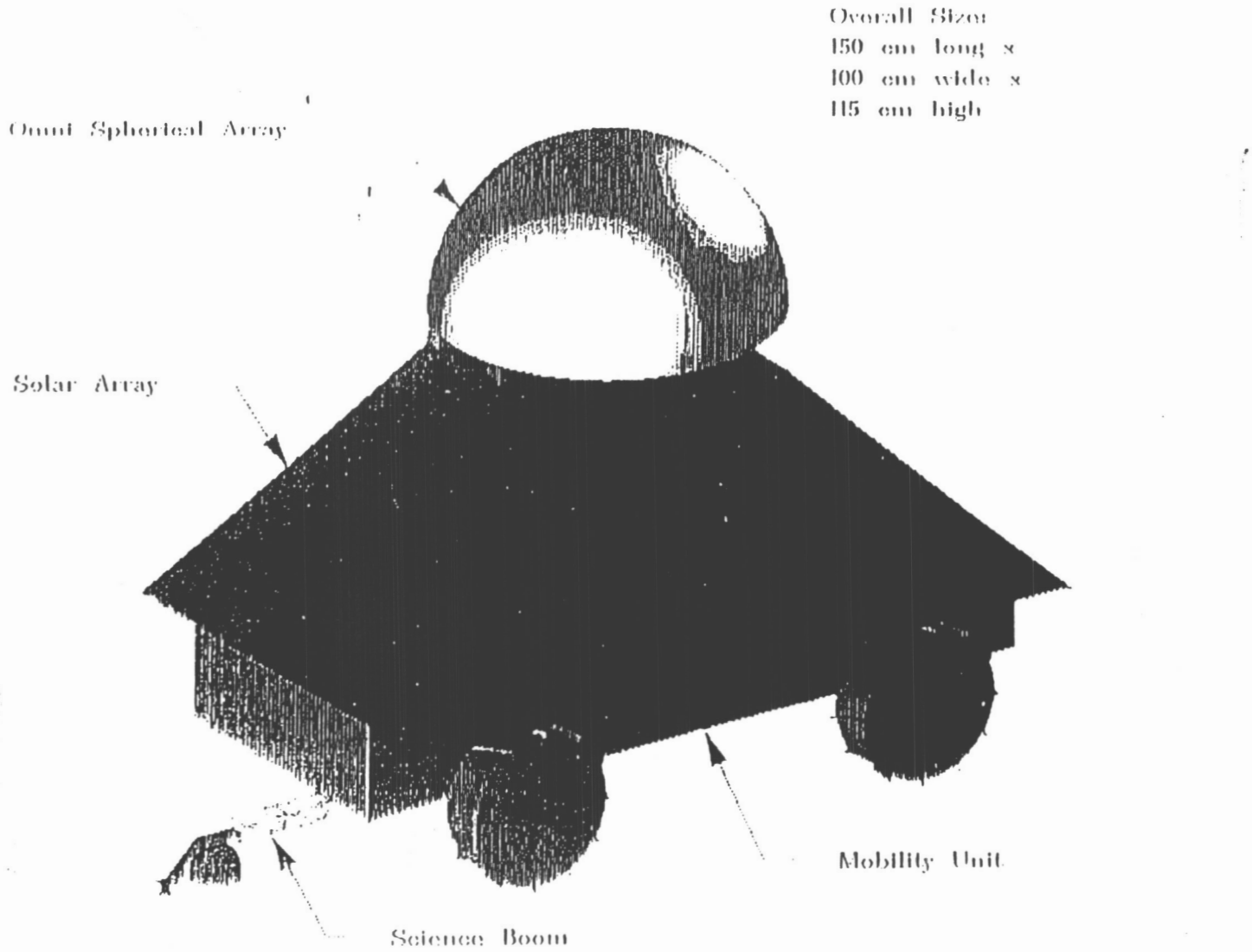

THE LaRC LUNAR ROVER CONCEPT
Lunar Rover
Classroom
Ground Control Van
Robotic
MoonBuggy2|92-1429

Figure 1

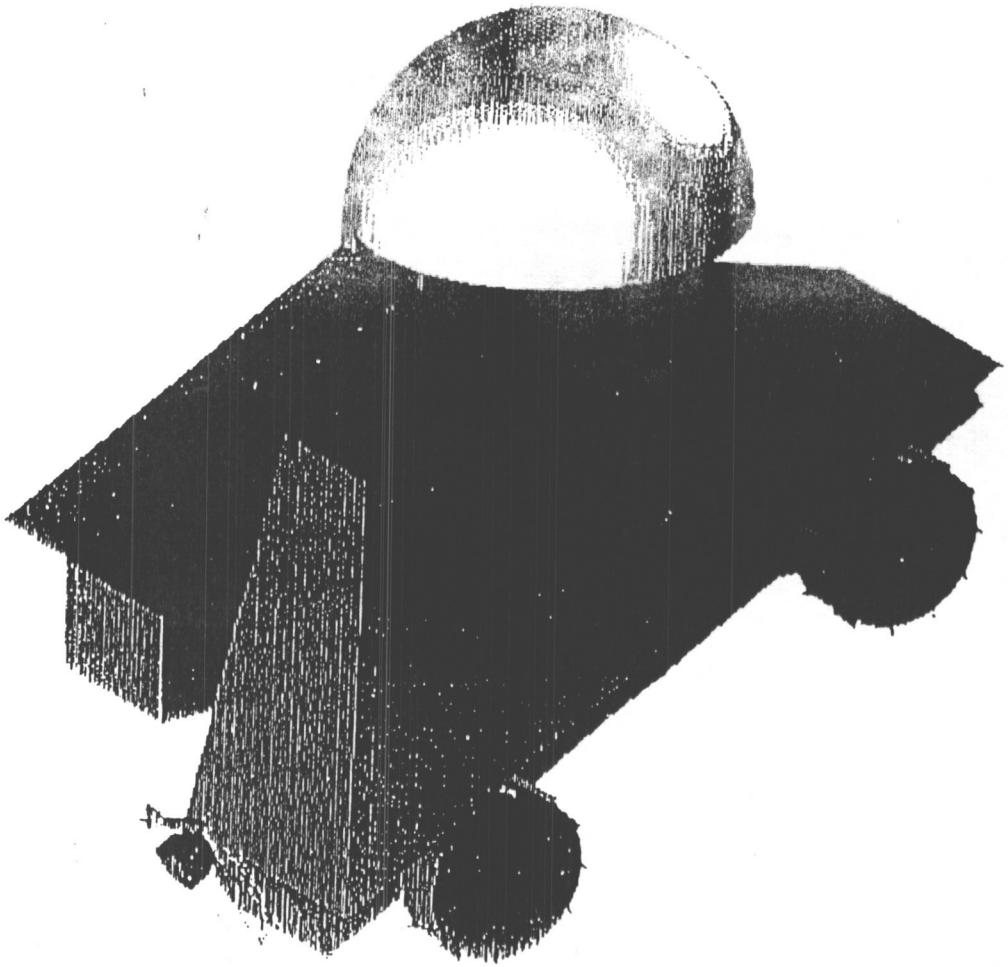




LaRC Lunar Rover

Left Side View

(8/18/92)



LaRC Lunar Rover

Left Side Rotated 30 degrees about Body Shaft

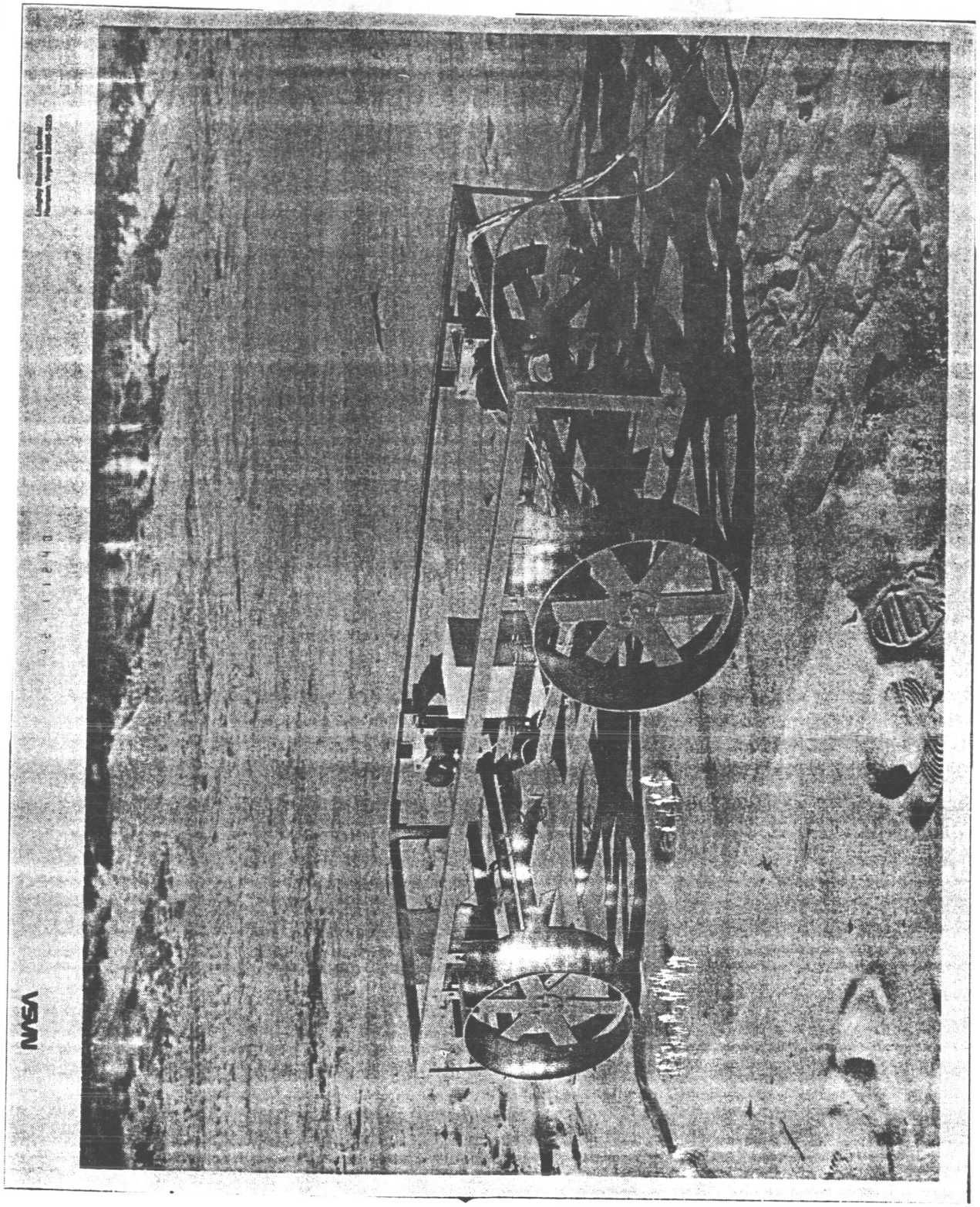
(8/18/92)

NASA

EXPLORATION

Issues Identified for Analysis and Basic Questions to be Addressed

- Mobility - Are the Bekker equations applicable to this class of vehicle?
- Power - Can solar arrays generate sufficient power for the vehicle to operate effectively?
- Imaging - Can the imaging requirements for science and mobility be met by a common camera?
- Telemanagement - What level of autonomy is required to operate the vehicle effectively?
- Communications - What data rate is required for effective operations and how can it be achieved within program constraints?


NASA

NASA

Telemanagement Results

- Teleoperation of Lunar Rovers is feasible, practical and within current technology
  - USSR used in early '70s
  - Sandia test results encouraging
- An autonomous robot is not required for lunar operations
- Some degree of telemanagement will be appropriate for lunar operations
  - Hazard detection
  - Position control

NASA

LaRC Early Lunar Mission Study Results

- A mission to accomplish generally accepted science objectives is practical within cost guidelines
- A total systems approach is required to maximize efficiency of subsystems
- No new technology is required; although in some areas would enhance success
  - Robotics
  - Instrumentation
  - GN&C
- Significant development required in some areas specifically batteries, science instruments
- Developed excellent working relationship with Corps of Engineers and Sandia National Labs
- Completed with no program resources

NASA

Concluding Remarks

- Experience and early test results show that early lunar rovers can be teleoperated
- Follow-on rovers and Mars rovers will require a high level of automation, although not all operations will be autonomous
- Significant development effort is needed for full autonomous operations
- Details of Lunar, Mars and lab hardware very significantly (attention to detail is an absolute must for mission success)
- LaRC has excellent working relationships with other NASA Centers, government agencies (including ESA & Russia), industry and academia

The Robotic All-Terrain Lunar Exploration Rover (RATLER) — Increased Mobility Through Simplicity

J. Bryan Pletta
Sandia National Laboratories
Albuquerque, New Mexico

Abstract

The Robotic All-Terrain Lunar Exploration Rover (RATLER)
- Increased Mobility through Simplicity

J. Bryan Pletta
Dept. 9616
Sandia National Laboratories
Albuquerque, NM 87185

A new concept mobility chassis for a robotic rover is described which is inherently simple with few moving parts or complex linkages. The RATLER design utilizes a four-wheel drive, skid steered propulsion system in conjunction with passive articulation of the dual body vehicle. This uniquely simple method of chassis articulation allows all four wheels to remain in contact with the ground even while climbing obstacles as large as 1.3 wheel diameters. A prototype mobility platform has been built which is approximately 1 m^2 with 0.5 m diameter wheels and all-wheel electric drive. The theoretical mobility limitations are discussed and compared with the results of field trials of the prototype platform. The theoretical model contrasted with measured performance is then used to predict the expected mobility of the RATLER design on the Lunar surface.

Sandia National Laboratories
Transportation Systems Center
Advanced Vehicle Development Department
Robotic Vehicle Range

- **Research and Development:**
  - Mobile Robotics
  - Mobile Manipulators
  - Man-Machine Interfaces
- **Applications Programs:**
  - Intelligent Vehicle Highway Systems (IVHS)
  - Lunar Surface Exploration
  - Military Battlefield
  - DOE & DoD Physical Security

Sandia National Laboratories

The Robotic All-Terrain Lunar Exploration Rover (RATLER) Increased Mobility Through Simplicity

- **Outline**
  - Description of RATLER
  - Lunar surface characteristics
  - Results of pathfinder field trials
  - Summary

This project is funded by the DOE Laboratory Directed Research and Development (LDRD) Program, Sandia National Laboratories LDRD Project Office.

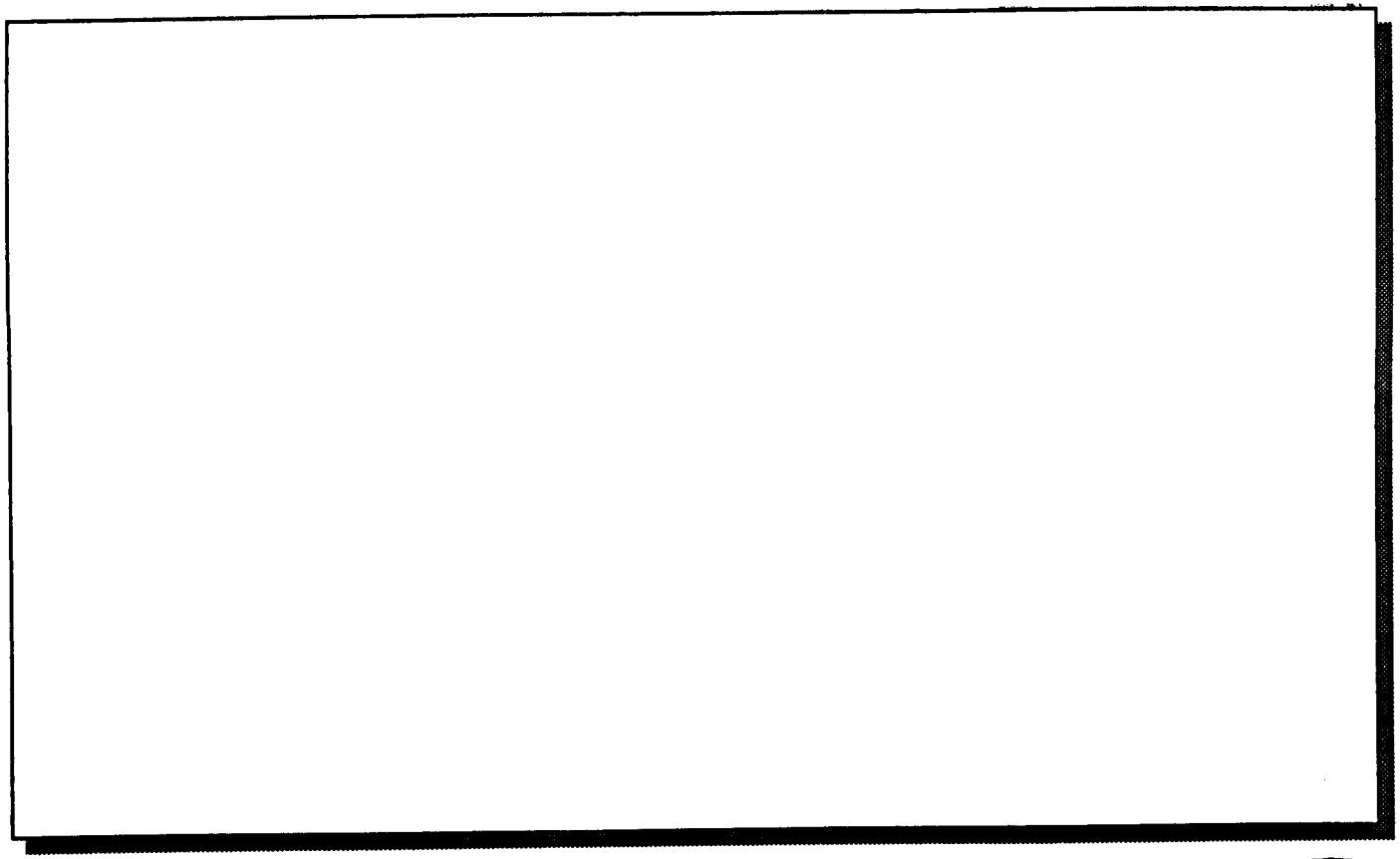
Sandia National Laboratories

RATLER Key Characteristics

- **Articulated chassis**
  - Central pivot
- **Four wheel electric drive**
  - All wheels remain in contact with terrain
  - 1/2 torque of conventional four wheeled vehicle for climbing
- **Simple construction**

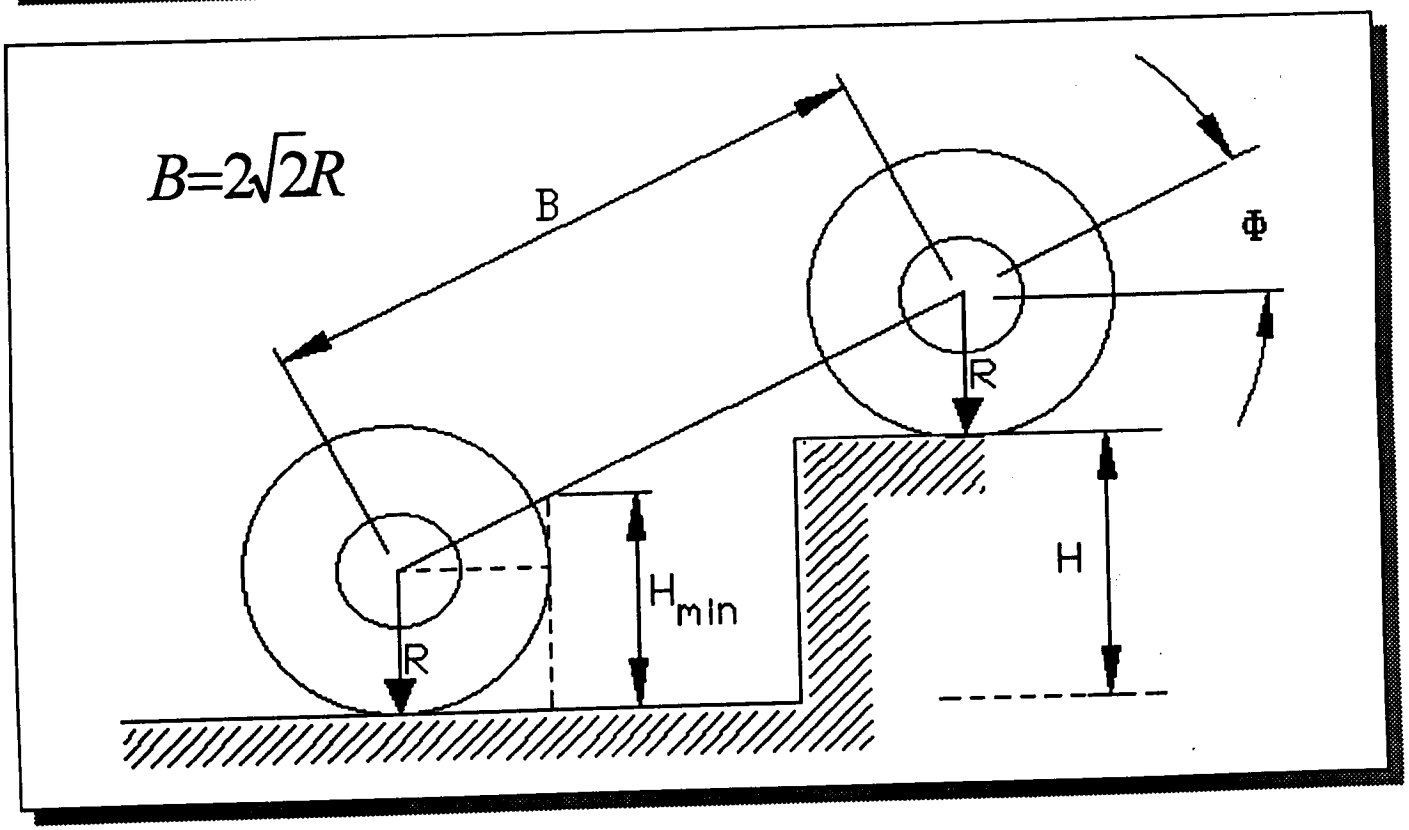
Sandia National Laboratories

RATLER Photo



Sandia National Laboratories

RATLER Optimum Wheel Base

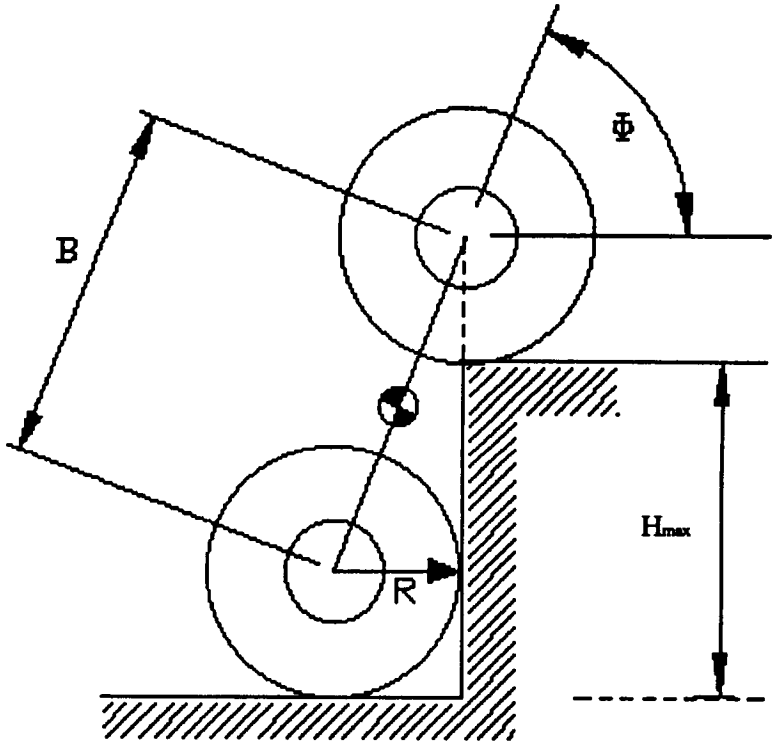


Sandia National Laboratories

ROBOTIC VEHICLE RANGE
SANDIA NATIONAL LABS

RATLER Maximum Step Height

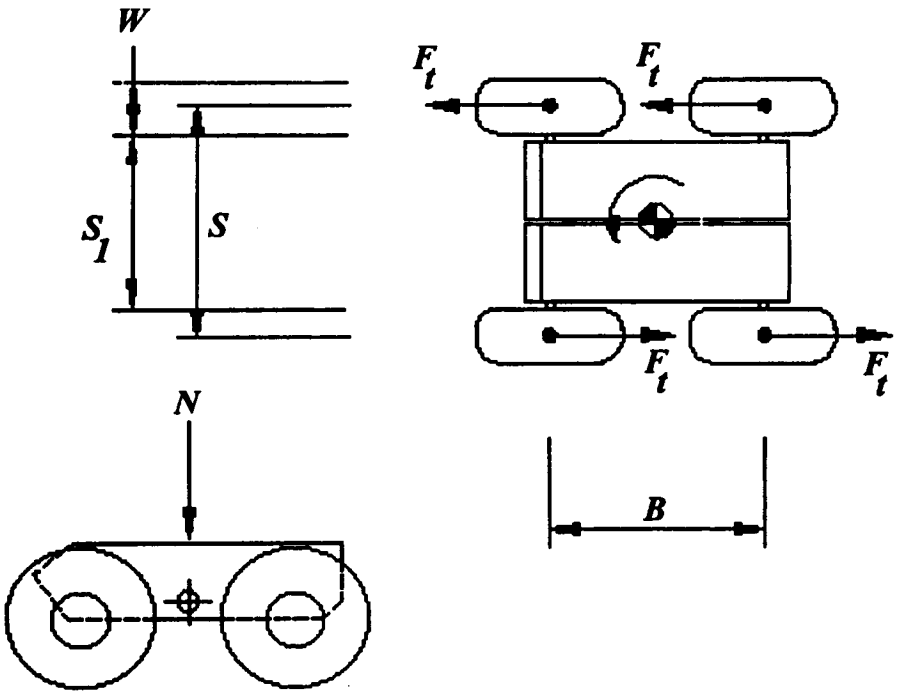
$H_{max}=\sqrt{7}R$



Sandia National Laboratories

RATLER Optimum Stance

$$S \geq B\mu_s/\mu$$



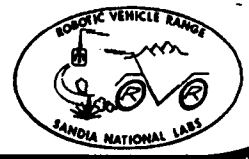
Sandia National Laboratories

ROBOTIC VEHICLE RANGE
SANDIA NATIONAL LABS

RATLER Pathfinder Configuration

» *Parameter*		*Units*
» Wheel Radius	24.8	cm
» Wheel Width	17	cm
» Wheelbase	72.4	cm
» Stance	81.3	cm
» Total mass	63	kg
» Total volume	.6	meters^3
» Maximum dimension	122	cm
» Slope Stability	~45	degrees
» Slope Climbing	22	degrees
» Crater Depth (traversable)	50	cm
» Boulder Height (traversable)	65	cm
» Ground Contact Pressure	3	kPa

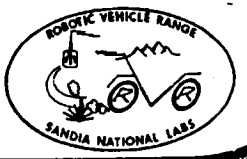
Sandia National Laboratories



Lunar Surface Characteristics
Slopes

- **Crater sides typically 10-30 deg**
  - Fresh craters may have slopes up to 45 deg
- **Mountains may possess slopes up to 40 deg**

Sandia National Laboratories

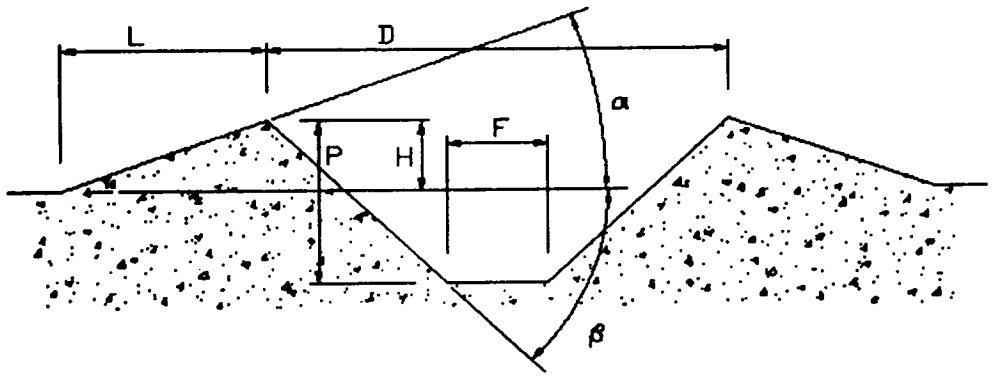


Lunar Surface Characteristics
Craters

- Consider only craters < 500 m in diameter
- 100 - 1000 craters/Km^2 in size of interest
- 70 - 90% of these are of the older degraded type

Sandia National Laboratories

Degraded Crater Model



D	H	P	F	L	α	β
meters	meters	meters	meters	meters	degrees	degrees
0.5	0.016	0.091	4.62E-05	0.118	7.8	20
50	1.726	9.511	0.157	12.433	7.9	20.9
5000	184.102	995.9	530.938	1307.952	8	24

Sandia National Laboratories

ROBOTIC VEHICLE RANGE
SANDIA NATIONAL LABS

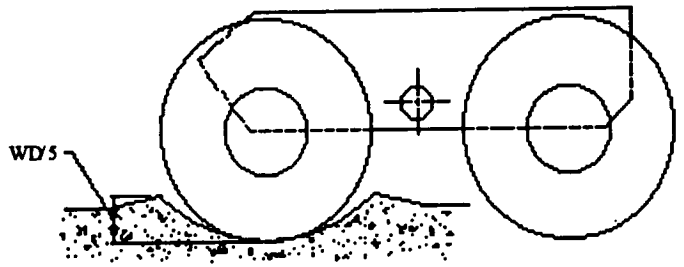
Lunar Surface Characteristics
Boulders

- **Boulder distribution from Lunokhod mission to Le Monier Crater**

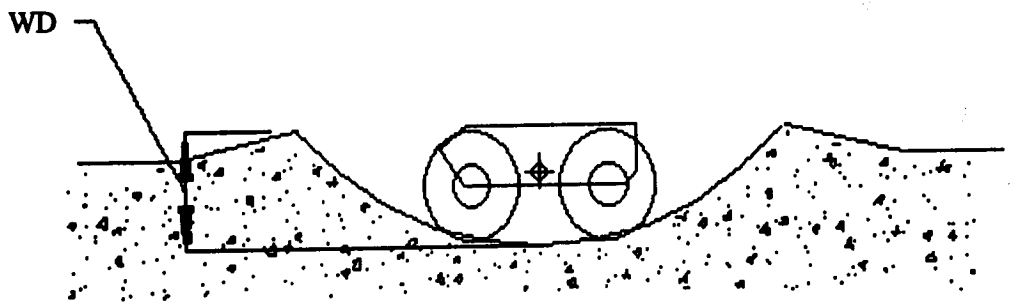
Size	# Observed	Area
Sb > 1 m	0	100 m^2
20 cm < Sb < 1 m	3	100 m^2
10 cm < Sb < 20 cm	10	100 m^2

Sandia National Laboratories

Lunar Surface Characteristics
Minimum and Maximum Significant Craters



Minimum Significant Crater Depth



Maximum Significant Crater Depth

Sandia National Laboratories

ROBOTIC VEHICLE RANGE
SANDIA NATIONAL LABS

RATLER Functional Requirements

Parameter	Upper Bound	Lower Bound	Units
Slope Climbing & Stability	45	0	degrees
Crater Depth (traversable)	WD	WD/5	wheel dia (meters)
Boulder Height (traversable)	WD*1.3	WD/10	wheel dia (meters)
Lander Deployment Height	1.4	~0.1	meters
Ground Contact Pressure	7	?	kPa
Total mass	65	~15	kg
Total volume	5.1	~0.75	meters^3
Maximum Dimension	2.8	~0.9	meters

Sandia National Laboratories

RATLER Pathfinder Field Trial Results

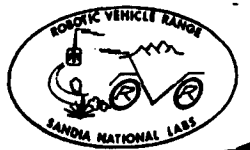
- 80:1 pancake motors, 15.9 Nm torque per wheel
  - successfully climbing 30 - 40 cm step-like rocks
  - 22 deg slope climbing
  - spin turns on high friction surfaces are difficult
- Higher torque in actual design should provide better climbing performance
  - 150:1 gearheads provide 22.7 Nm continuous torque

Sandia National Laboratories

RATLER Mechanical Design

- Pathfinder experience was very useful
- Initial layout concept done to fit current bodies
  - » 150:1 pancake motors; 1 ea, per wheel
  - » ~ 0.6 m/s top speed
  - » ~ 30° slope climbing capability
  - » ~ 40° static slope stability
  - » ~ 70 kg rover w/o payload
  - » ~ 18 kg payload capacity
  - » ~ 25° slope climbing w/ 18 kg payload

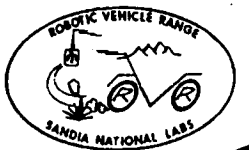
Sandia National Laboratories



RATLER Expected Lunar Mobility

- Limited surface information exists in fine detail
- Most boulder fields will not present significant obstacles
- Slope climbing is not a problem on older "degraded" craters
- Steep slopes in "new" craters and on mountainsides can be scaled via switchbacks

Sandia National Laboratories



Summary

- **RATLER Design's Advantages:**
  - Simpler than competing designs
  - Very good mobility characteristics
  - Compact, scales well
- **Current Program Status:**
  - Scale Models tested & documented
  - Analysis is ongoing
  - Pathfinder completed
  - Prototype development is underway
  - Initial Operational Capability projected for August 1993

Sandia National Laboratories

A Multitasking Behavioral Control System for the Robotic All-Terrain Lunar Exploration Rover (RATLER)

Paul Klarer
Sandia National Laboratories
Albuquerque, New Mexico

A Multitasking Behavioral Control System for the Robotic All Terrain Lunar Exploration Rover (RATLER)

P. Klarer

Sandia National Laboratories
Advanced Vehicle Development Department
Robotic Vehicle Range
Albuquerque, New Mexico

Abstract

An approach for a robotic control system which implements so called 'behavioral' control within a realtime multitasking architecture is proposed. The proposed system would attempt to ameliorate some of the problems noted by some researchers when implementing subsumptive or behavioral control systems, particularly with regard to multiple processor systems and realtime operations. The architecture is designed to allow synchronous operations between various behavior modules by taking advantage of a realtime multitasking system's intertask communications channels, and by implementing each behavior module and each interconnection node as a stand-alone task. The potential advantages of this approach over those previously described in the field are discussed. An implementation of the architecture is planned for a prototype Robotic All Terrain Lunar Exploration Rover (RATLER) currently under development, and is briefly described.

A Multitasking Behavioral Control System for the RATLER

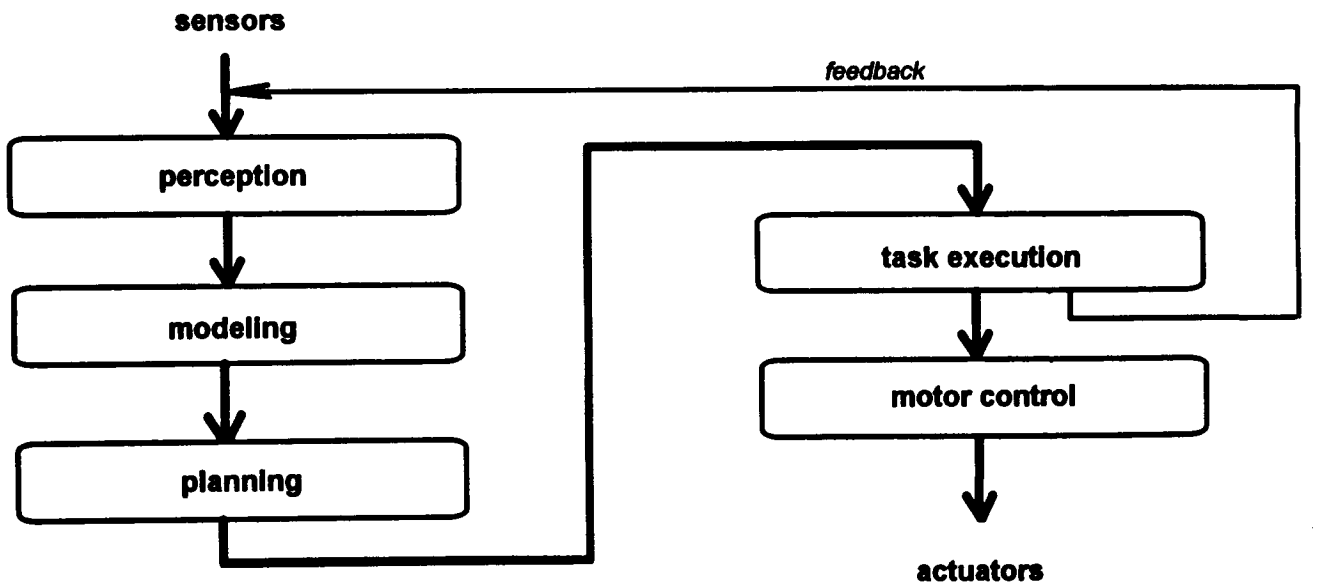
- **Outline**
  - Behavioral 'Minimalist' versus AI 'Traditionalist' Approaches to Autonomous Control
  - Generalized Hybrid System Description (Proposed)
  - Example Implementation Description
  - Summary

This project is funded by the DOE Laboratory Directed Research and Development (LDRD) Program, Sandia National Laboratories LDRD Project Office.

Sandia National Laboratories

A Multitasking Behavioral Control System for the RATLER

- **Traditional 'AI' Paradigm:**



Sandia National Laboratories

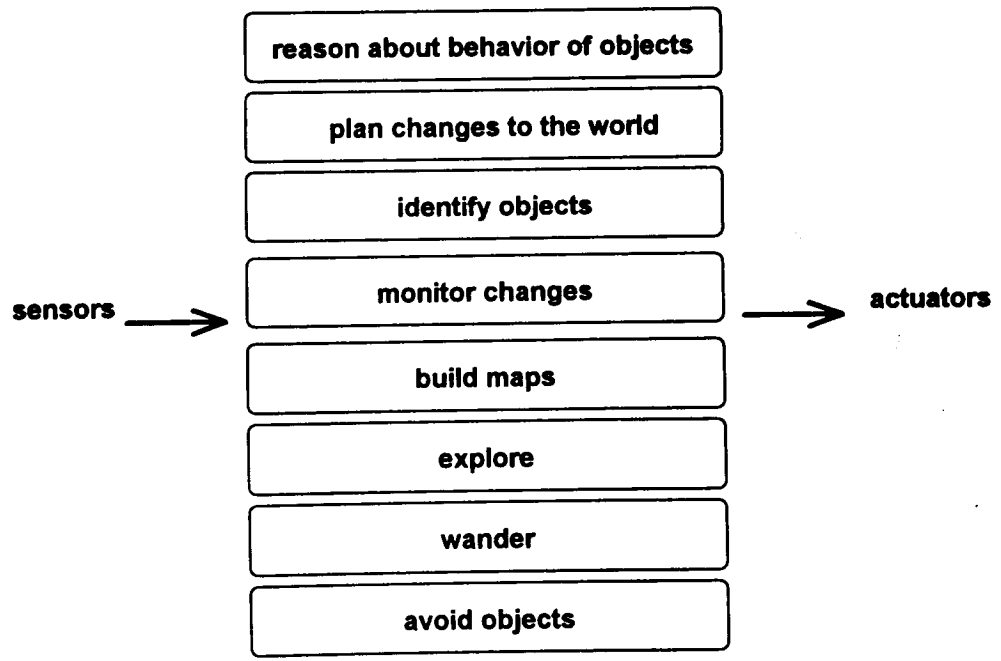
A Multitasking Behavioral Control System for the RATLER

- **Traditionalism**
  - Traditionalism as described by Moravec in 1983 *(The Stanford Cart & the CMU Rover)*
    - » World model is basis for perception and planning
    - » Depends highly on accurate sensor models for uncertainty factors
    - » Highly modularized, with synchronized module interactions
    - » Typically employs considerable computing power to achieve 'real-time' performance
    - » High autonomy, user interfaces generally very good

Sandia National Laboratories

A Multitasking Behavioral Control System for the RATLER

- **Minimalist Behavioral Paradigm:**



Sandia National Laboratories

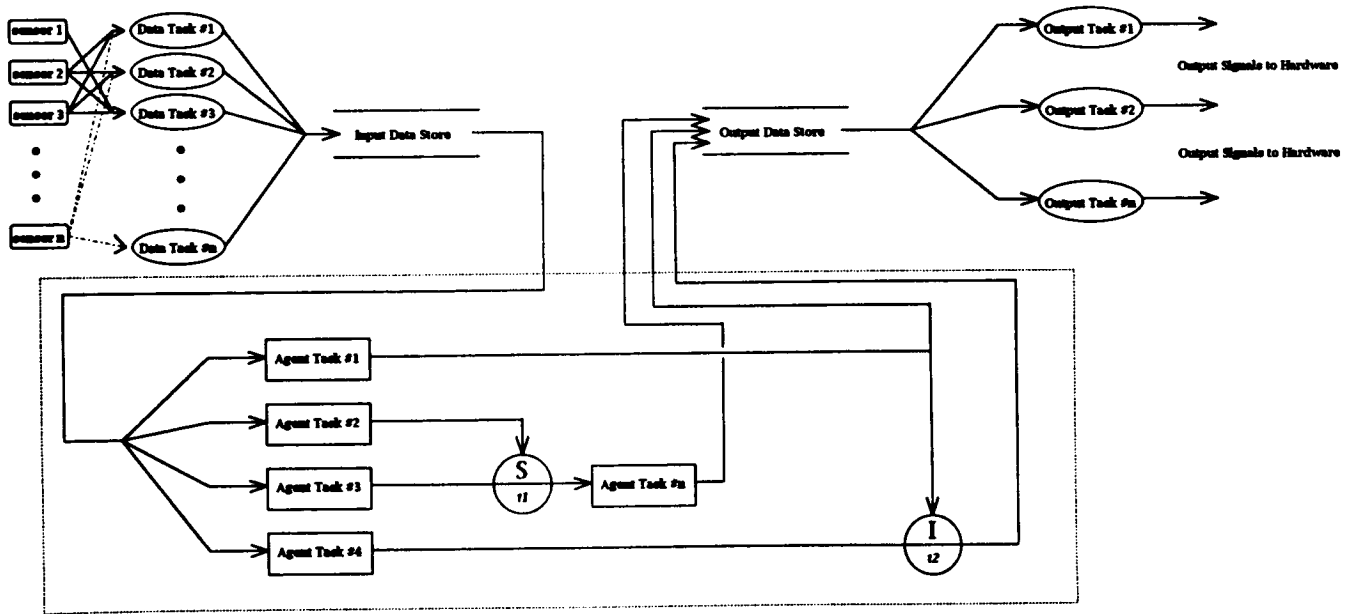
A Multitasking Behavioral Control System for the RATLER

- **Minimalism**
  - Minimalism as proposed by Brooks in 1986: *subsumption architecture*
    - » No internal state or world model is used
    - » Sensors provide real time view of world; ***the world IS the model***.
    - » Embodied as hierarchy of highly modularized ***levels of competence***
    - » Does not require synchronization between modules
    - » Simple computation, many small & inexpensive machines
    - » High autonomy, but user interface is limited

Sandia National Laboratories

A Multitasking Behavioral Control System for the RATLER

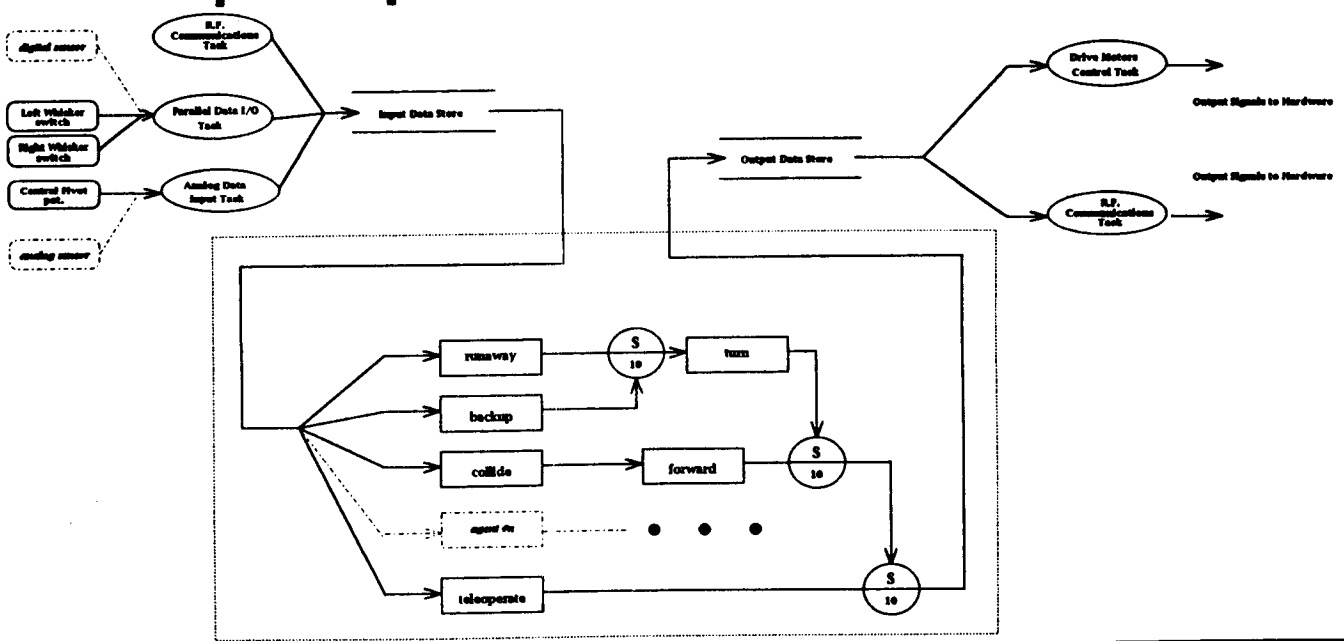
- Generalized Hybrid System (Proposed)



Sandia National Laboratories

A Multitasking Behavioral Control System for the RATLER

- Example Implementation for RATLER



Sandia National Laboratories

A Multitasking Behavioral Control System for the RATLER

- **Summary**
  - Realtime multitasking implementation proposed attempts to hybridize traditional and subsumption architectures.
  - Will specifically attempt to link a teleoperation capability with an 'anti-collision' behavior module.
  - Will be implemented on a multiprocessor machine, on a 32 bit backplane.
  - Expandable system will allow more processors, each is programmed independent of the others.
  - Coding will be done in C and C++.
  - Current scheduled target for initial operational capability is August 1993.

Sandia National Laboratories

PART II
Tour Presentations

Automated Structural Assembly Laboratory (ASAL)

Ralph Will
NASA Langley Research Center
Hampton, Virginia

A single robot arm constructs a 102 member planar truss with hexagonal reflector panels autonomous.y. The arm travels on as X-Y carriage and the truss is assembled on a rotating turntable. The truss hardware and end-effectors were developed in-house. Technology efforts have included automated error recovery, machine vision guidance, end-effector microprocessor control, path planning, sequence planning, artificial intelligence, automated end-effector change-out, and software design.

Intravehicular Automation and Robotics

Kelly Willshire
NASA Langley Research Center
Hampton, Virginia

A full-scale mockup of Space Station Freedom's Laboratory Module has been built. A robotically controlled 7-DOF arm rides on a track for mobility in the mockup. The purpose is to investigate how automation and robotics technology can improve the productivity of SSF experiments, especially when astronauts are absent. The facility will address protein crystal growth and furnace experiments in 1993 and 1994.

Flight Telerobotic Servicer Hydraulic Manipulator Testbed (FTS HMTB) and Vehicle Emulator System (VES)

Wallace Harrison and Robert L. Williams
NASA Langley Research Center
Hampton, Virginia

The FTS HMTB is the ground-based testbed for the FTS arm which will be delivered to NASA JSC in June 1993. The HMTB and its control system is identical to the flight arm, except that hydraulic actuation is required to lift representative space payloads in 1–G.

The VES is a six-legged hydraulic Stewart platform mechanism which is used to study disturbance compensation for external operations of space telerobotic systems. The class of devices modeled is compound manipulators, such as SPDM attached to the end of SSRMS, or a manipulator on a free-flying satellite. Manipulator arms will be mounted to a load cell on top the platform. Based on the inertial forces, an admittance model will drive the platform to emulate the motion of compliant manipulator vehicles in space.

Intelligent Systems Research Laboratory (ISRL)

Robert L. Williams and Ed Hogge
NASA Langley Research Center
Hampton, Virginia

The ISRL has developed dual arm, shared control of sensor-rich telerobotic systems. Several representative space tasks have been demonstrated over the years. Simultaneous control is possible combining hand controller inputs, automatic position commands, machine vision guidance, force control, and laser proximity control. Two standard PUMA arms have been used for some time, and two redundant 8-axis arms are recent additions.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1993	Conference Publication

4. TITLE AND SUBTITLE
Selected Topics in Robotics for Space Exploration

5. FUNDING NUMBERS
WU 233-03-01-01

6. AUTHOR(S)
Raymond C. Montgomery and Howard Kaufman, Editors

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA CP-10131

11. SUPPLEMENTARY NOTES
Raymond C. Montgomery: Langley Research Center, Hampton, Virginia
Howard Kaufman: Rensselaer Polytechnic Institue, Troy, New York

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified-Unlimited

Subject Category 12

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The papers and abstracts contained in this report represent both formal presentations and experimental demonstrations at the Workshop on Selected Topics in Robotics for Space Exploration which took place at NASA Langley Research Center, March 17 and 18, 1993. The workshop was cosponsored by the Guidance, Navigation, and Control Technical Committee of the NASA Langley Research Center and the Center for Intelligent Robotic Systems for Space Exploration (CIRSSE) at Rensselaer Polytechnic Institute,Troy, New York. Participation was from industry, government, and other universities with close ties to either Langley Research Center or to CIRSSE. The presentations were very broad in scope with attention given to space assembly, space exploration, flexible structure control, and telerobotics.

14. SUBJECT TERMS
Robotics, Telerobotics, Space Exploration, Space Assembly
Flexible Structure Control

15. NUMBER OF PAGES
294

16. PRICE CODE
A13

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified		

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

END DATE APRIL 7, 1994