(NASA-CR-1844[illegible]0) KC-135 MATERIALS HANDLING ROBOTICS Final Report, 9 Feb. 1990 - 1 Feb. 1991 (Alabama Univ.) 25 p

N93-11992

Unclas

G3/37 0019730

IN-37

P-25

Final Report

submitted to

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER, ALABAMA 35812

April 23, 1991

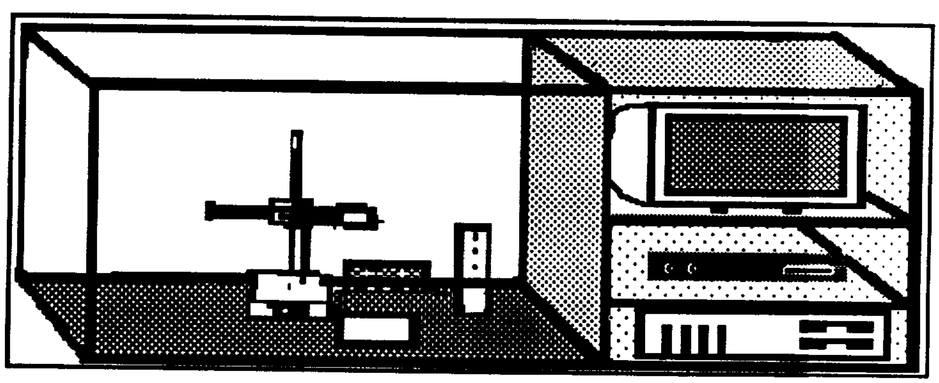
for NAS8 -36955 D.O. 78

entitled

KC-135 Materials Handling Robotics

by

Gary L. Workman, PhD.
Principal Investigator



Johnson Research Center
The University of Alabama in Huntsville
Huntsville, Alabama 35899

The University Of Alabama In Huntsville

Huntsville, Alabama 35899
(205) 895-6000
Telefax: (205) 895-6677

Research Administration

April 26, 1991

National Aeronautics and Space Administration
ATTN: Elaine Hinman/EB24
George C. Marshall Space Flight Center
MSFC, AL 35812

RE: Final Report
Contract No. NAS8-36955, D.O. #78

Dear Ms. Hinman:

Please find enclosed six (6) copies of the final report for the period of February 9, 1990 to February 8, 1991 as required by the above referenced contract. Additional distribution has been made as indicated below.

TABLE OF CONTENTS

INTRODUCTION 1

SPACE ROBOTICS.. 1

PREVIOUS WORK 2

CONCEPTS OF ROBOT DYNAMICS 4

RESULTS 7

CONCLUSIONS 14

SUMMARY 14

ACKNOWLEDGEMENTS 15

BIBLIOGRAPHY 16

APPENDIX A. AMIGA PROGRAMS 19

INTRODUCTION

Robot dynamics and control will become an important issue for implementing productive platforms in Space. Robotic operations will become necessary for man-tended stations and for efficient performance of routine operations in a manned platform. The current constraints on the use of robotic devices in a microgravity environment appears to be due to an anticipated increase in acceleration levels due to manipulator motion and for safety concerns. The objective of this study will be to provide baseline data to meet that need.

Most texts and papers dealing with kinematics and dynamics of robots assume that the manipulator is composed of joints separated by rigid links. However; in recent years several groups have begun to study the dynamics of flexible manipulators, primarily for applying robots in space and for improving the efficiency and precision of robotic systems. Some of the reasons for pursuing research in this area are:

1. Current industrial robot systems weigh substantially more than the payload that they are able to transport. For instance a 250 pound robot can usually only transport 8- 10 pounds at the end-effector. Hence by increasing the ratio of payload to arm weight will substantially improve the economics for using robots through more efficient usage of materials and power.

2. Improving the precision of robotic systems for performing delicate tasks will enable a broader base for robotics applications.

3. A number of applications, such as welding, deburring, etc. require trajectory tracking which may exceed the capability of today's industrial robot base, particularly when unexpected parameters cause the robot motion to deviate from the pre-programmed paths.

SPACE ROBOTICS

Robotic systems which are being planned for implementation in space have a number of constraints to overcome. [1 - 4] The need to quantify some of the characteristics that are required for structures such as the Space Station is very important at this time, even though the IOC will be 5 -10 years from now.

Additional concepts which have to be worked out in any robotic implementation for a space platform include teleoperation and degree of autonomous control. Lightweight arms are necessary for space, primarily for the cost benefits derived from their reduced weight. However, lighter weight arms have to necessarily flex during movement. Flexure of the arms performing a task requiring precision requires some control mechanism to insure that the end-effector is at the proper place and orientation with respect to the workpiece during any contact period.

Flexing motions of the arm can cause (1) accelerations to feed back into the base support of the robot and into the Space Station structure, (2) transmit accelerations into the sample being transported, or (3)take forever to perform a task. The first effect is obviously detrimental to the microgravity environment of the Space Station, while the second will impact experiments such as the delicate protein crystals which are to be grown in space and probably transported with robotic or telerobotic arms. In some cases slow movements may be acceptable for (3); however it certainly will not be suitable for most tasks required for operation on the Space Station.

One must also include the reasoning that for man and robots to co-exist in the space environment, the robot must be non-threatening to man. Lightweight arms satisfy that criteria. For space applications, a Cincinnati-Milacron T3 is not only over-weight, but it also is threatening to humans trespassing in its working volume.

PREVIOUS WORK

The robot used for the initial study in NAS8-36620 was a UMI RTX robot, which was adapted to operate in a materials processing work cell to simulate sample changing in a microgravity environment. The robotic workcell was flown several times on the KC-135 aircraft at Ellington Field. The primary objective of the initial flights was to determine operating characteristics of both the robot and the operator in the variable gravity of the KC-135 during parabolic manuevers.

This study demonstrated that the KC-135 aircraft can be used for observing dynamics of robotic manipulators. We also observed the difficulties associated with humans performing teleoperation tasks during varying G levels and provided insight into some areas in which the use of artificial techniques would provide improved system performance.

Allowing some degree of autonomy due to time delay communications for teleoperation over large distances is also necessary. The robot controls then will have certain motions embedded in the control software that do not need explicit operator communication, except for abort. Time delays make for precarious situations in performing teleoperation from large distances. Tasks can then be accomplished in a more reasonable manner and more successfully.

A major goal of the earlier study was to evaluate a small robot system, such as the UMI RTX, for materials processing applications in low gravity and to determine the characteristics of a robot arm in a space environment, particularly with respect to accelerations which might impact materials grown on a space platform.

A materials transfer workcell was assembled to simulate the changing of sample ampoules as might occur aboard a space laboratory. Accelerometer packages were included for determining the G levels of the workcell and the at the end-effector. Several KC-135 flights were made with the workcell, improving some data taking capabilities each mission. The human operator was able to train the robot to perform a materials transfer function within the 20 seconds desired. The first computer used with the experiment was a Toshiba 3100 which did not allow for both control of the robot and reading of the accelerometer pacKage at the same time. The microprocessor in that system was an 80286. In order to try to improve upon the multi-tasking capability of the computer, a 386 based computer was selected. This choice enabled some improvements in the data acquisition process; but the multi-tasking software used at that time still did not permit the I/O commands to the robot to operate properly. Consequently we never did get to control the robot and take acceleration data simultaneously in these flights.

A number of lessons were learned with this series of experiments. The RTX robot uses plastic belts for actuation of the links and optical encoders for position and velocity control. The slippage and flexing of the belts caused excessive jitter and accelerations at the end-effector. We believe that the belt-driven actuation would not be acceptable for experiments such as the protein crystal growth studies due to the lack of control of accelerations at the end-effector. The control system; however, is PID and appeared to work well whether the task was learned in 1 G and performed in low G or vice-versa. However, it was tedious to teach the robot during parabolas, mainly because we had few visual aids to assist in the correct orientation of the end effector. A particularly sensitive task was inserting the sample ampoule into its holder. A borrowed fiber-optic borescope provided little depth perception and was not useful for this study. In addition teaching a robotic device for precision movements can certainly be improved through more innovative approaches using embedded sensors or vision systems with some autonomous local control using artificial intelligence techniques.

For example, protein crystal growth experimentation has great interest at this time and will be a primary space experiment. The robot which has historically been used within that program for sample preparation and is within a reasonable price range, is the Zymate II robot manufactured by the Zymark Corporation. It possesses a more sophisticated controller than the UMI RTX used in the earlier study and has a greater potential for working in a multi-tasking environment.

The computer controller is important because robot programming, robot execution and data acquisition of the accelerometers has to be performed during parabolic manuevers. The MS-DOS platform used in earlier study doesn't allow all those functions to occur simultaneously. We chose to move to a more natural multi-tasking environment offered by the Commodore Amiga for the second phase activities.

The focus on protein crystal growth also required a different robot workcell to be designed and fabricated. Another major concept which might be important in terms of promoting telescience experiments is to use the KC-135 to implement the above experiments with remote manipulation from the ground. The KC-135 aircraft facility personnel at JSC have indicated interest in these types of experiments using the TDDRS satellite for transmission purposes.

CONCEPTS OF ROBOT DYNAMICS

The Newton-Euler Equations state be used to describe the effective forces distributed throughout the robotic system. In this work, we are particularly interested in the residual forces present at the end-effector.

$f_i = \sum m_i r_i$ Forces due to linear accelerations

$T_i = \sum (I_i w_i + w_i \times I_i w_i)$ Torques due to rotational velocities and accelerations

Note that these torques are referred to as dynamic torques since they arise only during motion, whereas static torques can exist such as when power is required to maintain a load against gravity. The three types of dynamic torques present in a moving robot arm are:

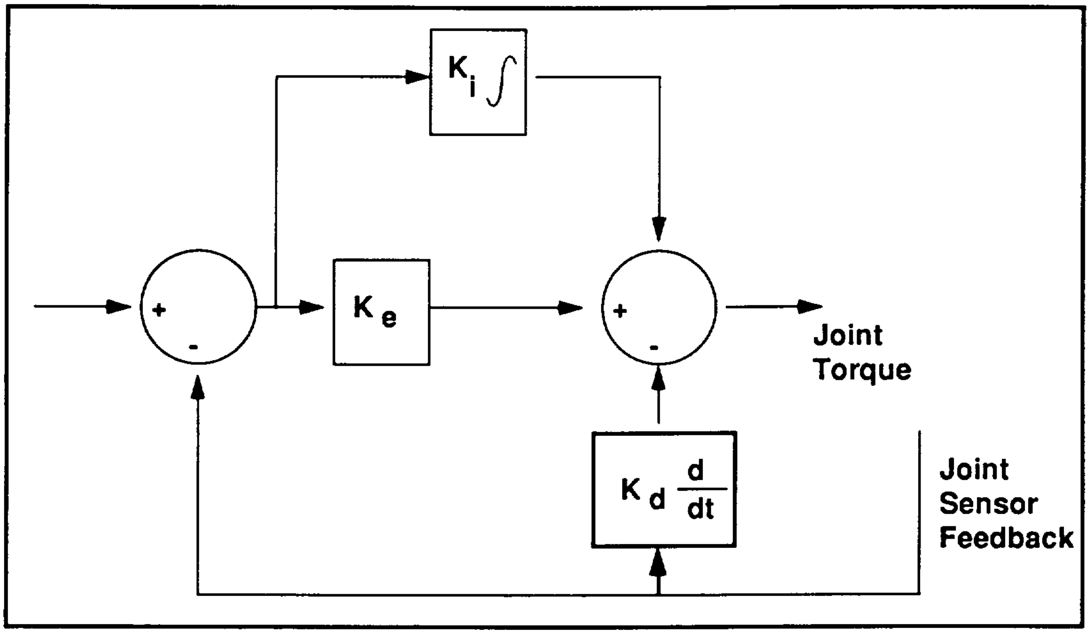
inertial torques - proportional to joint accelerations.
centripetal torques - proportional to velocities squared
Coriolis torques - proportional to joint velocities from two different links.

The inertial forces and torques arise from the normal action/reaction forces when accelerating a body. Obviously in order to move the arm from one resting position to another, both acceleration and decceleration will occur. Centripetal torques arise from contrained rotation about an axis. These torques will be present whenever a single rotary motion is executed. The Coriolis forces arise from the interaction of two simultaneously rotating systems.

When expanded in this manner, the dynamic equations increase in complexity with the number of joints. Each manipulator will have its own characteristic set of relations describing these torques. The complexity of the relations vary due to the joint interaction of these inertial, centripetal, and Coriolis torques.

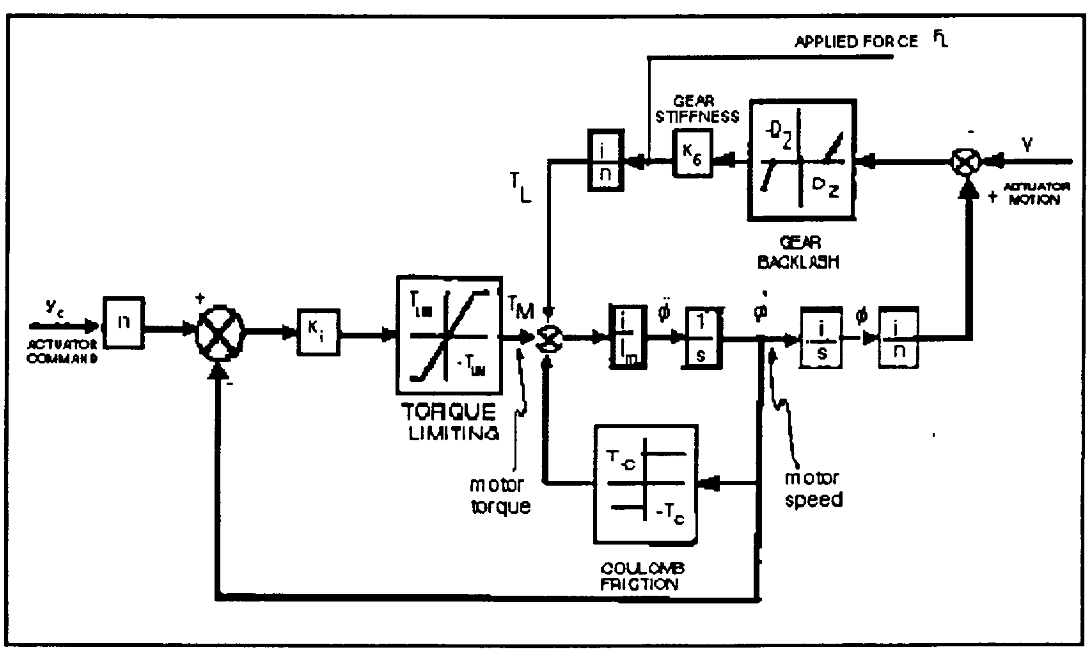
In order to provide some control over a robotic system in terms of a consistent and reproducible path generation for varying payloads, a PID controller is normally used. Figure 1 shows in a simplified form the normal block diagram used for such a controller.[8] In order to compensate for gravity, for example, the PID controller allows torque to be delivered to the motors just to maintain position. Additional torque is required for movement.

Figure 1. Simplified block diagram for a PID controller (taken from Snyder[8])



A more complex controller is available in Treetops which includes more variables such as friction, gear backlash, and stiffness. The block diagram for Motor4 is shown in Figure 2.

Figure 2. Block diagram for Motor4 as taken from the Treetops Users Manual



RESULTS

The work performed in this contract consisted of two distinct activities. One set was primarily concerned with the construction of a Zymate robotic workcell which could fly on the KC-135 and take accelerometer data for determining the g-levels which would be present within the robotic system when it was operating in a reduced gravity environment. The other set of activities concentrated on ground-based applications in order to determine what improvements were able to be attaiined in improving the preciseness of performing delicate robotic tasks, even in 1-g. These were primarily performed at the Alabama Center for Advanced Technology Transfer, (ACATT) and were performed with a Puma 560 robot owned by the Boeing Aerospace Corporation.

ZYMATE ROBOT CELL

The robot workcell which was constructed for the KC-135 simulations is shown in Figure 3. The workcell was flown in June, 1990 for a systems check-out flight. Unfortunately during the remainder of the contract, the KC-135 did not fly so we were unable to finish up all the sub-systems performing in the reduced gravity environment.

FIGURE 3. Schematic of Zymate Workcell for KC-135 Experiments

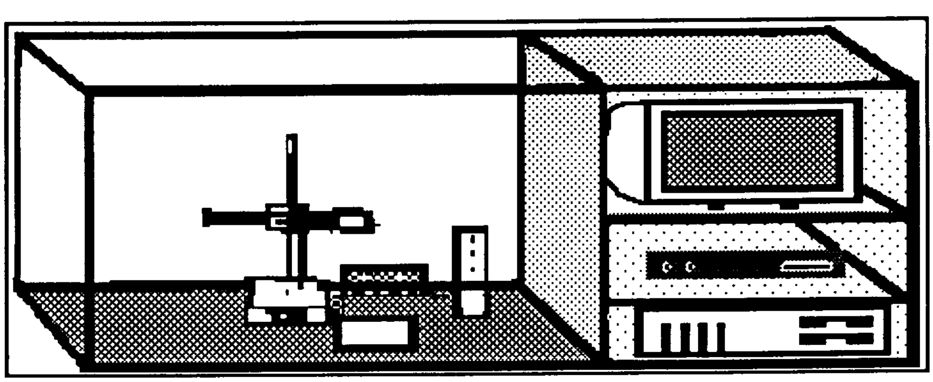
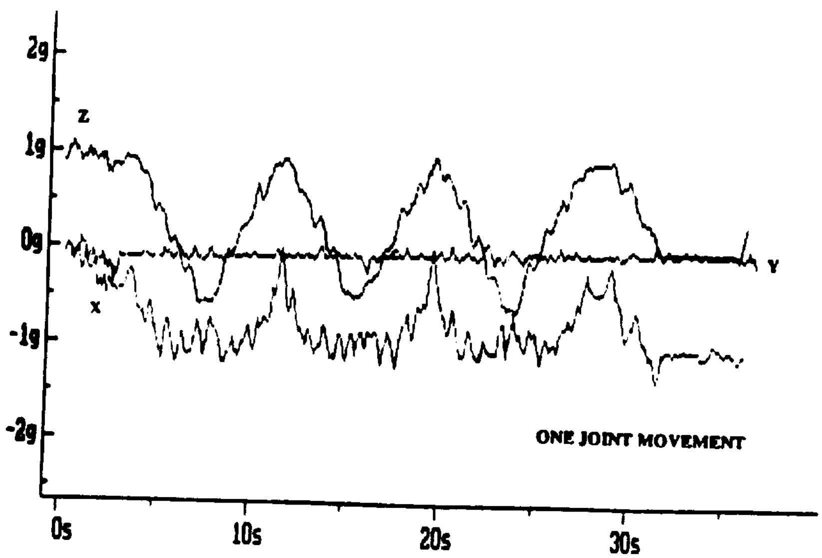
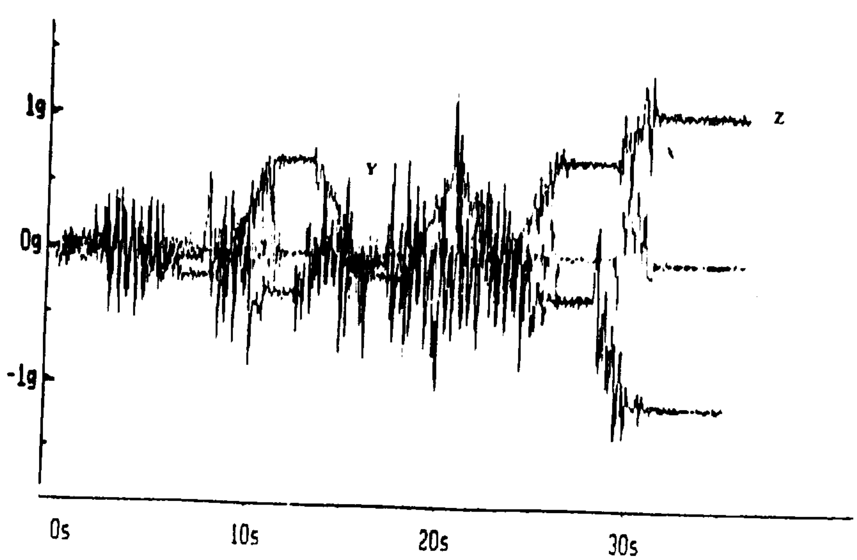


Figure 4. Acceration Measurements on Puma 560 Robot

a.) Overlays of accelerations in x, y, and z vectors for a one rotational joint motion.



b.) Overlays of x, y, and z accelerations during multiple motion trajectories.



TREETOPS SIMULATIONS

In parallel with the KC-135 experiments, we have also begun a number of simulated experiments with the intent of developing a better understanding of the dynamics of both rigid and flexible manipulators. The TREETOPS Simulation software was obtained from Dr. Henry Waites' group at Marshall Space Flight Center. One of the graduate students, Mr. Houchang Li has been performing these simulations on a VAX 785 located at the University. We feel that these simulations will be applicable to both reduced gravity and ground-based robotic applications.

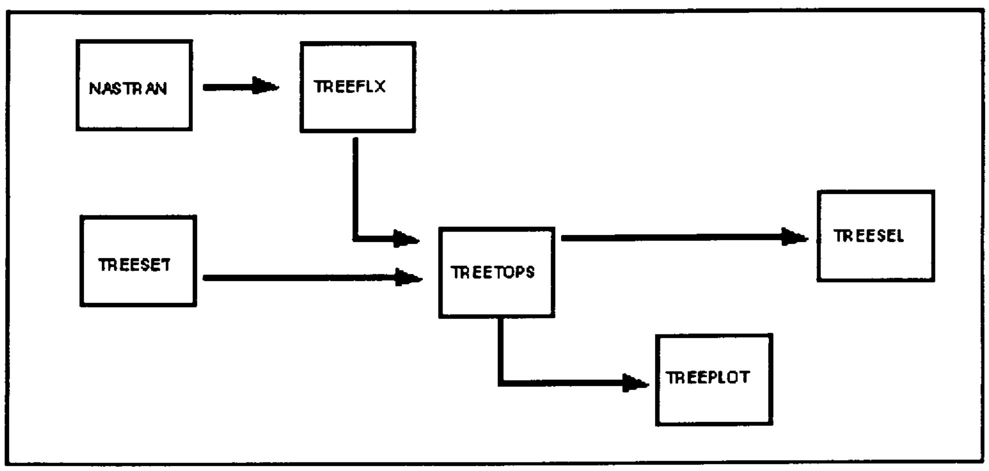
TREETOPS provides availability to a number of devices, actuators, sensors, etc. Figure 5 shows a table listing some of more significant components used for robotics simulations.

Figure 5. Simulation Components in TREETOPS which are pertinent to robotics.

COMPONENTS AVAILBLE TO THE USER FOR TREETOPS SIMULATION					
BODY	HINGES	SENSORS	ACTUATORS	FUNCTION GENERATORS	INTERCONNECTIONS
RIGID	ROTATION	RATE GYRO	REACTION JET	STEP	LINEAR SPRING
FLEXIBLE	TRANSLATION	RESOLVER	HYDRAULIC CYLINDER	RAMP	QUADRATIC SPRING
NASTRAN	MODAL	ANGULAR ACCELERATION	MOMENT ACTUATOR	PULSE	SOLID DAMPER
		VELOCITY	TORQUE MOTOR BRAKE	TRIANGLE	UPPER HARDSTOP
		POSITION	MAGNET	SINE	LOWER HARDSTOP
		TACHOMETER	GIMBAL CMG	NOISE	
		INTEGRATING GYRO	REACTION WHEEL	DOUBLET	
		IMU SENSOR	LOCK	USER	
		POSITION SENSOR	MOTOR DRIVES		
		VELOCITY VECTOR			
		CMG RESOLVER			
		CMG TACHOMETER			

In order to access these devices, TREETOPS provides several modules which allow one to enter the desired input parameters and simulate specific time histories. Figure 6 shows how these modules fit together.

Figure 6. TREETOPS Architecture



In cooperation with the Alabama Center for Advanced Technology Transfer, a graduate student in Mechanical Engineering has been using TREETOPS to model the dynamics of several industrial robot systems. Mr. Houchang Li has worked on models of both a Puma 560 for activities at ACATT to improve the the precision of robotic systems and the Zymate for this project. Examples of the dynamics which can be modeled using TREETOPS can be seen in the following analysis.

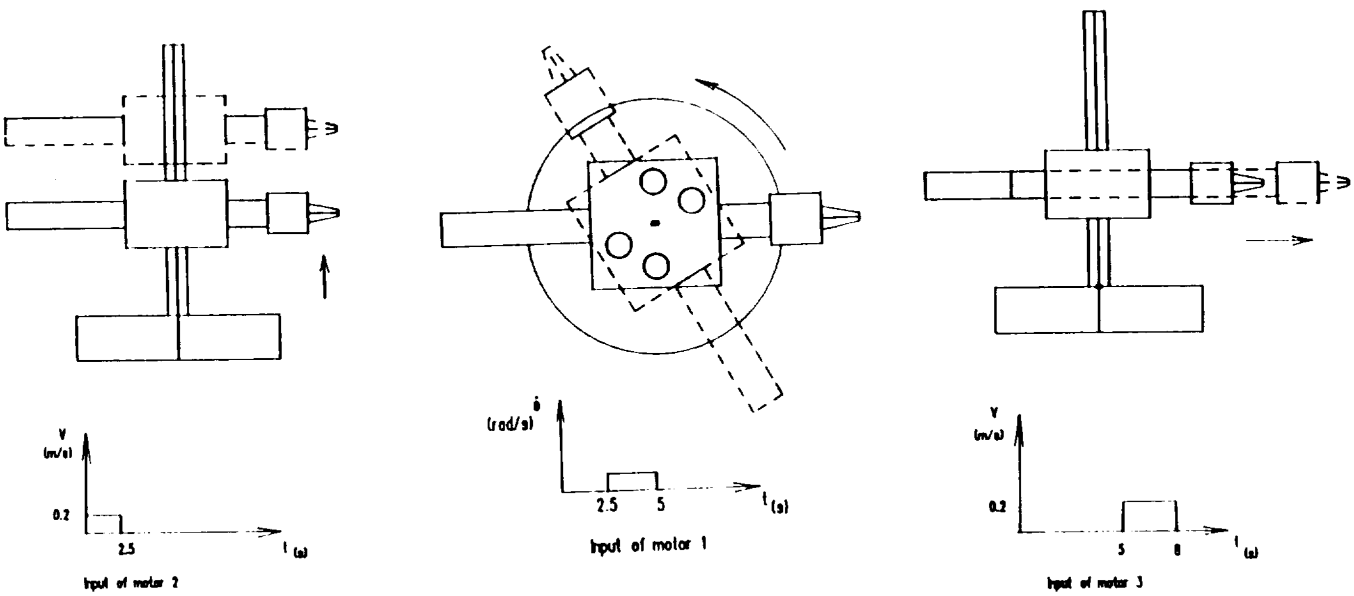
Using the motor4 model available in TREETOPS to drive the robot arm in three different motions, the effect of gravity can be investigated. Three different trajectories are displayed in Figure 7:

a.) A vertical motion into the motor labelled 2.

b.) A rotation about the vertical axis by motor 1.

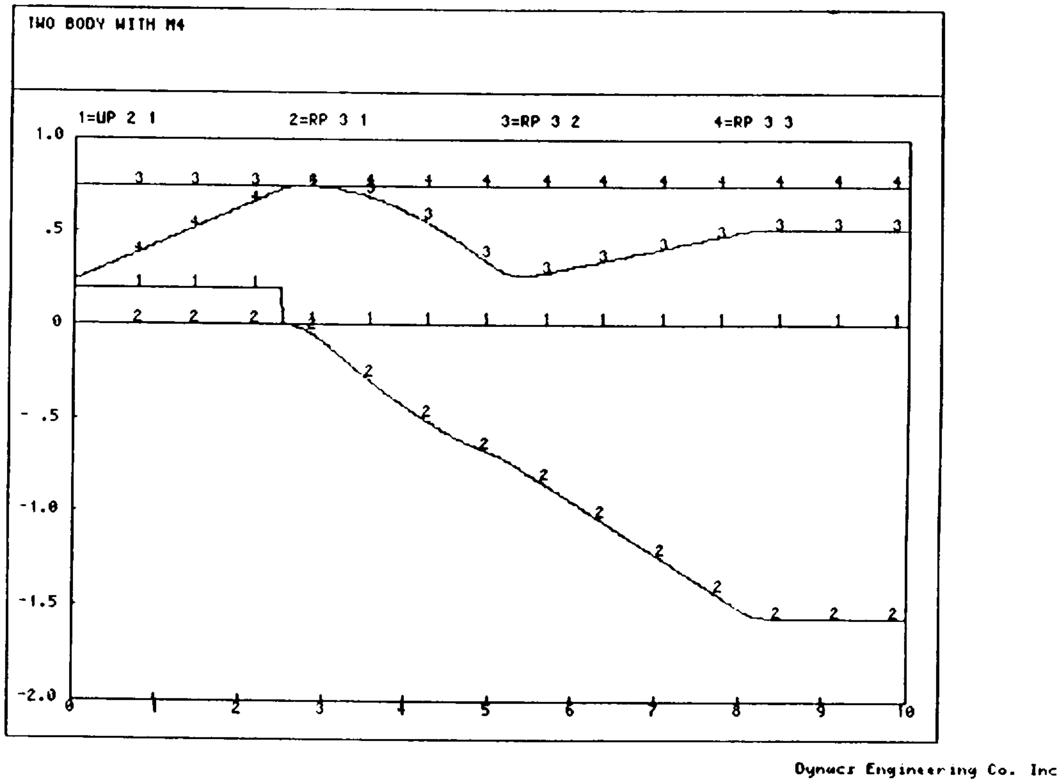
c.) A prismatic motion outward using motor 3.

Figure 7. Illustration showing sequences of motion for Zymate robot used in TREETOPS simulation.



The three trajectories are entered into TREETOPS as motions of the three links and the end-effector and the resulting time history of the motions are plotted in Figure 8. Note that the vertical motion follows the input pulse into motor 2 for the vertical motion from 0 to 2.5 seconds. This trace is labelled 1 in the figure. In order to simplify the chart, the other two input waveforms are left off; however all three motions of the robot are plotted.

Figure 8. TREETOPS plot showing the motion of the end-effector for the trajectory described in Figure 7.



The corresponding motions for the other links and the end-effector can be determined by relating the time motion that is occuring with the motor activated waveforms from Figure 7. Note that the label 2 refers to a displacement about the rotation axis, the label 3 refers to a displacement along the prismatic axis, and 4 is the vertical displacement of the end-effector.

The effect of gravity can be seen in the next two figures. Figure 9 shows the how the dynamics (or accelerations) at the end-effector occur during the performance of the trajectories with the gravity vector turned on. In Figure 10, the same motions are performed with the gravity vector turned off. Note that in the absence of gravity, there are more acceleration components occuring at the end-effector than in 1 g. Gravity does then tend to damp out some oscillations of the end-effector. In microgravity then for the same manipulator and control scheme, one would expect more oscillations.

Figure 9. The TREETOPS output for the vertical direction for the simulation decribed above with gravity.

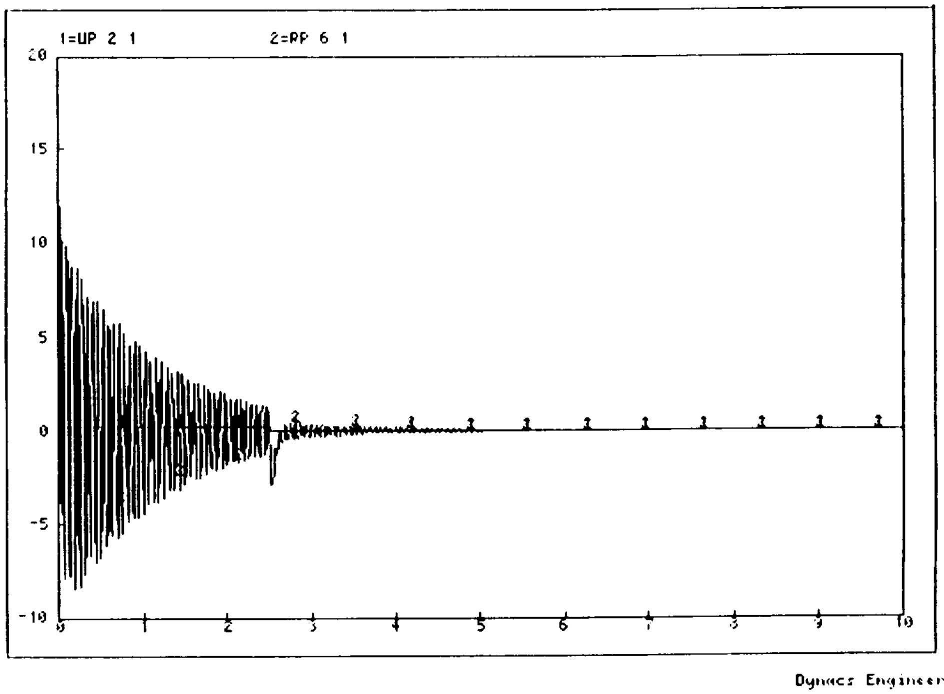
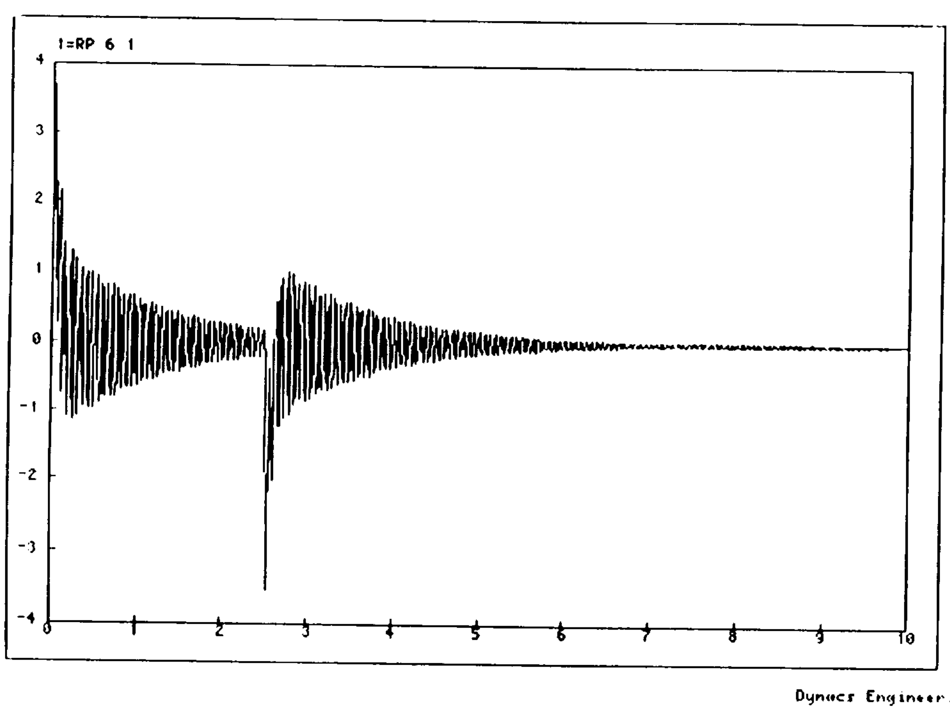


Figure 10. The TREETOPS output for the vertical direction for the simulation decribed above without gravity.



Conclusions

This work has resulted in a robot workcell which is qualified to fly on the KC-135 for measuring the responses of robotic devices in low gravity. Due to the 20-25 seconds of low gravity available during parabolic flight, one has to define trajectories which fit into that time period. In simulating sample changing operations, that type of trajectory works very well. Other trajectories should be suitable; however, if they are functional tasks that a robotic system would be performing in space, then they are more easily accepted as useful trajectories. More woerk needs to be done in identifying such tasks.

Another improvement that would be beneficial in the future is a better accelerometer system and interconnections to the computer. For this effort, accelerometers were borrowed from another KC-135 experiment. A smaller, lightweight triaxial accelerometer mount needs to replace the current single axis one used in this work due its overall mass.

Other improvements conceptually include a wireless signal for the accelerometer measurement, so that the cables do not interfer with the motion of the robot arm and also to reduce the potential for the cables to impart vibration to the accelerometer itself. These ideas can be looked at more closely in future projects.

During the course of this research a number of reports which dealt with acceleration control of robotic systems were found. The goals of these researchers were quite similar to ours in trying to improve precision and payload capability for industrial applications. These papers are listed in the bibliography and should be consulted for more information.

Summary

This research has generated some significant results in developing a robotic workcell for performing robotics research on the KC-135 in preparation for space-based robotics applications in the future. In addition, we have shown that TREETOPS can be used to simulate the dynamics of robot manipulators for both space and ground-based applications.

Acknowledgements

We wish to thank Ms. Elaine Hinman for her assistance in obtaining this contract and for her many efforts in performing the KC-135 experiments. Also to Guy Smith and his staff for constructing the robot frame, and George Meyers at MSFC for help in getting TREETOPS running at the Uninversity. Also we wish to thank ACATT for their assistance in funding Mr. Li to perform the TREETOPS simulations.

BIBLIOGRAPHY

1. Heer, E. and Bejczy, A.K.; Control of Robot Manipulators for Handling and Assembly in Space"; Mechanism and Machine Theory, 18, pp 23 - 34, 1981

2. R. A. Clift, "Program Plan for the Astronaut's Apprentice," Robotics 1, (1985) 251-264

3. D. R. Criswell, "Robotics and the Space Station," Robotics 1 (1985) 205-222

4. M.M. Clarke, "Recent Advances in Teleoperation: Implications for the Space Station," SPACETECH 85

5. W.H. Sunada and S. Dubowsky, "On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators with Elastic Members," Trans. ASME Mech. Transmission Automat. Design 105, (1983) 42-51

6. Fukuda, T.; "Flexibility Control of Elastic Robot Arms"; Jour. Robotic Systems 2, pp 73 - 88 (1985)

7. Brady, M., J.M. Hollarbach, et. al., "Robot Motion: Planning and Control", MIT Press, Cambridge, 1983

8. Snyder, W.E. " Industrial Robots: Computer Interfacing and Control", Prentice-Hall, Englewood Cliffs, 1985

9. Wang, P.K.C. and J. Wei; "Feedback Control of Vibrations in a Moving Flexible Robot Arm with Rotary and Prismatic Joints"; Proc. IEEE Int. Conf. on Robotics and Automation, Raleigh, N.C., 1987

10. Wang, P.K.C. and J. Wei; "Vibrations in a Moving Flexible Robot "; Jour. of Sound and Vibration 116, pp 149 -160 (1987)

11. Hannaford, B. and W. Rosar, "Effects of Vibrations on Grasp Control", NASA TECH BRIEF 13, #11, Item # 28, JPL Invention Report NPO-17698/7202

12. Van Winssen, J.C. and C.W. deSilva, "Accurate Trajectory Control of Robotic Manipualtors", CMU-RI-TR-85-15

13. Khosla, P.K. and T. Kanade, "Experimental Evaluation of Nonlinear Feedback and Feedforward Control Schemes for Manipulators", CMU-RI-TR-87-7

14. Khosla, P.K., "Choosing Sampling Rates for Robot Control", CMU-RI-TR-87-5

15. Khosla, P.K. and T. Kanade, "Real-Time Implementation and Evaluation of Computed-Torque Scheme", CMU-RI-TR-87-6

16. Khosla, P.K. and T. Kanade, "An Algoritm to Estimate Manipulator Dynamics Parameters", CMU-RI-TR-87-7

17. Khosla, P.K., "Estimation of Robot Dynamics Parameters: Theory and Application", CMU-RI-TR-87-8

18. Kim, Y. and Subhas Desa, "Acceleration Properties of Planar Manipulators from a Dynamic Standpoint", CMU-RI-TR-88-16

19. Chung, C.L., S. Desa, and C.W. deSilva, "Base Reaction Optimization of Redundant Manipulators for Space Applications", CMU-RI-TR-88-17

20. Feliu, V. , H. B. Brown Jr., and K.S. Rattan, "Design and Control of a Two-Degree-of-Freedom Lightweight Flexible Arm"; CMU-RI-TR-89-21

21. Feliu, V. , K.S. Rattan, and H. B. Brown Jr. "A New Approach to Control Single-Link Flexible Arms. Part II. Control of the Tip Position in the Presence of Joint Friction"; CMU-RI-TR-89-14

22. Feliu, V. , K.S. Rattan, and H. B. Brown Jr. "A New Approach to Control Single-Link Flexible Arms. Part II. Adative Control of the Tip Position with Payload Changes"; CMU-RI-TR-89-15

23. Kim, Y. and S. Desa; "Definition, Determination and Characterization of Acceleration Sets for Spatial Manipulators"; CMU-RI-TR-89-18

24. Kim, Y. and S. Desa; "Acceleration Sets of Planar Manipulators Part I. Theory"; CMU-RI-TR-89-16

25. Kim, Y. and S. Desa; "Acceleration Sets of Planar Manipulators Part II. Applications and Experiments"; CMU-RI-TR-89-17

26. Streit, D.A., C.M. Krousgrill, and A.K. Bajaj; "Nonlinear Response of Flexible Robotic Manipulators Performing Repetitive Tasks"; Trans. ASME 111, pp. 470 - 480 (1989)

27. Nagathan, G. and A.H. Soni; "Nonlinear Modeling of Kinematic and Flexibilty Effects iin Manipulator Design"; Jour. Mech, Trans, and Auto. in Design, 110, pp. 243 - 254 (1988).

28. Calhoub, N. and A. Ulsay; "Dynamic Simulation of a Flexible Robot Arm and Controller"; Proc. Amer. Control Conf., San Diego, 2, pp 631 - 637 (1984)

29. Singh, S.N. and Schy, A.A.; "Robust Torque Control of an Elastic Robotic Arm Based on Invertibility and Feedback Stabilization"; Proc. 24th Conf. on Decision and Control, Ft. Lauderdale, pp. 1317 - 1322 (1985)

30. Kotnik, Paul; "Acceleration Feedback for Control of a Flexible Manipulator Arm", Jour. Robotic Sys. 5, pp 181 - 196, (1988).

31. McInroy, J. E. and G. N. Sardis; "Acceleration and Torque Feedback for Robotic control: Experimental Results", Jour. Robotic Sys. 7, pp 813 - 832 (1990).

APPENDIX - Programs developed for the Amiga 2500 to acquire and process accelerometer data from the Zymzte robot.

```
/**********************************************************************
 * read1c.c                                                          *
 * by David Gilliam  UAH                                  *
 *                                                                   *
 * Reads one channel of the accelerometer and saves the data after     *
 * averaging every group of four values.                             *
 *                                                                   *
 *********************************************************************/

#include <exec/types.h>
#include <stdio.h>
#include <p40.h>
#include <dos.h>

main()
{
        short data[24000];
        short finaldata[6000];
        short channel, success;
        short gain, rate, i;
        long samples;
        float volts = 0.0;
        int fh, error;
        char fname[40],c;
        unsigned int count;

        if((success = p40_init())==0) {
                printf("Couldn't configure the Proto-40k\n");
                exit(0);
        }

        printf("\nInput data file name : ");
        i = scanf("%s",fname);
        printf("\n");
        c = getchar();

        samples = 4096;
        gain = 0;   /* gain of 10 */
        channel = 0;
        rate = 400;  /* in Hz */
```

```
        *p40_modeselect = 0;
        *p40_mux      = (channel << 2);       /* Set A/D channel */
/*      p40_setgain(gain);
        p40_setrate(rate);  */
        printf("\nPress enter to begin taking data\n");
        c = getchar();
        printf("Taking data, please wait...\n");
        if(p40_atod(data,rate,samples,0,gain) == 0)
                printf("Couldn't acquire data\n");
        printf("Data acquisition complete.\n");
        printf("Smoothing data\n\n");

        for(i=0;i<samples;i++)
                finaldata[i] = (data[4*i] + data[4*i+1] + data[4*i+2] + data[4*i+3])/4;

        printf("\nDisplay data? (y/n)");
        c = getchar();
        printf("\n");
        if (c == 'y') {
        printf("Count   Channel %02d\n", channel);
        printf("-----   -----------\n");

        for (i=0; i<samples/4; i++) {
                volts = ((float)finaldata[i] / 204.8) - 10.0;

                printf("%04x %04d %7.4f\n", finaldata[i], finaldata[i], volts);

        }
        } /* end of if */

        printf("\nSaving data to disk\n");

        fh = creat(fname, 0);
        count = write(fh, &finaldata, 12000);
        error = close(fh);

}
/************************************************************************
 * show1c.c                                                             *
 * by David Gilliam   UAH                                               *
 *                                                                      *
 * Displays a 'single channel' data file                                *
 ************************************************************************/

#include <exec/types.h>
#include <stdio.h>
```

```c
#include <p40.h>
#include <dos.h>
#include <fcntl.h>

main()
{
        short data[6000];
        short channel;
        short gain, rate, i;
        long samples;
        float volts = 0.0;
        int fh, error;
        char *fname,c;
        unsigned int count;

        printf("\nInput data file name : ");
        i = scanf("%s",fname);
        printf("\n");
        c = getchar();

        samples = 600;
        gain = 1;   /* gain of 10 */
        channel = 0;
        rate = 100;  /* in Hz */

        printf("\nLoading data from disk\n");
        fh = open(fname, O_RDONLY, 0);
        count = read(fh, &data, 12000);
        error = close(fh);

        for (i=0; i<samples; i++) {
                volts = ((float)((data[i] / 204.8) - 10));

                printf("%04x %04d %7.4f\n", data[i], data[i], volts);

                if(i % 21 == 20) {
                        c = getchar();
                        if (c == 'q' | c == 'Q')
                                i = samples;
                        }
        }
}

/***********************************************************************
 * conv1.c                                                             *
```

```
 * by David Gilliam   UAH                                     *
 *************************************************************************/

#include <exec/types.h>
#include <stdio.h>
#include <p40.h>
#include <dos.h>
#include <fcntl.h>

main()
{
        FILE *fp;
        short data[6000];
        short channel;
        short gain, rate, i;
        long samples;
        float volts = 0.0;
        int fh, error;
        char fname[40],c;
        unsigned int count;

        printf("\nInput data file name : ");
        i = scanf("%s",fname);
        printf("\n");
        c = getchar();

        samples = 1200;
        gain = 1;   /* gain of 10 */
        channel = 0;
        rate = 100;  /* in Hz */

        printf("\nLoading data from disk\n");
        fh = open(fname, O_RDONLY, 0);
        count = read(fh, &data, 12000);
        error = close(fh);

        strcat(fname,".asc");
        fp = fopen(fname, "w");
        if (fp == 0) {
                printf("unable to open destination file\n");
                exit();
        }

/*      printf("Count   Channel  0\n", channel);
        printf("-----   -----------\n"); */
```

```
        fprintf(fp,"\naccelerometer data\n\n");
        for (i=0; i<samples; i++) {
/*              volts = ((float)(data & 0x0fff - 2048)/ 20480.0); */
                volts = ((float)((data[i] / 204.8) - 10));
/*              volts = (volts - 0.6) / 0.6;  */

                fprintf(fp,"%7.4f\n", volts);
        }
        fcloseall();
}
```