An Active Vertical-Direction Gravity Compensation System

Gregory C. White and Yangsheng Xu

Abstract—**To perform simulations of partial or microgravity environments on earth requires some method of compensation for the earth's gravitational field. This paper discusses an active compensation system that modulates the tension in a counterweight support cable in order to minimize state deviation between the compensated body and the ideal weightless body. The system effectively compensates for inertial effects of the counterweight mass, viscous damping of all pulleys, and static friction in all parts of the gravity compensation (GC) system using a hybrid PI (proportional plus integral)/fuzzy control algorithm. The dynamic compensation of inertia and viscous damping is performed by PI control, while static friction compensation is performed by the fuzzy system. The system provides a very precise gravity compensation force, and is capable of nonconstant gravity force compensation in the case that the payload mass is not constant.**

I. Gravity Compensation for Space Robotics Experiments

TO conduct earthbound research on robotic applications for outer space environments, compensation for the gravitational field must be considered. Several schemes have been employed to compensate for the earth's gravity, including passive counterweight, active motor control [1], and underwater experiments. All three have advantages and disadvantages. The passive counterweight is simple, but introduces additional inertia and friction. The active motor control does not add inertia, but can be dangerous in the event of power failure. The underwater approach introduces effects of fluid dynamics and complexity of underwater operation but can be used with any number of objects for complex simulations.

In the Robotics Institute at CMU we have developed a passive counterweight gravity compensation (GC) system [2] in which the robot is supported by a lightweight cable which passes through a mobile support point and several pulleys before terminating on a counterweight that has an effective mass equal to the mass of the robot. The mobile support point is actively positioned directly over the moving robot by a servo-controlled boom and carriage system (Fig. 1). When the system is at rest, the support cable applies an upward force that exactly compensates for the force of gravity. This is the essence of gravity compensation as it is currently implemented.

For accurate, reliable, fast tracking of the robot motion, and complete compensation of gravity force, we need to improve the following weaknesses:

Manuscript received October 2, 1992; revised April 20, 1994.

The authors are with the The Robotics Institute, Carnegie, Mellon University, Pittsburgh, PA 15213 USA.

IEEE Log Number 9406104.

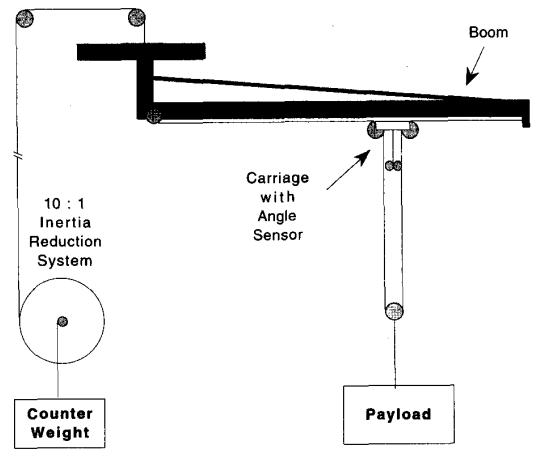

Fig. 1. CMU gravity compensation system.

1) The passive counterweight adds additional inertial mass to the system resulting in an increase in the apparent mass of the robot, and reducing the robot system bandwidth in the vertical direction. This increase in mass is kept to 10% by an inertia reduction mechanism, but nonetheless is still a troublesome nonideality of the system.
2) The inherent friction of the pulley system is considerable because of the required complexity of the cable routing. The cable passes over nine individual pulleys before reaching the counterweight. Static friction has proven to be the greatest liability in the current design. Since the robot manipulator is a lightweight, flexible, low-power system, precise end-effector positioning is hampered considerably by the effects of static friction.
3) The mass of the counterweight is currently set by trial and error; weights are added or removed until the payload "feels" as if it is effectively gravity compensated. This can result in a residual gravitational force being experienced by the payload, which is dependent on the accuracy of the counterweight mass.

We propose to actively control the vertical direction of the current GC system. This system will be a hybrid of the passive counterweight and active motor control schemes. A strain gauge will be used to sense cable tension, and an encoder will sense payload position. A DC servo motor can be used to drive one of the pulleys, thereby modulating the tension in the support cable. By doing so, we aim to minimize the effects of

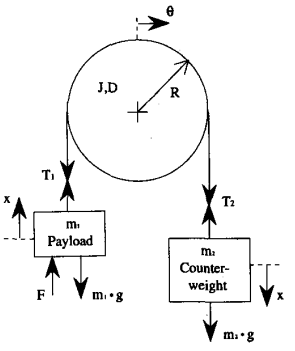

Fig. 2. Simplified system.

TABLE I
NEWTONIAN DYNAMICALLY DIFFERENTIAL EQUATIONS

For m_1:	$T_1 = m_1(g + \ddot{x}) - F$	
For m_2:	$T_2 = m_2(g - \ddot{x})$	
For pulley:	$J\ddot{\theta} + D\dot{\theta} = \Gamma_{ms} + R(T_2 - T_1)$	
where:		
T_1 = robot cable tension		m_1 = robot mass
T_2 = counterweight cable tension		m_2 = counterweight mass
x = vertical position		J = inertial moment of pulley
θ = angular position		D = viscous damping of pulley
F = externally applied force		R = radius of pulley
Γ_{ms} = effective motor torque		g = acceleration of gravity

friction and counterweight inertia, and completely compensate for gravitational force. The advantages of this approach are that it can overcome the three problems outlined above, while retaining the safety of the counterweight mass. In addition, the system can be designed to allow partial gravitational compensation, thus making the system easy to use for partial gravity simulations (e.g., lunar, or nonearth planetary).

II. Modeling of System Dynamics

The system can be simplified to obtain approximations of the model structure. This simpler system consists of a single pulley over which a rigid cable supports two equal masses on either side as is shown in Fig. 2. The inertial moment of the model pulley is equal to the sum of the inertial moments of the individual pulleys in the real system.

The following assumptions were made to reduce the model order to a more manageable degree: the mass of the cable is negligible compared to the mass of the payload and counterweight; the longitudinal compliance of the cable contributes insignificantly to the response of the system; there is no slippage between the cable and pulleys; the electrical time constant of the drive motor is considerably smaller than its physical time constant. Leaving out these higher order effects greatly reduces the model order and makes control design easier. Also, some of the high-order dynamics are dependent upon the lengths of the cables, and change constantly as the payload moves vertically. To attempt to control these effects would be difficult and will be discussed later.

The fundamental dynamics of the simplified system can be expressed by three Newtonian differential equations, one

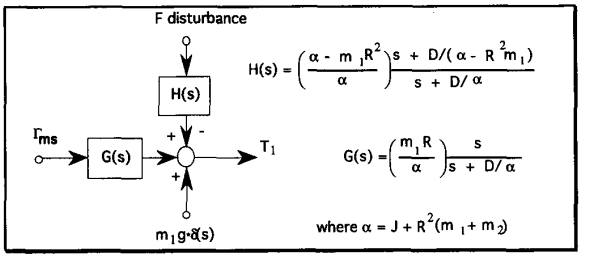

Fig. 3. Block diagram and transfer functions.

for each body from Fig. 2. The equations and parameters are shown in Table I.

If we assume zero compliance in the cables, $x = R\theta$. In this system T_1 is the parameter to be controlled, and can also be sensed directly using a strain gauge. We will assume that the system is modeled well enough that T_1 will give us a good indication of state, and we will sense θ only for friction compensation (see Section V).

Combining these three equations results in the system diagram shown in Fig. 3.

III. Controller Design and Simulations

At any point in time, the center of mass of a body in linear motion can be described by its position, velocity, and acceleration, which can be represented as a state vector $\underline{x} = [\ddot{x} \quad \dot{x} \quad x]^T$. A body in zero gravity will have an acceleration equal to the outside force applied to the body divided by the mass of the body (F_o/m). Assuming we use an inertial frame of reference so that the initial position and velocity are zero, the resulting state vector is

$$\underline{x} = \frac{1}{m}\left[F_0 \quad \int F_0 dt \quad \iint F_0 dt^2\right]^T.$$

If the tension in the support cable of the gravity compensation system deviates from the nominal value of $m_1 g$ by δT, the state of the compensated body will deviate from the state of the ideal zero-gravity body by

$$\underline{\delta x} = \frac{1}{m}\left[\delta T \quad \int \delta T dt \quad \iint \delta T dt^2\right]^T.$$

To maintain exact state tracking, all three of these terms must equal zero. This is, of course, impossible to accomplish with finite bandwidth control. The only way to reconcile this is to admit some state deviation, but to attempt to minimize it as much as possible. This, then, is the prime objective: to minimize the dynamic state deviation of the compensated mass with respect to an equivalent mass in a zero (or partial) gravity environment.

Since all disturbances will be of finite duration, and most will be applied over a fairly short time frame (~1 s), the most straightforward approach is to use a type I (PI) controller, which in this system configuration, will completely reject impulse-type disturbances, and track step-type disturbances with a finite steady-state error.

In order to design the discrete-time (DT) controller, it is necessary to discretize the system model so that the DT root

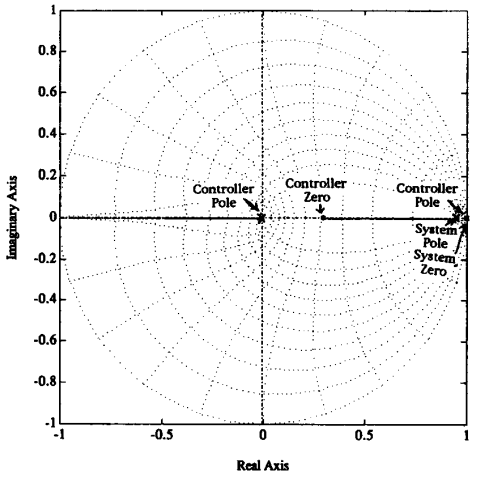

Fig. 4. Z-plane root locus versus K of $G(z)C(z)$.

loci can be drawn. Using the matched pole-zero approach, the DT transfer functions for the system shown above are

$$G(z) = \frac{m_1 R}{\alpha}\left(\frac{z-1}{z-e^{-\mathrm{DT}/\alpha}}\right)$$
$$H(z) = -\frac{1}{\alpha}\left(\frac{z-e^{-\mathrm{DT}/\alpha}(\alpha - m_1 R^2)}{z-e^{-\mathrm{DT}/\alpha}}\right).$$

There is a delay of one time period between sensing and actuation; therefore an additional controller pole at the origin is unavoidable. The DT PI controller transfer function and time series are then

$$C(z) = \frac{\Gamma}{T_1} = K\left(\frac{z-b}{z(z-1)}\right)$$
$$\Gamma[n] = K(T_1[n-1] - b \cdot T_1[n-2]) + \Gamma[n-1],$$

using the notation $x[n] = x(nT)$.

The DT root locus versus K of $G(z)C(z)$ for a chosen b is shown in Fig. 4.

To simulate the operation of this system in the continuous-time domain, it is necessary to convert the digital controller into continuous time, assuming a zero-order hold for output, and using the (2, 2) Padé approximant [5] to approximate the finite time delay (one sampling period) between sensing and actuation. This approximation is valid to within $5 \cdot 10^{-3}\%$ for $s < 0.5/T$:

$$C(s) = \left(\frac{12T}{T^2s^2 + 6Ts + 12}\right)C(z);$$

where

$$z = e^{sT} = \frac{T^2s^2 + 6Ts + 12}{T^2s^2 - 6Ts + 12}.$$

The time responses for the uncontrolled and controlled systems, due to a step disturbance of magnitude 1 N, has been simulated on computer, and are shown in Figs. 5 and 6. Notice that the uncontrolled system exhibits a rather large steady-state error to a step disturbance, while this is completely eliminated in the controlled system.

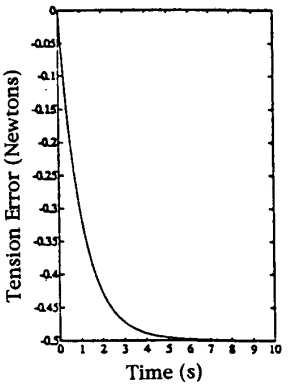

Fig. 5. Uncontrolled step disturbance rejection.

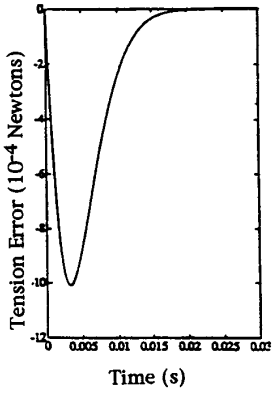

Fig. 6. Controlled step disturbance rejection.

TABLE II
EXPERIMENTAL SYSTEM PARAMETERS

Motor Parameters	System Parameters
$K_T = 52 \cdot 10^{-3}$ Nm/A	R = 0.0095 m
$I_{static\ friction} = 0.1$ A	$m_1 = 0.2615$ kg
$J = 65 \cdot 10^{-6}$ Nms^2	$m_2 = 0.3272$ kg
$D = 6.5 \cdot 10^{-6}$ Nms	servo gain = 1.482 A/V
Strain gauge output Y = (1.06 V/N) · T_1 + 0.438 V	

IV. EXPERIMENTAL SETUP

An experimental model of the system has been built, using the system in Fig. 2 as a basis. It consists of the following:

Hitachi FCL "pancake" motor with quadrature encoder;
Omega LCL-010 strain gauge;
National Semiconductor LH0036 instrumentation amplifier;
Two masses (262 g, 327 g);
Lightweight nylon string;
Personal computer with i80286 CPU, and i80287 FPU;
Data Translation, DT2801 12-b A/D converter;
Technology 80, Model 5312 Quadrature decoder;
PMI Torque servo drive, type WXA 48-B-16.

Unfortunately the parameters for the motor were unavailable, so experimental estimates were made. The other system parameters can be easily measured for the experimental system, and are shown in Table II.

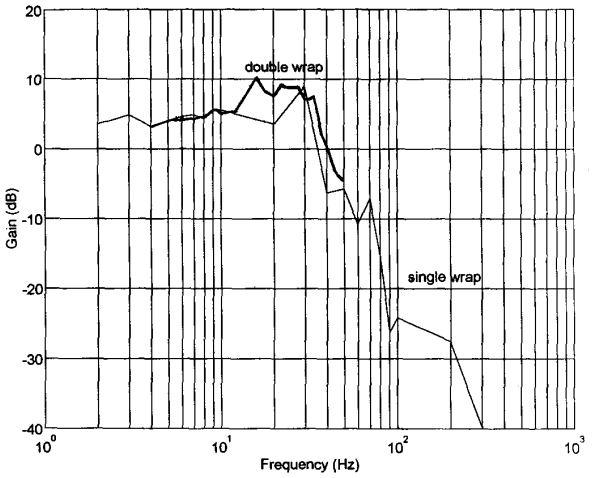

Fig. 7. Bode plot of $G(s)$.

The system transfer functions are

$$G(s) = \frac{21s}{s + 0.055} \quad H(s) = (0.8)\frac{s + 0.0688}{s + 0.055}.$$

The controller transfer function is

$$C(z) = (K/77 \cdot 10^{-3})\frac{z - b}{z(z - 1)}$$

where K and b have been chosen using root locus techniques to be $K = 21.225$, and $b = 0.83$. The time series that results from this transfer function is: $\Gamma[n] = 260 \cdot Y[n-1] - 216 \cdot Y[n-2] + \Gamma[n-1] - 19.41$, where $Y[n]$ is the sampled strain gauge output at time nT.

It was found experimentally that the real system exhibited instability for this selection of gains, suggesting that the high-frequency effects were not negligible as was initially assumed. The gains were backed off until the system was stable, and a more comprehensive analysis was performed to estimate the high-frequency dynamics.

First, the step response of $G(s)$ was recorded by applying a series of 20 step torques to the motor drive, recording the resulting tension responses, then time averaging the 20 responses to improve S/N ratio. The FFT of the response was taken, and it showed a cluster of oscillations around 20–30 Hz.

Next, a Bode analysis of $G(s)$ was performed to try to gain more insight into the structure of these dynamics. This analysis was done using a sinusoid generator to apply an oscillatory (2–300 Hz) command torque input to the system. The resulting tension signal was compared to the input signal to obtain frequency response information. Fig. 7 shows the results for both a single-wrapped pulley (which allowed some slippage), and a double-wrapped pulley (which did not). Notice that the Bode analysis for the double-wrapped pulley begins with a slope of +5 dB/dec; this suggests that there is one more zero than poles at the low frequencies. Note also that the Bode plot contains two peaks followed by a slope of −60 dB/dec; this suggests that there are four poles in this vicinity, in two sets of complex conjugate pairs. The first set of poles appears to be located at about 18 Hz, the second set at about 28 Hz.

TABLE III
MODEL POLES AND ZEROS

Model Poles	Model Zeros	
-1.70	-1.36	<- original set
-3.5 ± 58 i	-8.5	<- higher
-1.3 ± 90 i	∞	frequency

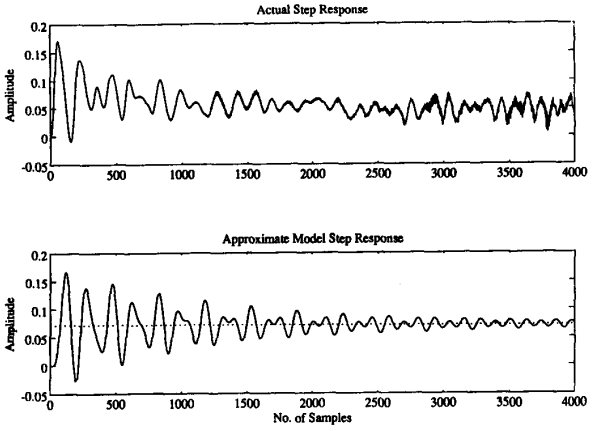

Fig. 8. Model comparison.

The step response also suggests that the original zero at the s-plane origin is not exactly at the origin; instead it is slightly to the left of the origin. This fortunately results in better control than had the zero been exactly at the origin. Now, the system under PI control can effectively reject step-type disturbances, instead of just impulse type. The poles and zeros are shown in Table III.

Using this new information, we improved upon our estimate of the system structure. It must be noted that these additional dynamics are coupled to the support cable lengths, and so are not constant as the system moves. In fact, as the robot moves from one end of the vertical workspace to the other, the two sets of oscillatory poles will come together, then move apart again. The resulting system is therefore nonlinear, but can be modeled using a linear approximation.

The actual step response, in comparison with this model structure's step response, is shown in Fig. 8. These dynamics are fairly lightly damped, so it is easy to see how they might become unstable as the feedback gain is turned up. It therefore appears that the performance will be limited by these higher frequency dynamics which cross the stability boundary when the feedback gain is increased. The best values for feedback gain (K) and zero location (b) determined experimentally are: $8.3 \cdot 10^{-4}/0.3$, respectively.

One way to minimize the effects of these dynamics is to filter them from the sensor output. A lowpass filter will allow all the frequencies of interest to pass through the control algorithm, while suppressing the higher frequency oscillations. In other words, a predetermined pair of dominant poles is placed at a lower frequency than that of the oscillations, in order to decrease the amplitude of the higher frequency oscillations. A second-order Butterworth digital lowpass filter with a cutoff frequency of 1/20 that of the sampling frequency was implemented, and this helped diminish the effects of

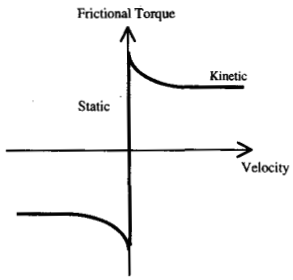

Fig. 9. Tustin's friction approximation.

these oscillations. The results from this control algorithm are discussed in the section on results.

V. Friction Compensation

Preliminary experiments showed that the performance of the PI controller was very good for the dynamic case. But, when a force was applied while the system was at rest, the effects of static friction were very noticeable (see Section VII). It has been shown that fuzzy systems [4] can be used effectively to compensate for static friction, and we decided to implement a fuzzy system to work in parallel with the PI controller, in order to minimize the effects of static and coulomb kinetic friction.

When the system is at rest, static friction appears as a dead zone in the otherwise linear torque response of the system. In other words, until a certain threshold amount of torque is applied, there will be no effective output torque, and the system will remain at rest. Any increase in torque beyond this "breakaway" threshold will result in an accelerating torque being applied to the system.

When the system is in motion, however, there is still some frictional effect due to coulomb kinetic friction. This appears as a constant deceleration torque regardless of the velocity of rotation or applied torque. This deceleration torque is generally smaller in magnitude than the breakaway torque; in fact in our experimental system it was approximately one-half the breakaway torque. The two domains of static and coulomb kinetic friction are not mutually exclusive; there is some blending of the two for very slow velocities. According to Tustin's approximation [3], the blending can be modeled by an exponential as shown in Fig. 9.

Since modeling the parameters of this relationship for a real system is very difficult [3], precise compensation is not practical. A much more feasible solution is to use a fuzzy control algorithm to generate the friction compensation torques.

To compensate for static friction, we must be able to sense whether the motor is stationary. If it is indeed stationary, we must know the command torque being applied by the PI controller, and the torque being created by the tension imbalance between the payload and counterweight. Using this information, we can generate a static friction compensation torque to add to the PI controller output.

To compensate for coulomb kinetic friction, we must know which direction the motor is turning in order to add a compensation torque in that direction to the controller output.

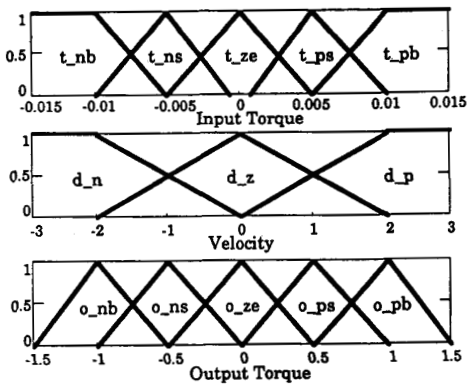

Fig. 10. Fuzzy membership functions.

	t_nb	t_ns	t_ze	t_ps	t_pb
d_n	o_ns	o_ns	o_ns	o_ns	o_ns
d_z	o_nb	o_ns	o_ze	o_ps	o_pb
d_p	o_ps	o_ps	o_ps	o_ps	o_ps

Fig. 11. Fuzzy rule base.

These then will be the three inputs to our fuzzy friction compensation: motor velocity (as estimated by the first difference in encoder position), PI command torque, and tension imbalance torque (as estimated by payload tension T_1). The two torque signals have identical impact on the static friction, so they can be combined to reduce computational overhead in the fuzzy control algorithm.

The two inputs have the following sets of membership functions. There are five membership functions for input torque: negative big, negative small, near zero, positive small, positive big (nb, ns, ze, ps, pb). There are three membership functions for velocity: negative, near zero, and positive. There are also five membership functions for output torque. All three sets of membership functions are illustrated in Fig. 10.

The fuzzy rule base is as follows: if the velocity is negative or positive, the output should be small in the direction of rotation, regardless of applied torque. If the velocity is near zero, then the output should be large in the direction of the applied torque. This should effectively compensate for coulomb kinetic and static friction. When the system is at rest, however, a very small applied torque will result in a large friction compensation torque being applied. To reduce this undesirable artifact, a small output torque is substituted for the large one when the applied (input) torque is small. This can be more easily seen in the rule base table shown in Fig. 11.

VI. Offset Drift Adaptation

The preceding fuzzy control algorithm assumes that when the system is at rest, the PI output is zero. However, this is not always the case. If the two masses are not exactly equal, there will be a constant dc torque output required to balance them. Also, there is an offset bias in the torque servo, which needs to be compensated for. These offsets fall under the normal

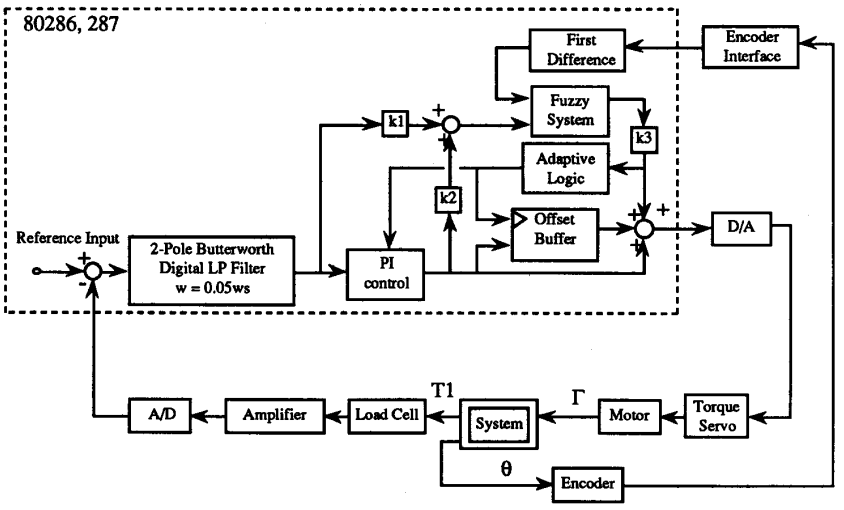

Fig. 12. Control system block diagram.

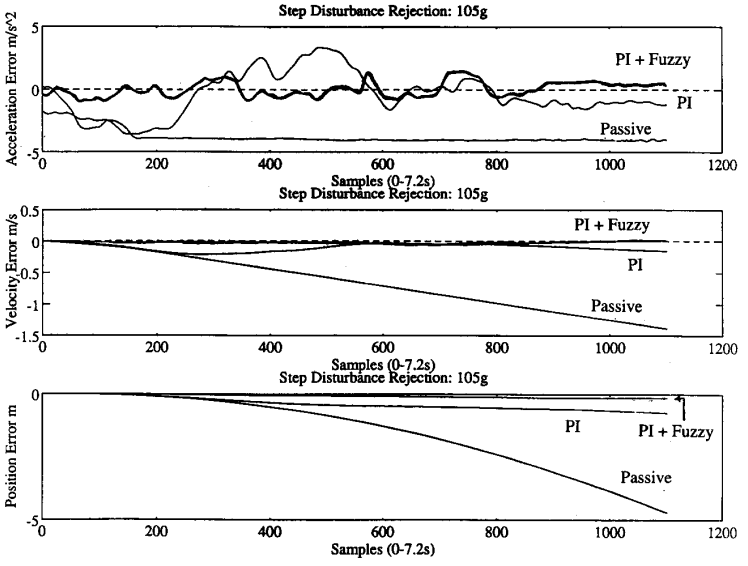

Fig. 13. Response to large step disturbance.

operation of the PI control, but they do not adhere to the above design assumption made for the fuzzy control algorithm.

A solution to this problem is to remove any dc component from the output of the PI control, and use a buffer to hold this dc value and add it to the output torque command. We implemented an offset buffer that relieves the PI control of the responsibility of maintaining all dc outputs. This ensures that the fuzzy algorithm will work correctly. This offset buffer begins with a value that has been experimentally determined to compensate for mass imbalance, and it is updated whenever a dc offset term is detected on the output of the PI control. The detection scheme is the following: if the output of the fuzzy controller has remained zero (stationary system) for 100 cycles, then the average value of the PI command torque over those 100 cycles is simultaneously added to the buffer and subtracted from the PI integral. But, if the variance of the PI command torque during this interval is greater than a specified amount, this operation is suppressed. This suppression is done to prevent a very slow acceleration from being mistaken for a dc offset.

VII. Results

The complete system is shown in Fig. 12, where the double boxed "System" is that shown in Fig. 3. The entire software control system is enclosed in a dashed box.

The results (Fig. 13) show the experimental system response to a step-type input disturbance force of 1.03 N. Given are the acceleration error, velocity error, and position error as

functions of time. In each plot, the step disturbance errors of the system are shown with three types of gravity compensation: passive, PI controlled, and PI control with fuzzy friction compensation.

The system was running at 230 Hz, with a feedback gain of $8.3 \cdot 10^{-4}$ and a time constant of -3. Notice that in the first set of responses, using PI + fuzzy compensation the instantaneous acceleration error is reduced by over 65% compared to the passive compensation scheme. Notice also that when using PI + fuzzy, velocity error remains finite and very near zero, and position error remains finite and small, as compared to the passive compensation scheme where velocity error is linear in time, and position error is quadratic in time.

The effect of fuzzy friction compensation is substantial. By comparing the PI compensation and the PI + fuzzy compensation, one can easily see the effect of static friction compensation. Notice in the velocity error plot of Fig. 13 that the PI-controlled response initially follows the passive response until the integral term has built up sufficiently to break the static friction. As a result, the acceleration error overshoots substantially before returning to its steady-state value. Fuzzy friction compensation eliminates this undesired effect remarkably well, resulting in a more uniform tracking of the ideal trajectory.

It is estimated that acceleration error reduction could be increased to about 85% if more time were spent on optimizing the gains. Note that the acceleration error is still fairly oscillatory; this is due to the high-frequency dynamics, whose damping may be increased by manipulation of the controller gains.

VIII. Conclusion and Discussion

It was proposed that an active control system be designed to improve the passive gravity compensation system currently in use. These nonideal effects were: increased inertial mass of robot in the vertical direction, frictional damping (static, coulomb kinetic, viscous), and imprecise counterweight mass. The proposed method of PI control and fuzzy friction compensation has been shown through simulations and experiments to overcome these three effects.

This work is significant in that it allows for a more realistic simulation of zero or partial gravity than has been achieved through passive compensation. It also addresses the problem of static friction in the system which has heretofore been neglected. A disadvantage in this method is that the high-frequency dynamics are not actively controlled, only filtered.

One topic to be addressed in future work is how to actively compensate for the nonlinear oscillatory behavior of the system due to cable compliance. Ideally, these dynamics can be modeled based on cable lengths, so that an adaptive control algorithm could be used to adjust the controller to the current position of the robot.

References

[1] Y. Sato *et al.*, "Micro-G emulation system using constant-tension suspension for a space manipulator," in *Proc. 1991 Int. Conf. Robot. Automat.*, Sacramento, CA, IEEE, pp. 1893–1900.
[2] Y. Xu, B. Brown, S. Aoki, and T. Kanade, "Mobility and manipulation of a light-weight space robot," *J. Robotics Autonomous Syst.*, vol. 13, pp. 1012, 1994.
[3] B. Armstrong, "Friction: Experimental determination, modeling and compensation," in *Proc. 1988 Int. Conf. Robot. Automat.*, Philadelphia, PA, IEEE, pp. 1422–1427.
[4] C. C. Lee, "Fuzzy logic in control systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–35, 1990.
[5] G. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Second Ed. Reading, MA: Addison-Wesley, 1991.
[6] G. S. Virk, *Digital Computer Control Systems*. New York: McGraw-Hill, 1991.

Gregory C. White received the B.S. degree in electrical engineering in 1992 from Carnegie Mellon University, Pittsburgh, PA.

While pursuing his bachelor's degree, he worked as an engineer and researcher at the Robotics Institute, Carnegie Mellon University. As a member of the Self-Mobile Space Manipulator research group, his work included research into fuzzy and hybrid control systems, system analysis and modeling, and force reflection for teleoperation. His current research interests include system analysis, signal processing, acoustics and psychoacoustics.

Yangsheng Xu received the Ph.D. degree from the University of Pennsylvania in 1989.

Since November 1989, he has been working with the Robotics Institute of Carnegie Mellon University where he is currently a Research Scientist. He developed a control system for the Self-Mobile Space Manipulator (SM^2) and has been working on the issues related to dynamics and control of space robots, learning control and skill learning, manufacturing automation telerobotics, and assembly planning. His current interests include space robotics, and dynamically stable machines.