N91-20646

PERFORMANCE CAPABILITIES OF A JPL DUAL-ARM ADVANCED TELEOPERATION SYSTEM

Z.F. Szakaly and A.K. Bejczy
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

ABSTRACT

The system comprises a) two PUMA 560 robot arms, each equipped with the latest JPL-developed smart hands which contain 3D force/moment and grasp force sensors, b) two general-purpose force-reflecting hand controllers, c) a NS32016 microprocessors based distributed computing system together with JPL-developed universal motor controllers, d) graphics display of sensor information, e) capabilities for time-delay experiments, and f) automatic data recording capabilities. Several different types of control modes are implemented on this system using different feedback control techniques. This paper describes some of the control modes and the related feedback control techniques, and reports on the achievable control performance for tracking position and force trajectories. The interaction between position and force trajectory tracking is illustrated. The best performance is obtained by using a novel, task-space error feedback technique.

INTRODUCTION

The JPL dual-arm advanced teleoperation hardware system is shown in Figure 1. It employees a novel generalized bilateral force-reflecting control method for manual control of remote manipulators. The novel features of this control method are the following: (1) The master controller is a general purpose Force-Reflecting Hand Controller (FRHC), not a replica of any slave arm. It can be used to control different robot arms through the appropriate kinematic transformations. (2) Force reflection to the operator's hand is referenced to a three-d.o.f. force-torque sensor mounted to the base of the robot hand. (3) The control system is based on distributed computing; it uses two computing nodes for control and information display: one at the control station (FRHC) site and one at the remote robot site.

The system permits a spectrum of operations between full manual, "shared" manual and automatic, and full automatic (called "traded") control, and can be operated with variable active compliance referenced to force-torque sensor. Shared control is implemented by freezing the data output of the master controller (FRHC) in some task space coordinates which are selectable by the operator from a menu. Motion in the frozen task space coordinates can then be controlled by a computer algorithm which can be referenced to force-torque or to some other (e.g., proximity) sensor information.

The overall hardware system, electronic architecture, software system including control modes, control algorithms and the software development system, the real-time graphics (preview and predictive displays) including force-torque sensor data displays, and time-delay simulation capabilities are described in previous publications [1 and 2] which contain further references on other hardware and software details. The "smart hands" attached to the robot arms also represent special features of the JPL dual-arm advanced teleoperation system. The Model B and Model C "smart hands" (shown in Fig. 1) mechanical and electronic details are described in [3].

The purpose of this paper is to describe in detail the currently available control modes and the related feedback control techniques implemented on the JPL dual-arm advanced teleoperation system, and to report on the achievable control performance for tracking position and force trajectories. In the description of performance results, emphasis is given to comparing position and force tracking performance with and without Cartesian servo.

Cartesian (or task-space) servo is a novel feedback technique to correct in the time continuum for position errors. In this technique, task space errors are computed from actual joint space values and actual task space commands. (Eventually, task space errors can be measured directly when such measurement system becomes available.) This novel Cartesian error feedback technique can be applied either to FRHC manual trajectory commands or to trajectory commands generated by an algorithm. Here we formulated and used a novel trajectory generator algorithm. This novel trajectory generator algorithm directly acts on task space position commands without a time-based polynomial decomposition of position commands into joint space or task space trajectories. The velocity (when it is not a constant) and, implicitly, the change of velocity in this novel trajectory generator algorithm follows the profile of harmonic functions. Hence the name: Harmonic Motion Generator (HMG).

First we describe the control modes followed by a discussion of performance data.

CONTROL MODES

The overall data flow diagram of the JPL advanced teleoperation system (for a single arm, for the sake of simplicity) is shown in Fig. 2. It is noted that the computing architecture of this system is a fully synchronized pipeline, where the local servo loops at both the control station and the remote manipulator nodes operate at 1000 Hz rate. The end-to-end bilateral (i.e., force-reflecting) control loop operates at a 200 Hz rate as indicated in the computation system timing diagram, Fig. 3. More on the computational system critical path functions and performance can be found in [4].

The actual data flow depends on the control mode chosen. The different selectable control modes are the following:

- Freeze mode
- Neutral mode
- Current mode
- Joint mode
- Task mode

In Freeze mode the brakes of joints 1,2,3 are locked, the motors are turned off. Joints 4,5,6 are servoed to maintain their last positions. This mode is primarily used when the robot is not needed for a short period of time but turning it off is not desired.

In Neutral mode all position gains are set to 0, gravity compensation is active to prevent the robot from falling down. In this mode the user can manually move the robot to any position and it will stay there.

In Current mode the six motor currents are directly commanded by the data coming in from the fiber optic link. This mode exists for debugging only.

In Joint mode the hand controller axes control individual motors of the robot. The correspondence is set up such that in the most common lower elbow/ inverted wrist configuration the joint mode controls the robot in the naturally expected directions i.e., similar to task mode.

In Task mode the inverse kinematic transformation is performed on the incoming data, the hand controller controls the end effector tip along the three Cartesian and pitch, yaw and roll axes. This mode is the most frequently used for task execution or experiments, and this is the one shown explicitly in Fig. 2.

The format of the data packet transmitted to the robot side is the same in all modes. The header byte defines the mode the robot should be in. This is followed by the six motion command bytes, the grasping force command and a checksum. If the mode byte changes the robot waits until the new mode has been stable for 1000 servo loops or one second. After one second the new mode becomes active.

The data packet coming back from the robot is always formatted the same way independent of what mode the robot is in. The following data is transmitted to the local site:

- Six words of force sensor data
- Grasping force and finger opening
- Robot joint position
- End effector tip Cartesian positions

The control system on the remote site is designed to prevent sudden robot motions. The motion commands received by the fiber optic link are incremental, they are added to the current parameter under control. Sudden large motions are also prevented in case of mode changes. This necessitates proper initialization of the inverse kinematics software at the time of the mode transition. The current Cartesian coordinates from the forward kinematics are input into the inverse one. Besides this the configuration parameters such as upper or lower elbow, normal or inverted wrist have to be correctly initialized.

The data flow diagram shown in Fig. 2 illustrates the organization of several servo loops in the system. The innermost loop is the position control servo of the robot side. This servo uses a PD control algorithm, where the damping is purely a function of the robot joint velocities. The incoming data to this servo is the desired robot trajectory described as a sequence of points at 1 mSec intervals. This joint servo is augmented by the gravity compensation routine to prevent the weight of the robot from causing joint positioning error. Since this servo is a first order one there will be a constant position error that is proportional to the joint velocity.

In basic Cartesian control mode the data from the fiber optic link is integrated first and added to the desired Cartesian position. From this the inverse kinematics generates the desired joint positions. The joint servo moves the robot to this position. From the actual joint position the forward kinematics computes the actual Cartesian positions. The force torque sensor data and the actual positions are fed back to the hand controller side to provide force feedback.

This basic mode can be augmented by the addition of the following:

- Compliance control,
- Cartesian servo,
- Sticktion, friction compensation.

Figure 4 Specifically shows the compliance control and Cartesian servo augmentations.

There are two forms of compliance, integrating and spring type (see Fig. 5). In integrating compliance the velocity of the robot end effector is proportional to the force felt in the corresponding direction. To eliminate drift a dead-band is used. The zero velocity band does not have to be a zero force, a force offset may be used. Such a force offset is used if, for example, we want to push against the task board at a given force while moving along other axes. Any form of compliance can be selected along any axis independently.

In case of the spring type compliance the robot position is proportional to the sensed force. This is similar to a spring centering action. The velocity of the robot motion is limited in both the integrating and spring cases.

There is a wide discrepancy between the robot response bandwidth and the force readings. The forces are read at a 1000 Hz sampling rate although the hand is capable of delivering more than 5000 samples per second. The robot motion command has an output response at a 5 Hz bandwidth. To generate smooth compliance response, the force readings go through two subsequent filters. The first one is a simple averaging of ten force readings. This average is called 100 Hz force and is computed at a 100Hz rate. From this 100 Hz force a 5 Hz force is computed by a first order low pass filter. This 5 Hz force reading is also computed at a 100 Hz rate. The 5 Hz force is used for compliance computations. The subsequent equations define the force filters and the compliance control algorithms.

Force Filter:

Input F_{1000}: Force at 1 KHz

F_{100}: Force at 100 Hz computed as

$$F_{100}(t) = \frac{1}{10}[F_{1000}(t) + F_{1000}(t-1) + \ldots + F_{1000}(t-9)]$$

F_{100} is computed at 100 Hz

F_5: Force at 5 Hz computed as

$$F_5(t) = F_5(t-1) + K_F[F_{100}(t) - F_5(t-1)],$$

$$K_F = \frac{1}{20}$$

F_5 is also computed at 100 Hz

Compliance Control: operates by modifying Cartesian set point X_S

$$X_{S2} = \underbrace{X_{S1} + K_I(F_{5X}-S_{IX})}_{\text{Integrator}} + \underbrace{K_S\{F_{5X}(t)-S_{SX}(t) - [F_{5X}(t-1)-S_{SX}(t-1)]\}}_{\text{Spring}}$$

K_I : integrating constant
K_S : spring constant
X_{S1} : X setpoint coming from hand controller
S_{IX} : X integrating force setpoint
S_{SX} : X spring force setpoint

It is interesting to observe the similarities and differences between averaging and a low pass filter (see also Fig. 6). In order to average them we have to store the ten previous force readings. For the low pass filter a single stored variable is adequate. The step input transfer function of the averaging filter is a linearly increasing output (or more exactly ten equal steps). The same function for the low pass filter is one that exponentially approaches the steady state output value (i.e., the steps become smaller and smaller in time). In terms of filtering, the two have similar effects on the signal, but low pass filtering requires much less memory and computations.

As shown in Fig. 4, the Cartesian servo acts on task space (X,Y,Z, pitch, yaw, roll) errors directly. These errors are the difference between desired and actual task space values. The actual task space values are computed from the forward kinematic transformation of the actual joint positions. This error is then added to the new desired task space values before the inverse kinematic transformation determines the new joint position commands from the new task space commands.

TRAJECTORY GENERATOR

A trajectory generator algorithm was formulated based on observations of profiles of task space trajectories generated by the operators manually through the FRHC. Three important features were observed in hand-generated task space trajectory profiles: (1) The operators always generated trajectories as a function of the relative distance between start point and goal point in the task space or, in general, as a function of the present position state relative to the desired position state of the end effector in the task space. In other words, the operators manually do not generate trajectories based on time (on clock signals). (2) The velocity-position phase diagrams of motion typically resembled a harmonic (sine) function. (3) Between the start and completion phases, the operator-generated trajectories typically attained a constant velocity profile.

Based on these observations, we formulated a Harmonic Motion Generator (HMG) with a sinusoidal velocity - position phase function profile as shown in Fig. 7. The motion is parameterized by the total distance traveled, the maximum velocity, and the distance used for acceleration and deceleration. Both the accelerating and decelerating segments are quarter sine waves, with a constant velocity segment connecting them. This scheme still has a problem, the velocity being 0 before the motion starts. This problem is corrected by adding a small constant to the velocity function.

It is noted that the HMG introduced in this paper is quite different from the typical trajectory generator algorithms employed in robotics which use a polynomial position-time function. Our algorithm generates the motion as a trigonometric (harmonic) velocity versus position function. The position versus time and the corresponding velocity versus time functions generated by the HMG are shown in Fig. 8.

PERFORMANCE RESULTS

Space assembly and servicing tasks are very rich in capability requirements for a dual-arm teleoperation system. For instance, if the Solar Max Repair Mission would have been performed with a dual-arm teleoperation system, the operator(s) of the dual-arm system would have faced the following subtasks: thermal blanket removal, hinge attachment for electrical panel, opening of electrical panel, removal of electrical connectors, relining of cable bundles, replacement of electrical panel, securing parts and cables, replug of electrical connectors, closing of electrical panel, and reinstating thermal blanket. In order to perform all these subtasks, the dual-arm teleoperation system should be endowed with certain generic performance features. Such generic performance features are: move along a straight line and exert a given push force in a given direction (that is, cutting a thermal blanket by knife); hold a given force in a given direction while turning/rolling operation is being performed (that is, removal or reinstatement of panel screws); follow a given path while pulling a flexible object (that is, relining of cable bundles); etc.

Several performance experiments were carried out recently in order to evaluate position and force tracking capabilities of the JPL advanced dual-arm teleoperation system using various control modes and feedback techniques implemented in the system. The subsequent 12 figures (Figs. 10 through 21) show and summarize the performance capabilities. The reference frame in which the motion/force commands are interpreted is shown in Fig. 9.

One-Dimensional Straight Lines

Figures 10 through 12 show performance results of straight one-dimensional (X,Y, or Z) trajectory following, with and without Cartesian servo. The trajectories are commanded from the FRHC at 1 KHz increments, and servoed at the same rate at the remote manipulator. The FRHC task space commands can be true one-dimensional straight lines by inhibiting the computer reading of FRHC motion in the other two orthogonal task space directions. For instance, when commanding a horizontal Y straight line motion, the X and Z directional commands are automatically kept at zero, and servoed accordingly at the remote manipulator. That is, a one-dimensional straight line command is independent of the operator's ability to move the FRHC on a straight line. This capability is automatically guaranteed by the command/control software.

It is clear from Figs. 10 to 12 that Cartesian servo gives a superior and very satisfactory trajectory following performance over the non-Cartesian (that is, pure joint servo) performance. Indeed, it compensates very well for sticktion, friction, and for some level of uncertainties in gravity loading. It is noted that the remote manipulator was operated with about 80% gravity compensation control only and without sticktion and friction compensation.

Two-Dimensional Straight Lines

Figures 13 through 15 show performance results of two-dimensional (X-Z, Z-Y, Y-X) straight line trajectory following tasks, with and without Cartesian servo. Again, the trajectories were commanded from the FRHC at 1 KHz increments, and servoed at the same rate at the remote manipulator. It is noted that the quality of a straight line trajectory in a plane depends on the operator's ability to generate a true straight line with his hand motion in that plane. It is automatically guaranteed, however, that the trajectory command will be in the selected plane by inhibiting the computer reading of any FRHC motion perpendicular to the selected plane.

Again, it is clear from Figs. 13 to 15 that Cartesian servo yields a superior and very satisfactory trajectory following performance over the non-Cartesian (pure joint servo) performance. It compensates very well for sticktion, friction and uncertainties in gravity loading.

One-Dimensional Straight Lines With Force Command

Figures 16 through 19 show performance results for tasks of one-dimensional straight line trajectory following with the added requirement of maintaining a given force in a given direction along the straight line trajectory. Force control was automatic by selecting the "integrator" component of the compliance control algorithm (see Fig. 6 and the corresponding equations in the text) along the appropriate direction and at the appropriate level.

It is clear from Figs. 16 and 17 that Cartesian position servo considerably improves trajectory position following performance along the commanded motion direction. It is not clear, however, what is the role of Cartesian position servo along the commanded force-maintaining direction referenced to force sensor data. Theoretically, the two control loops contradict each other. In the actual performance, however, the "integrator-compliance" loop seemingly overrules the Cartesian position servo loop along the compliance axis. In any case, automatic compliance control shown very satisfactory performance within the mechanical limits (backlash, hysteresis, etc.) of the PUMA 560 manipulator.

For future applications it is recommended to disable Cartesian position servo along the commanded compliance axis and keep Cartesian position servo only acting along the axes where no force compliance is required.

Figures 18 and 19 also clearly show the output profiles of the 100 Hz and 5 Hz force-torque sensor data filters described previously and applied in the compliance control algorithm. The actual mechanical response profile of the manipulator's compliant interaction with the environment is along the 5 Hz filter trajectory.

Harmonic Motion Generator (HMG) Trajectories

Two examples are quoted here. Figure 20 illustrates the same trajectory following example which was shown in Fig. 10. There, the trajectory was generated by FRHC motion. Here, it is generated by the HMG outlined previously. Again, Cartesian position servo provides a much better trajectory following performance than the pure joint servo.

Figure 21 illustrates the same trajectory as shown above in Fig. 20 as generated by the HMG algorithm, with the additional requirement of maintaining a given force level in X direction along the Y-directional trajectory. For maintaining force, the integrator part of the automatic compliance algorithm was used. Cartesian servo was disabled along the compliance axis (X) but was retained along the other two (Y and Z) orthogonal axes. To make the task more challenging, the task board along the Y direction was disoriented by about 5 degrees relative to the nominal Y direction. That is, to maintain a constant force along the X direction while moving in the Y direction required an automatic position correction in the X direction based upon force sensing. As seen in Fig. 21, the automatic control system performed excellently.

It is noted that the example shown in Fig. 21 is equivalent to cutting a 40 cm long material with a knife with 5N cutting force automatically, and such that misalignment between cutting board and knife along the cut direction is automatically corrected based on the sensing of the required cutting force.

CONCLUSIONS

The quoted examples have shown the performance utility of (a) Cartesian position servo in trajectory following tasks and (b) automatic compliance in force following/maintaining tasks. Comparing Fig. 21 to Fig. 19, one can also conclude that for certain well-defined tasks (e.g., cutting a material), an automatic HMG combined with an automatic compliance control can give smoother results than an FRHC generated trajectory combined with automatic compliance control.

Future plans include the expansion of the quoted control capabilities formalized into easy operator menus. The capabilities will then be exercised on Solar Max Repair Mission (SMRM) tasks in realistic mission simulation settings in order to demonstrate existing and missing (or, to be improved) capabilities for space applications.

REFERENCES

[1] Bejczy, A.K., Szakaly, Z., Kim, W.S., "A Laboratory Breadboard System for Dual-Arm Teleoperation," Proc. of Third Annual Workshop on Space Operations, Automation and Robotics, JSC, Houston, TX, July 25-27, 1989, NASA Conf. Publication 3059, pp. 649-660.

[2] Szakaly, Z., Kim, W.S., Bejczy, A.K., "Force-Reflective Teleoperated System with Shared and Compliant Control Capabilities," Proc. of the NASA Conf. on Space Telerobotics, Jan. 31, 1989, JPL Publication 89-7, Vol. IV, pp. 145-155.

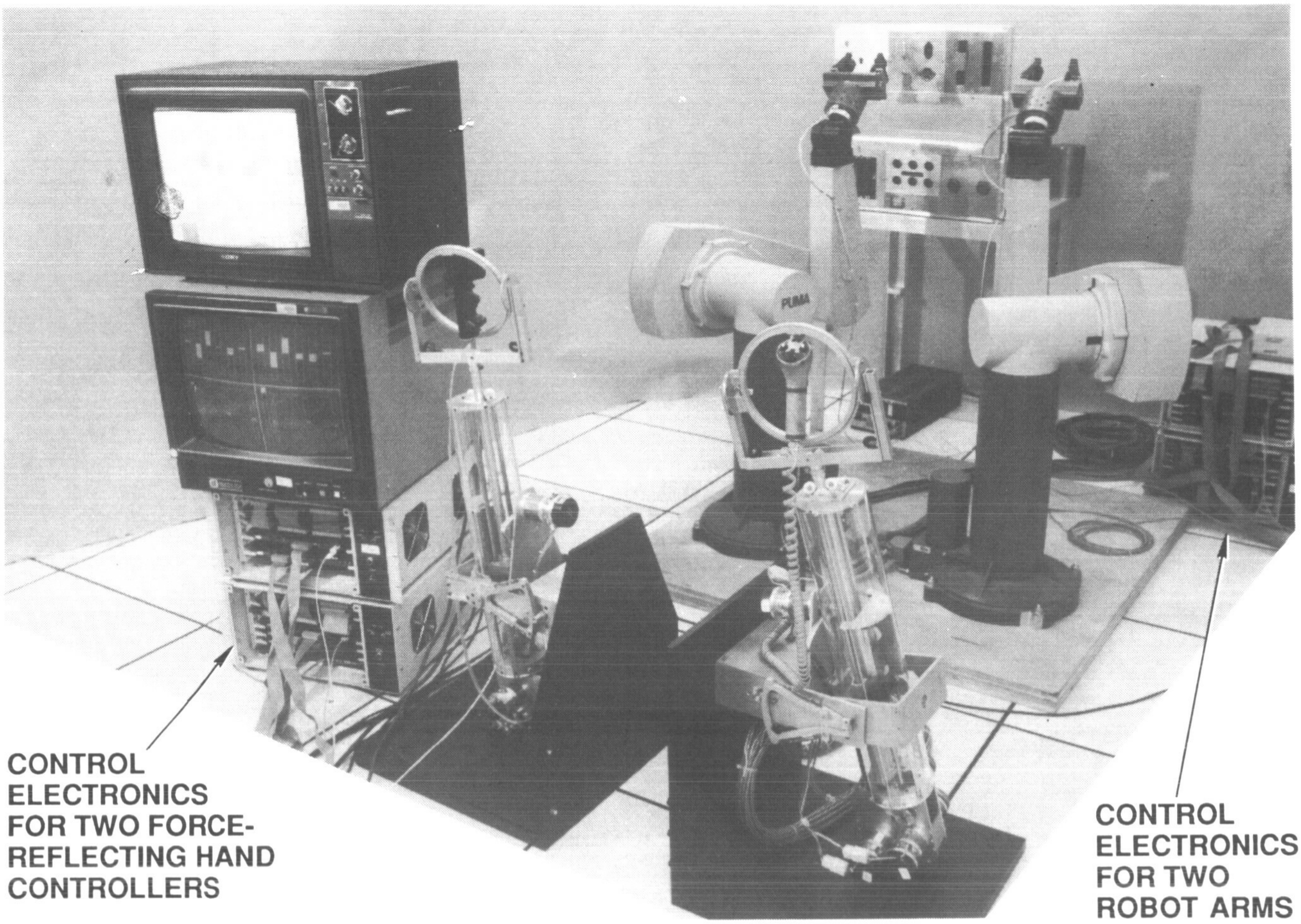
[3] Bejczy, A.K., Szakaly, Z., Ohm, T., "Impact of End Effector Technology on Telemanipulation Performance," Proc. of Third Annual Workshop on Space Operations, Automation and Robotics, JSC, Houston, TX July 25-27, 1989, NASA Conf. Publication 3059, pp. 429-440.

[4] Szakaly, Z., Fleischer, G., "JPL Advanced Teleoperation Control System Critical Path Performance," JPL Memo 3470-90-332.

ACKNOWLEGEMENTS

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract to the National Aeronautics and Space Administration.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH



CONTROL ELECTRONICS FOR TWO FORCE-REFLECTING HAND CONTROLLERS

CONTROL ELECTRONICS FOR TWO ROBOT ARMS

Figure 1. JPL Dual-Arm Advanced Teleoperation System

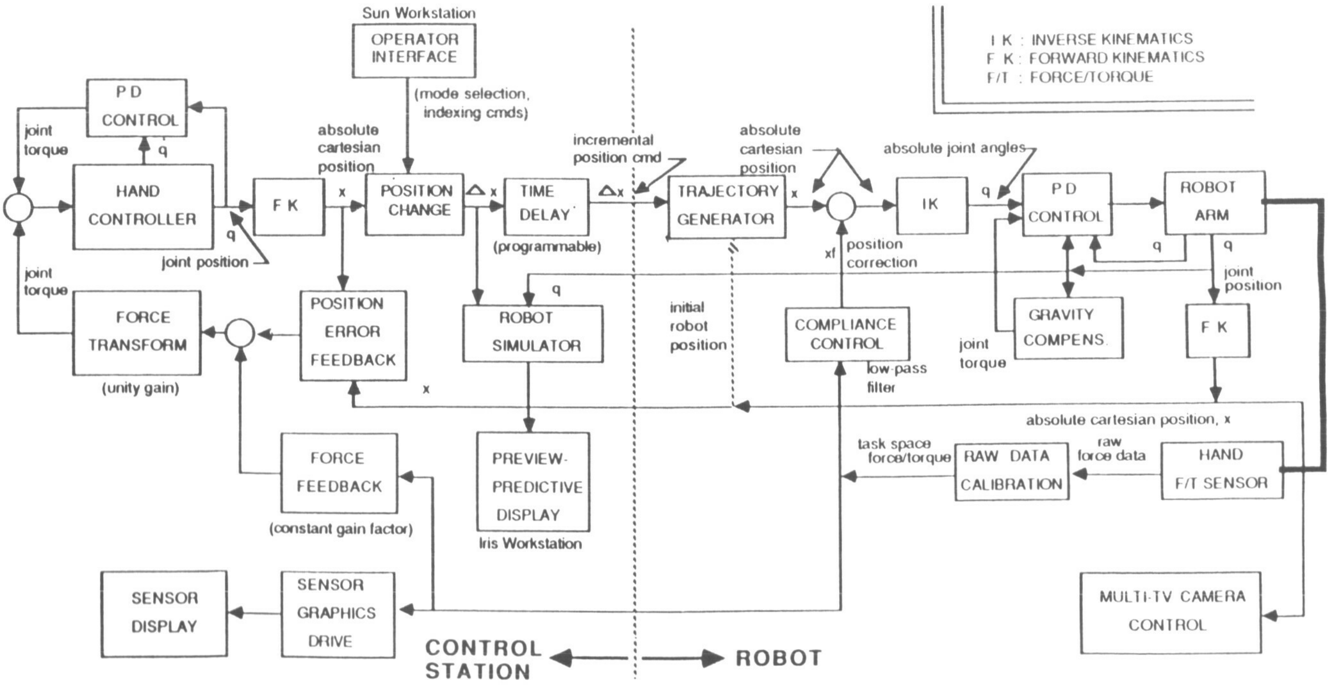


Figure 2. System Flow Diagram

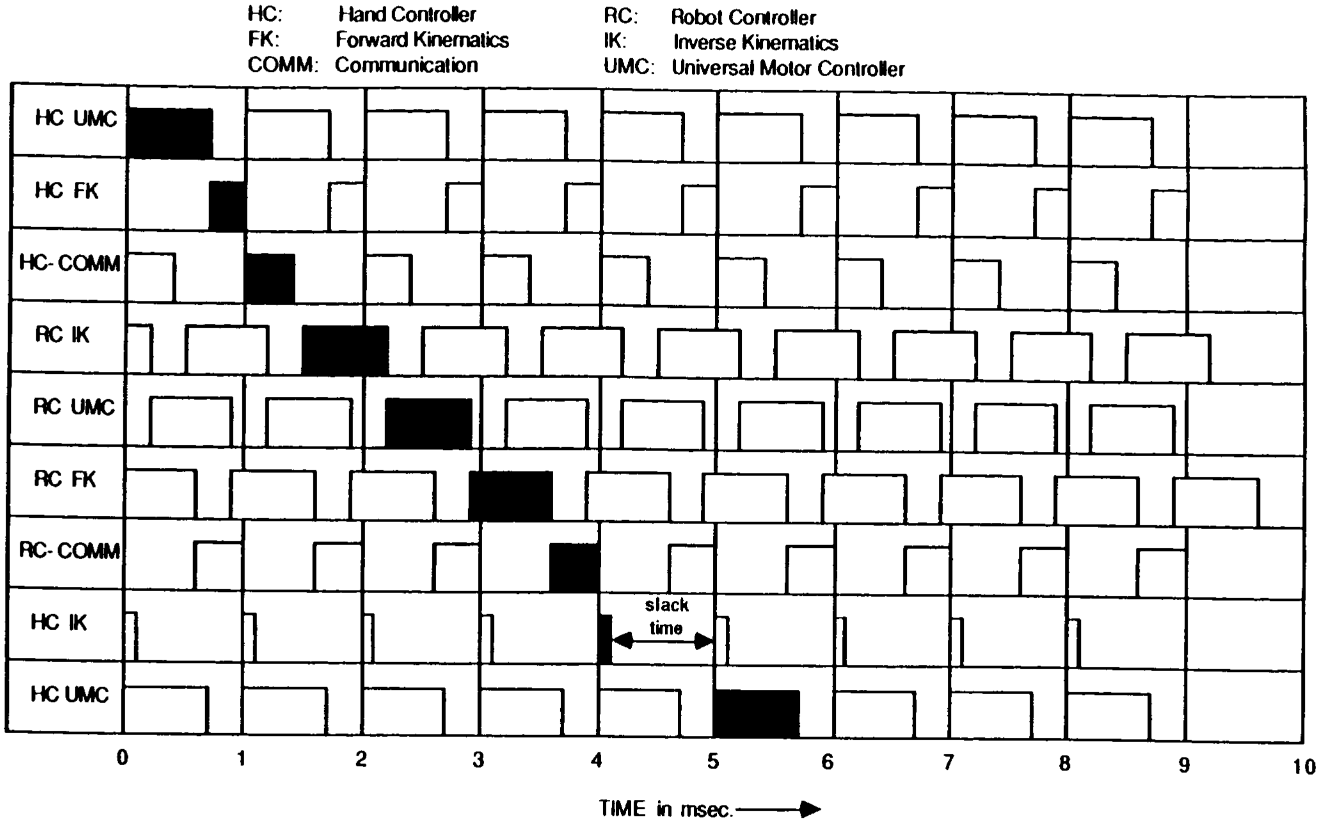


Figure 3. System Timing Diagram

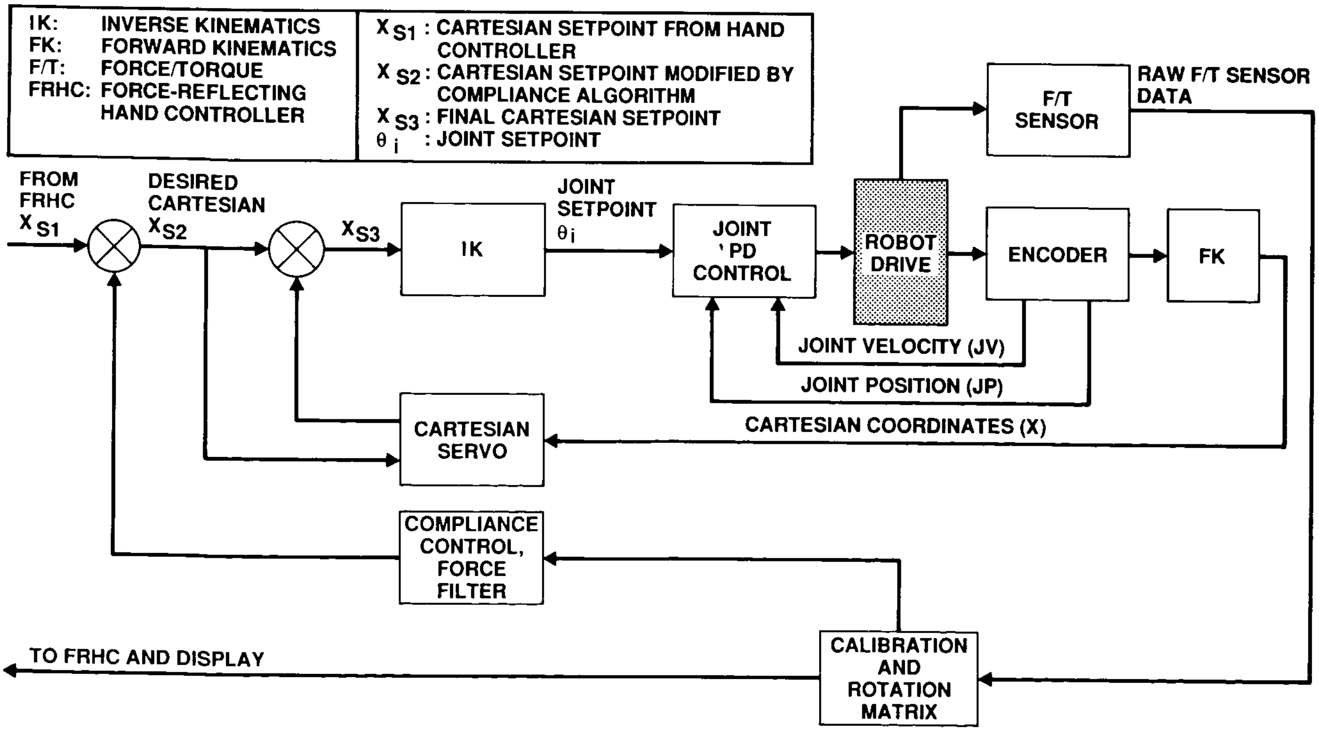

IK: INVERSE KINEMATICS
FK: FORWARD KINEMATICS
F/T: FORCE/TORQUE
FRHC: FORCE-REFLECTING HAND CONTROLLER
X_{S1} : CARTESIAN SETPOINT FROM HAND CONTROLLER
X_{S2} : CARTESIAN SETPOINT MODIFIED BY COMPLIANCE ALGORITHM
X_{S3} : FINAL CARTESIAN SETPOINT
θ_i : JOINT SETPOINT
FROM FRHC X_{S1}
DESIRED CARTESIAN X_{S2}
X_{S3}
IK
JOINT SETPOINT θ_i
JOINT ' PD CONTROL
ROBOT DRIVE
F/T SENSOR
RAW F/T SENSOR DATA
ENCODER
FK
JOINT VELOCITY (JV)
JOINT POSITION (JP)
CARTESIAN SERVO
CARTESIAN COORDINATES (X)
COMPLIANCE CONTROL, FORCE FILTER
TO FRHC AND DISPLAY
CALIBRATION AND ROTATION MATRIX

Figure 4. Control Schemes: Joint Servo, Cartesian Servo, Compliance Control

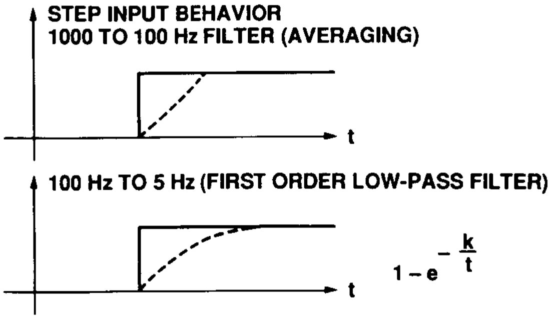


Figure 5. Force-Torque Sensor Data Filters

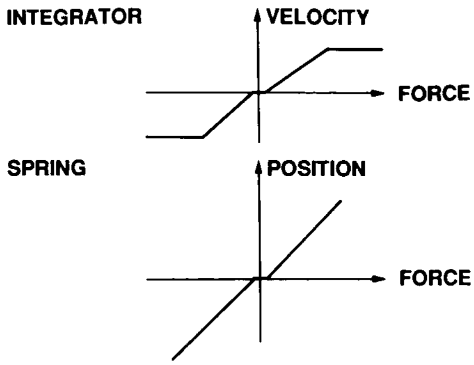


Figure 6. Compliance Components and Interpretations

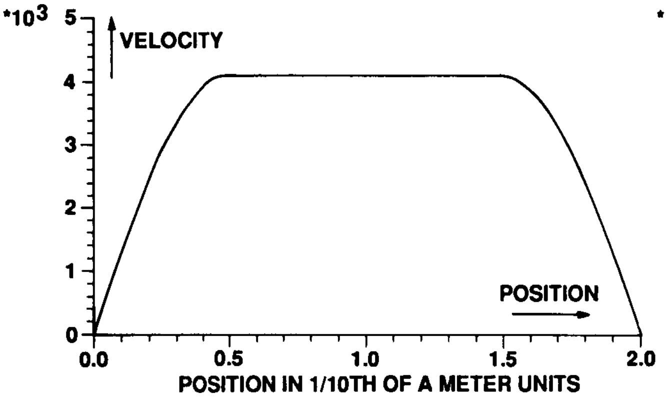


Figure 7. Harmonic Motion Generator Velocity-Position Function

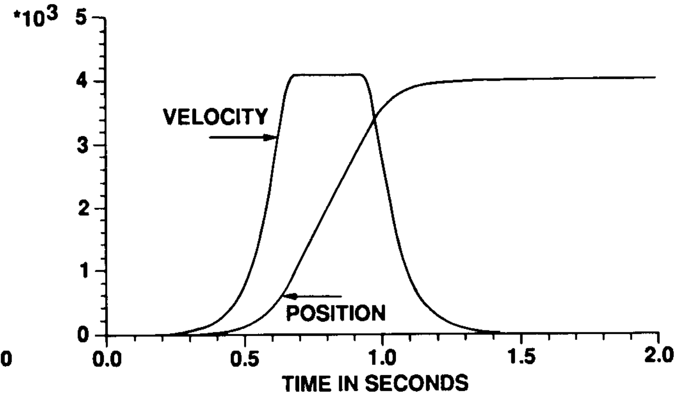


Figure 8. Harmonic Motion Generator Position and Velocity Time Functions

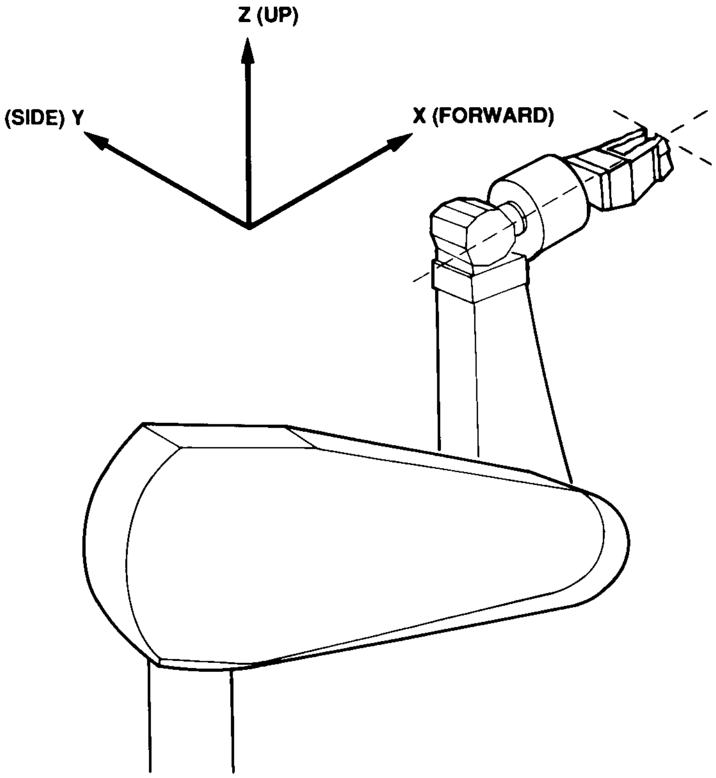


Figure 9. Reference Frame

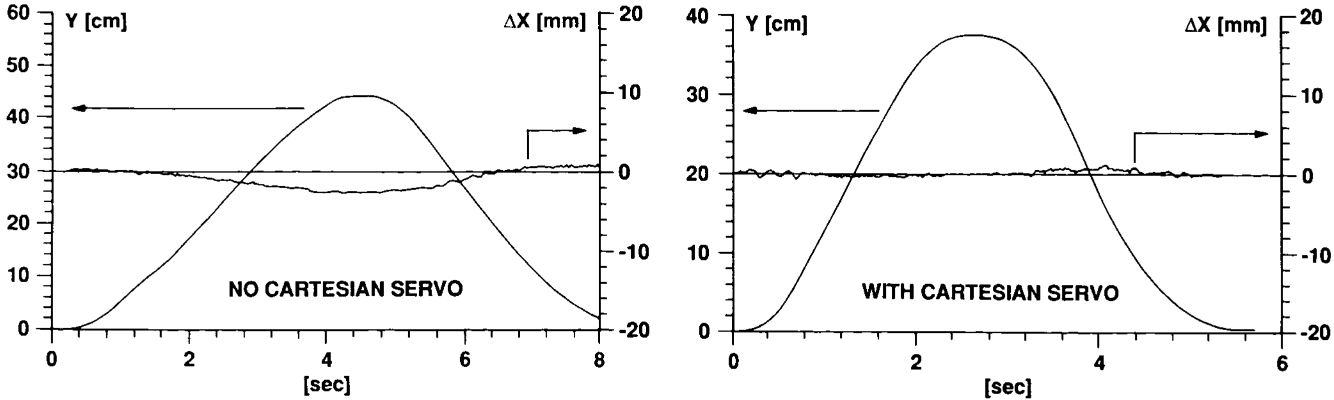


Figure 10. Horizontal (Y) Straight Line Trajectory and ΔX Error.

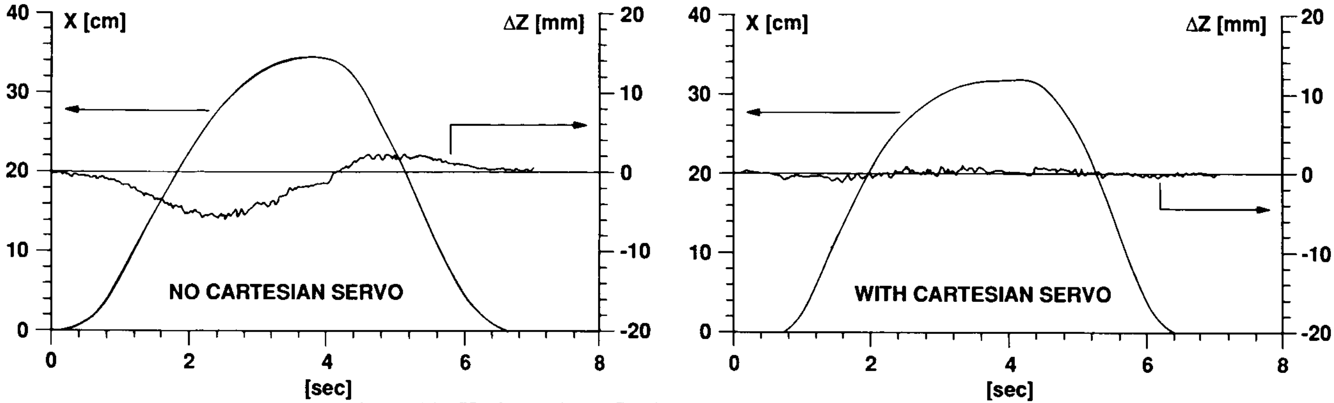


Figure 11. Horizontal (X) Straight Line Trajectory and ΔZ Error.

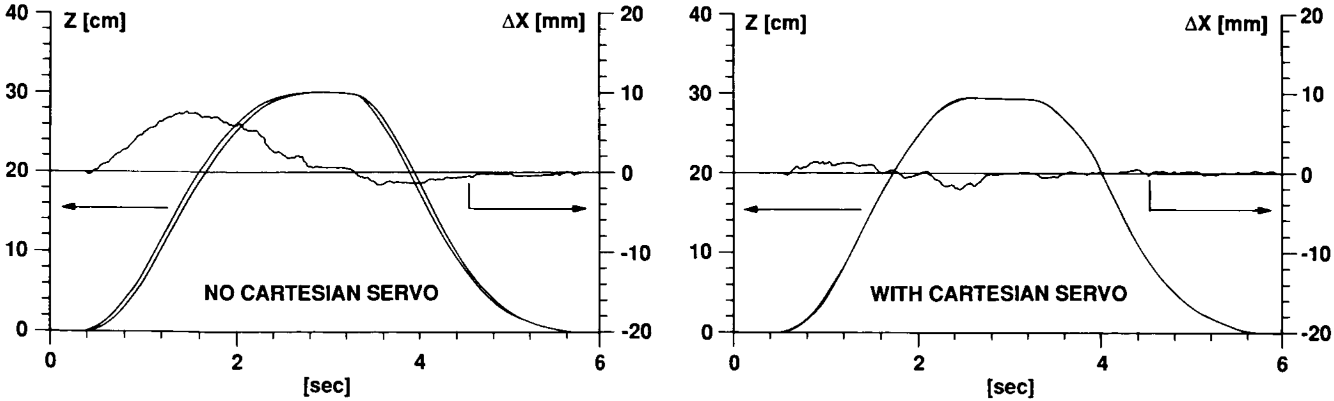


Figure 12. Vertical (Z) Straight Line Trajectory and ΔX Error.

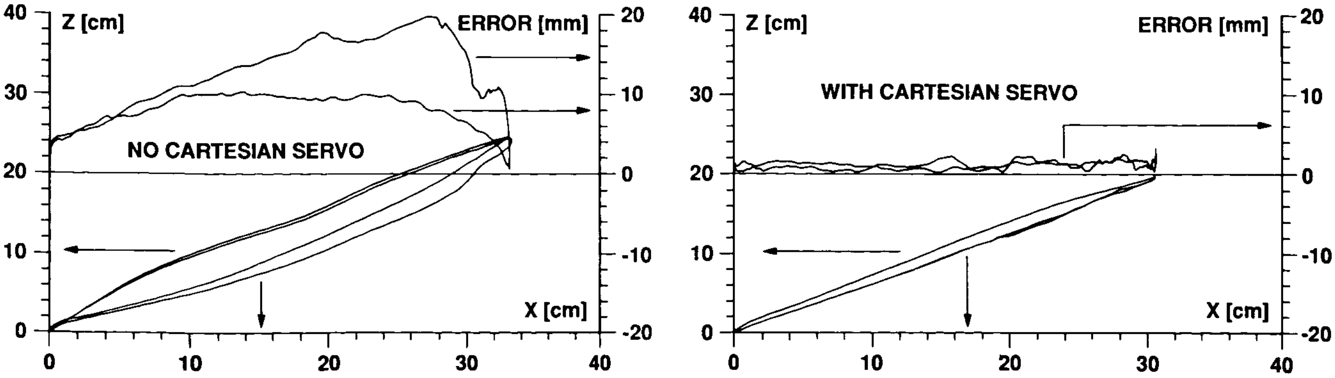


Figure 13. X-Z Plane Forward-Up/Backward-Down Trajectory and Absolute Error in X-Z Phase Plane.

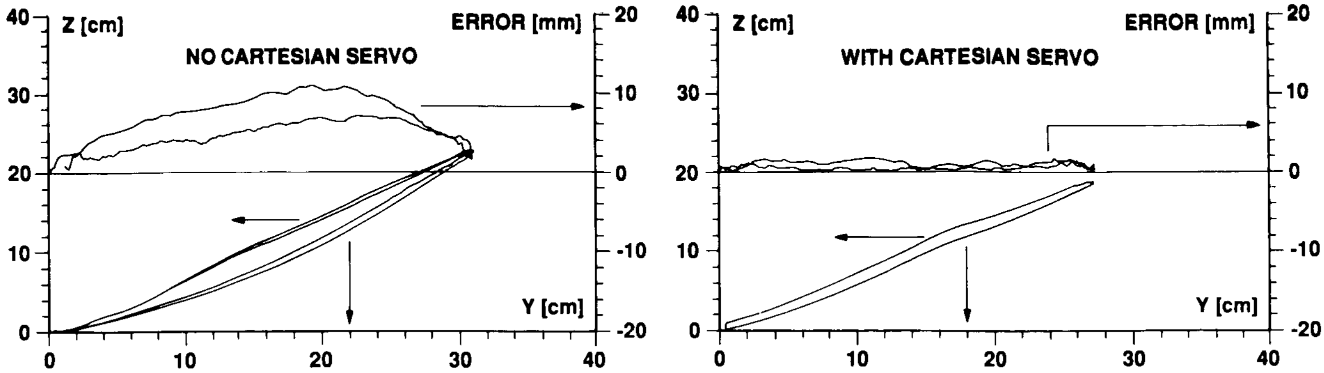


Figure 14. Y-Z Plane Right-Up/Left-Down Trajectory and Absolute Error in Y-Z Plane.

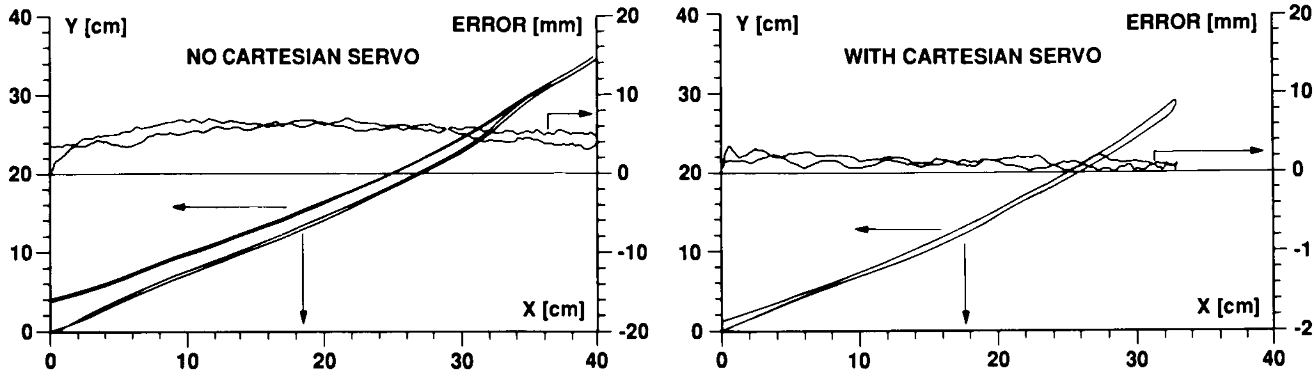


Figure 15. X-Y Plane Forward-Right/Backward-Left Trajectory and Absolute Error in X-Y Phase Plane.

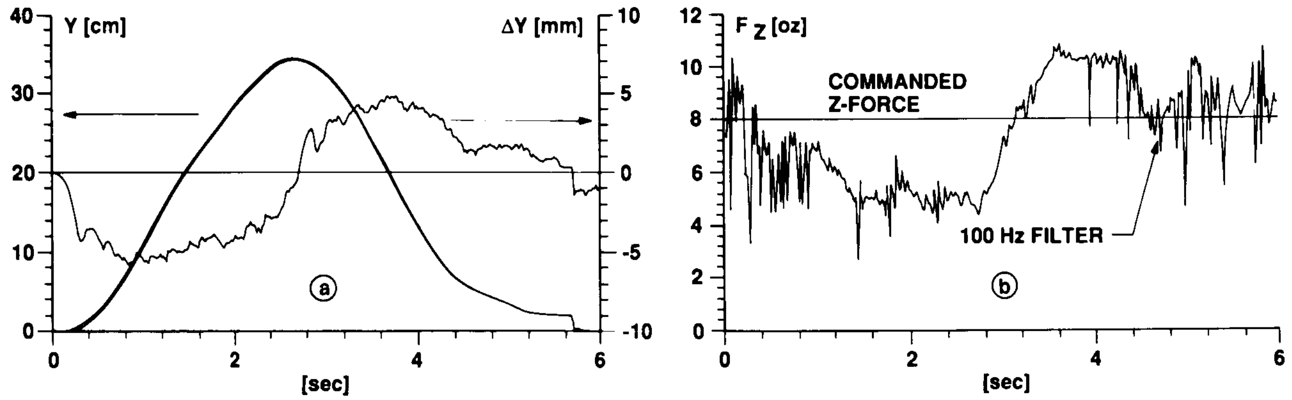


Figure 16. Horizontal (Y) Straight Line Trajectory, With Constant Z-Force Command, Without Cartesian Servo: ⓐ Position Tracking, ⓑ Force Tracking.

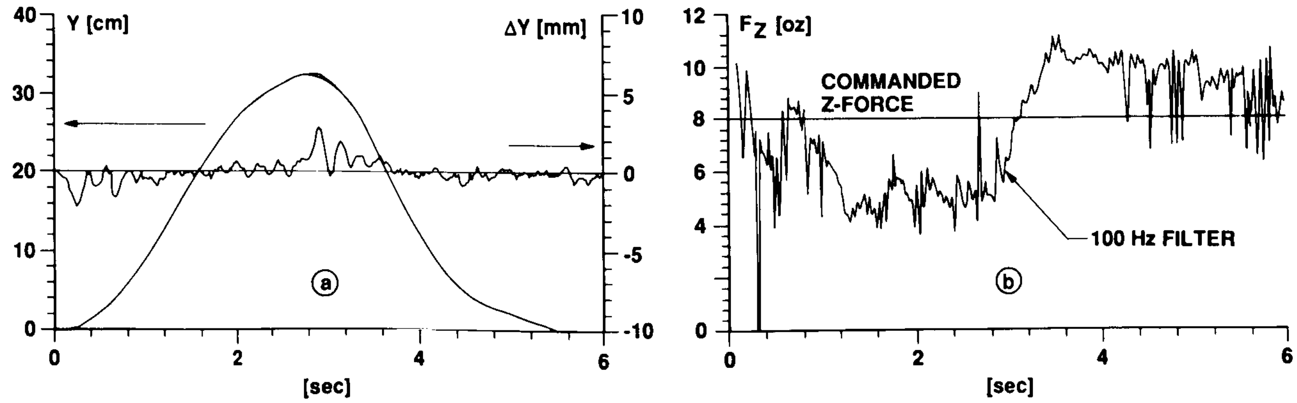


Figure 17. Content the Same as Fig.16, But With Cartesian Servo.

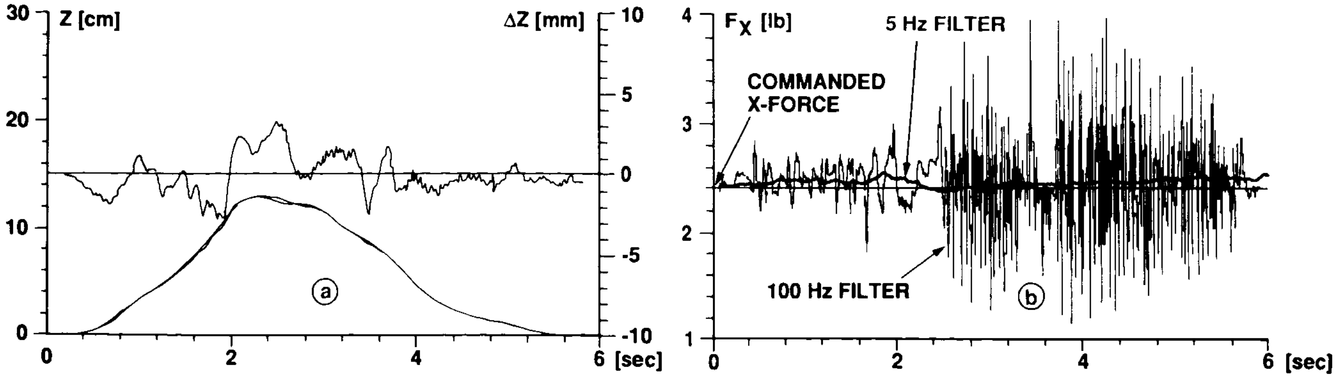


Figure 18. Vertical (Z) Straight Line Trajectory, With Constant X-Force Command, With Cartesian Servo: ⓐ Position Tracking, ⓑ Force Tracking.

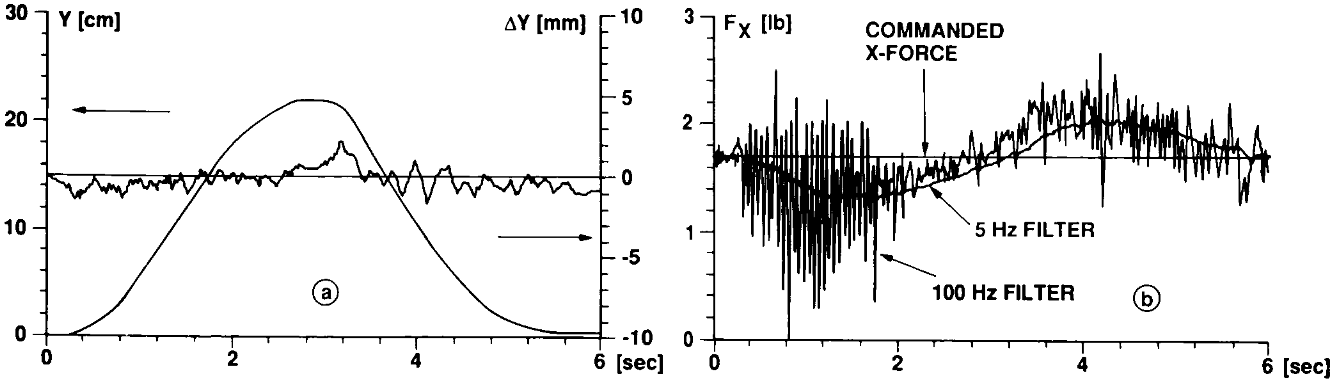


Figure 19. Horizontal (Y) Straight Line Trajecotry, With Constant X-Force Comand, With Cartesian Servo: ⓐ Position Tracking, ⓑ Force Tracking.

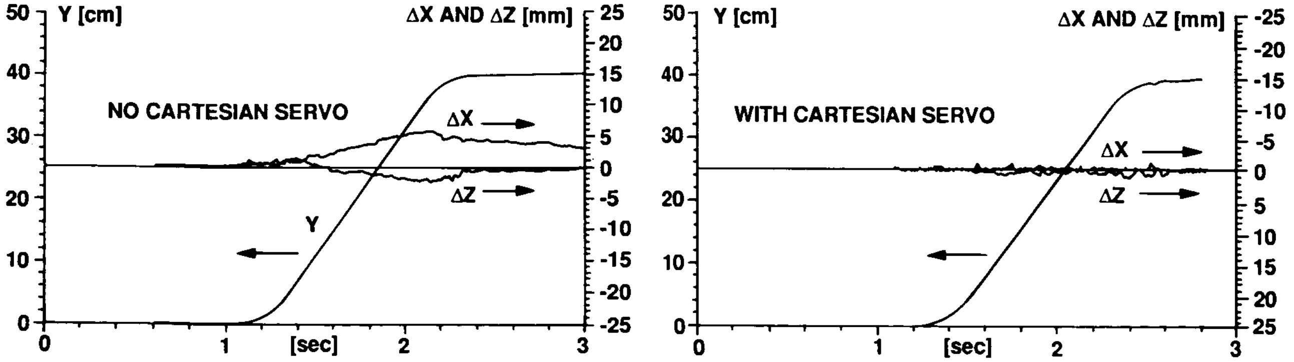


Figure 20. Horizontal (V) Straight Line Trajectory from Harmonic Motion Genrator and ΔX 2 ΔZ Errors.

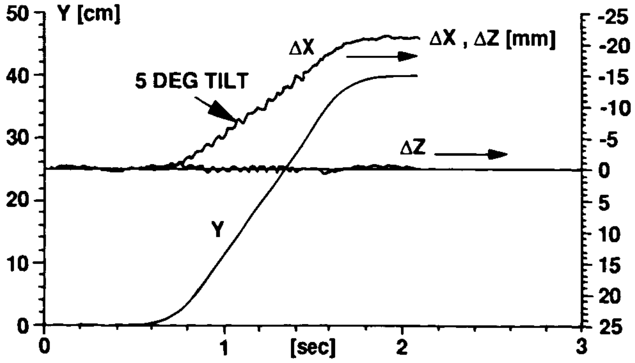


Figure 21. Horizontal (Y) Straight Line Trajectory With Constant X-Force Command and With Cartesian Servo. (Task Board Tilted by 5 Degrees Relative to the Nominal Horizontal Straight Line.)