Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems Yokohama, Japan July 26-30, 1993

AN ACTIVE Z GRAVITY COMPENSATION SYSTEM

Gregory C. White, Yangsheng Xu
Robotics Institute, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213 USA.

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 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 non-constant gravity force compensation in the case that the payload mass is not constant. The only additional hardware requirements needed for the implementation of this system on a passive counterweight balance system are: a strain gauge tension sensor, and a torque motor with encoder.

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], or 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 has clear disadvantages, but can be used with any number of objects for complex simulations.

The Robotics Institute has 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 (figure 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.

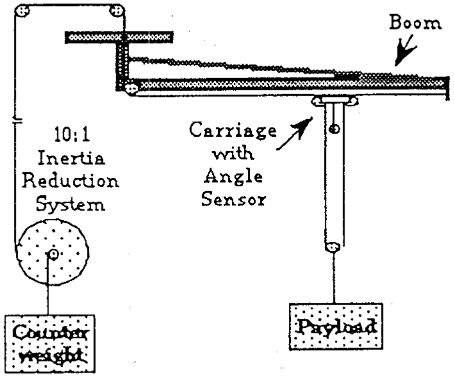


Fig. 1: CMU Gravity Compensation System (GCII)

As was briefly mentioned above, there are inherent weaknesses in this system that need to be improved upon for accurate, reliable, fast tracking of the robot motion, and complete gravity force compensation.

1) The passive counterweight adds additional inertial mass to the system. The result is an increase in the apparent mass of the robot, which reduces the robot system bandwidth in the vertical direction. This increase in mass is kept to 10% by an inertia reduction mechanism, but none the less is still a troublesome non-ideality of the system.

2) The inherent friction of the pulley system is considerable because of the complexities of the cable routing. The cable passes over no less than 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 z-axis (vertical direction) of the current GC system. This system will be a hybrid of the passive counterweight and active motor control systems. 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 system's pulleys, thereby modulating the tension in the support cable. By doing so, we aim to minimize the effects of 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 easily useable for partial gravity simulations (e.g. lunar, or non-earth planetary).

0-7803-0823-9/93/$3.00 (C) 1993 IEEE

II. Modelling of System Dynamics

The system is 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 figure 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.

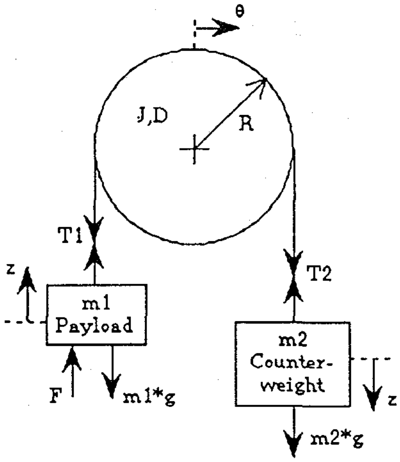


Fig 2: Simplified System

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 dependant 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 for each body from figure 2.

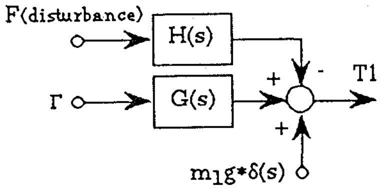
Table 1

For m_1:	$T_1 = m_1(g + \ddot{z}) - F$
For m_2:	$T_2 = m_2(g - \ddot{z})$
For pulley:	$J\ddot{\theta} + D\dot{\theta} = \Gamma_{ms} + R(T_2 - T_1)$

where:
T_1 = robot cable tension
T_2 = counterweight cable tension
m_1 = robot mass
m_2 = counterweight mass
z = vertical position
θ = angular position
J = inertial moment of pulley
D = viscous damping of pulley
F = externally applied force
Γ_{ms} = effective motor torque
R = radius of pulley
g = acceleration of gravity

If we assume zero compliance in the cables, z=Rθ. 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 modelled 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 figure 3 .



$$G(s) = \left(\frac{m_1 R}{\alpha}\right)\frac{s}{s + D/\alpha}$$

$$H(s) = \left(\frac{\alpha - m_1 R^2}{\alpha}\right)\frac{s + \dfrac{D}{\alpha - m_1 R^2}}{s + D/\alpha}$$

$$\text{where } \alpha = J + R^2(m_1 + m_2)$$

Fig. 3: Block Diagram and Transfer Functions

III. Controller Design and Simulations

Since the parameter to be controlled is the tension in the cable supporting the payload, it is most straightforward to measure it directly, using a strain gauge. The control torque can be provided by a DC torque motor and torque servo system.

The controller should drive the motor to provide the proper torque so that the mass-acceleration product ($m\ddot{z}$) is equal to the externally applied force (F). Solving the differential equations of table 1 yields:

$$\Gamma_{ms} = -\left[\left(\frac{R^2 m_2 + J}{R}\right) * a + (D/R) * v + Rg(m_1 - m_2)\right].$$

It is clear that at steady state (i.e. after all oscillations have settled), the motor torque must be proportional to a linear combination of the acceleration, the velocity, and a constant term related to the possible mass imbalance. Using this equation, a motor can be chosen that will provide enough torque for the desired operating range.

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 $\underline{z} = [\,\ddot{z}, \dot{z}, z\,]^T$. 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:

$$\underline{z} = \frac{1}{m}\left[F_o \quad \int F_o dt \quad \int\!\!\int F_o dt^2\right]^T.$$

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:

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

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 (~1s), the most straight forward approach is to use a type I (PI) controller, which will completely reject impulse type disturbances, and track step type disturbances with a finite steady state error in this system configuration.

In order to design the discrete time (DT) controller, it is necessary to discretize the system model so that DT root loci can be drawn. Using the matched pole-zero approach, the DT transfer functions for the system shown above are:

$$G(z) = \frac{m_1 R}{\alpha} * \left(\frac{z - 1}{z - e^{-DT/\alpha}} \right) \qquad \&$$

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

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:

$$C(z) = \frac{\Gamma}{T_1} = K * \left(\frac{z - b}{z(z - 1)} \right) \qquad \&$$

$$\Gamma(n) = K*(T_1(n-1) - b*T_1(n-2)) + \Gamma(n-1) .$$

The DT root locus of G(z)C(z) is shown in figure 4.

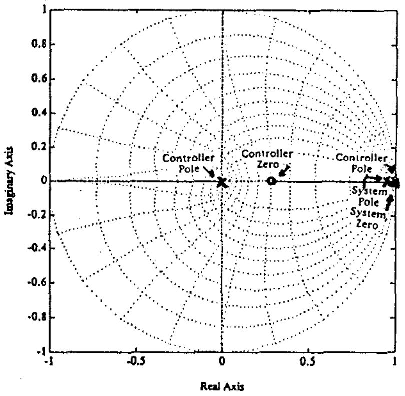


Fig. 4: Z-Plane Root Locus of G(z)C(z)

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 $5*10^{-3}\%$ for $s < 0.5/T$.

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

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

The time response for the uncontrolled and controlled systems, due to a step disturbance of magnitude 1 Newton has been simulated on computer, and is shown in figures 5 & 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.

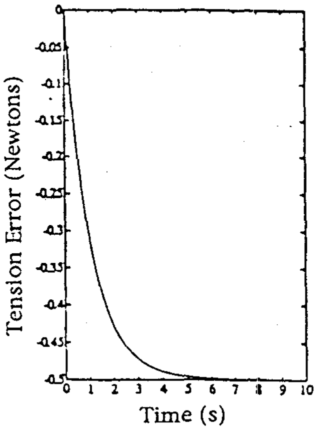


Fig. 5: Uncontrolled System Step Disturbance Rejection

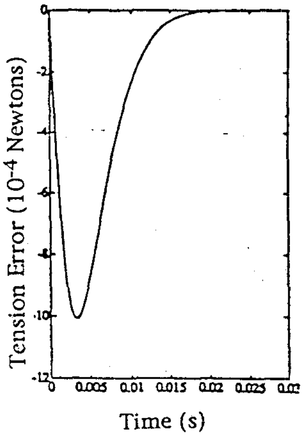


Fig. 6: Controlled System Step Disturbance Rejection

IV. Experimental Setup

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

Hitachi FCL "pancake" motor with quadrature encoder
Omega LCL-010 strain gauge
National Semiconductor LH0036 inst. amplifier
Two masses (262g, 327g)
Lightweight nylon string

Personal computer with i80286 CPU, and i80287 FPU
Data Translation, DT2801 12 bit 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 must be made. The other system parameters can be easily measured for the experimental system. The results are shown below.

Table 2

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

The system transfer functions are:

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

The controller transfer function is:

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

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:

$$\Gamma(n) = 260*T_1(n-1) - 216*T_1(n-2) + \Gamma(n-1) - 19.41 .$$

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 more a 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 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 (10 - 50 Hz) command torque input to the system. The resulting tension signal was compared to the input signal to obtain frequency response information. Figure 8 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 plot for the double wrapped pulley begins at 14 Hz with a slope of +20dB/dec, this suggests that there is one more zeros 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.

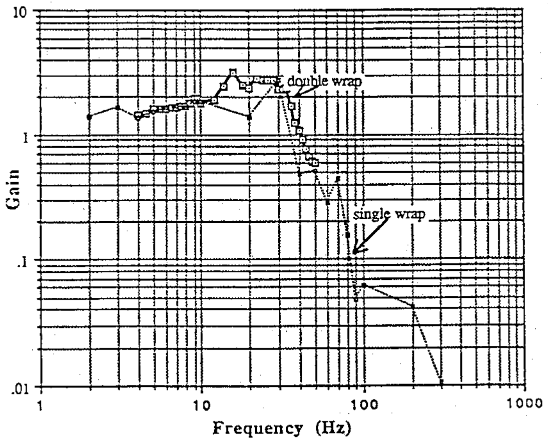


Fig. 8: Bode Plot

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.

Using this new information, an improved estimate of the system structure was made. It must be remembered 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 modelled using a linear approximation.

Table 3

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

The actual step response, in comparison with this model structure's step response is shown in figure 9. 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: $8.3*10^{-4}$, 0.3.

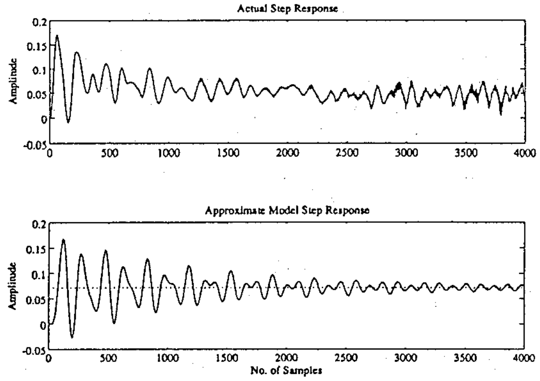


Fig. 9: Model Comparison

There are most likely higher frequency pole-zero pairs introduced by the motor and torque servo. These are generally non-oscillatory, and small in amplitude. These dynamics do not contribute to the instability, but could introduce problems if a more complex controller were attempted, for example, if the controller design was such that it added damping to the poles described above.

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 pre-determined pair of dominant poles are placed at a lower frequency than that of the oscillations, in order to decrease the amplitude of the higher frequency oscillations. A 2nd 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 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. It has been shown that fuzzy systems [4,6] 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 figure 10.

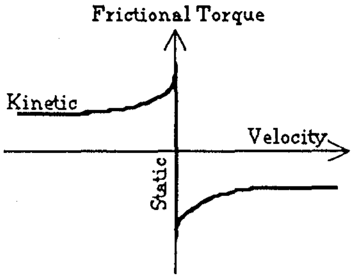


Fig. 10: Tustin's Friction Approximation

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 kinetic friction compensation torque in that direction to the controller output.

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 membership functions. There are three membership functions for velocity: negative, near zero, and positive. There are five membership functions for applied torque: negative big, negative small, near zero, positive small, positive big. There are also five, similarly named membership functions for output torque. All three sets of membership functions are illustrated in figure 11.

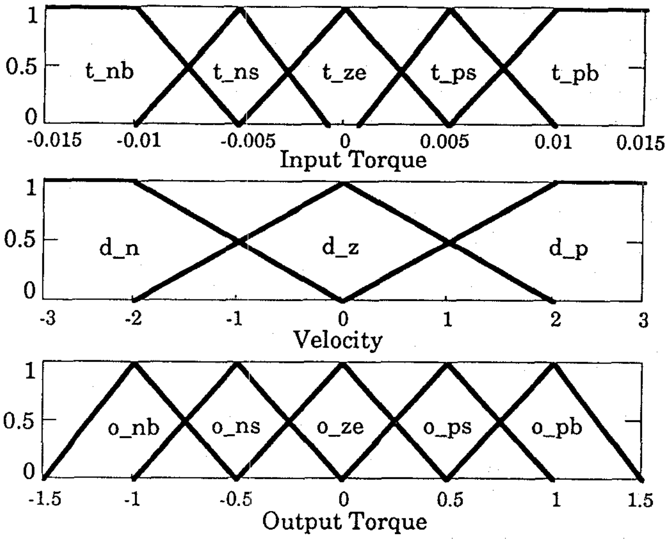


Fig. 11: Fuzzy Membership Functions

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 figure 12.

	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. 12: Fuzzy Rule Base

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 in the torque servo, which needs to be compensated for. These offsets fall under the normal 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 variable 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 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 (static 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 figure 13, where the double boxed "System" is that shown in figure 3. The entire software control system is enclosed in a dashed box.

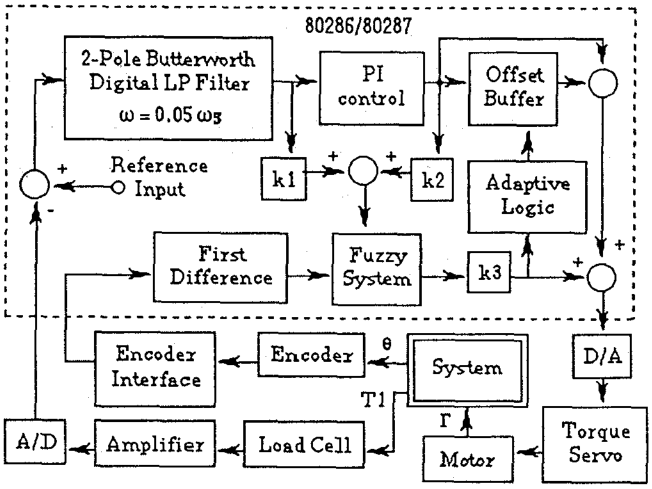


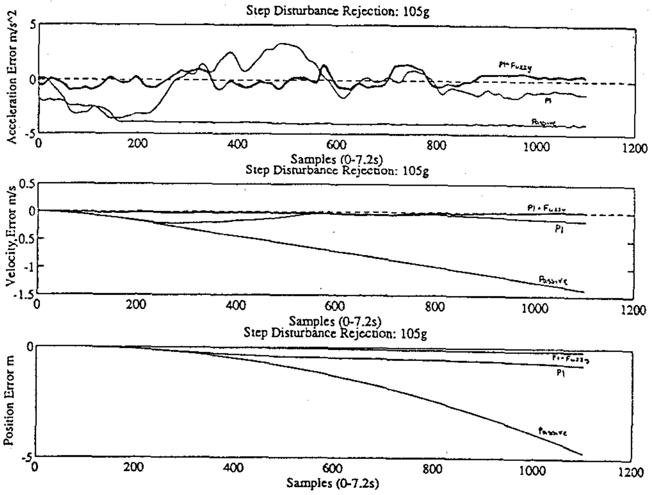
Fig. 13: Control System Block Diagram

The results (figures 14-16) 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(**K**) of: 8.3*10^{-4} and a time constant (**b**) 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 figure 15 where 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 PI controller 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.

These results come after a couple of days of tuning the controller. I would estimate 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.



Figs. 14-16: Response to Large Step Disturbance

VIII. Conclusions and Discussions

It was proposed that an active control system be designed to reduce the non-ideal effects of the passive Gravity Compensation system currently in use. These non-ideal effects were: increased inertial mass of robot in 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. The disadvantages in this method are that the high frequency dynamics are not actively controlled, only filtered. Perhaps a more complex linear control algorithm could add damping to these dynamics.

This is one topic to be addressed in future work; how to actively compensate for the non-linear oscillatory behavior of the system due to cable compliance. Ideally, these dynamics can be modelled based on cable lengths, so that an adaptive control algorithm can be used to adjust the controller to the current position of the robot.

References

[1] Sato, Y., et al., "Micro-G Emulation System Using Constant-Tension Suspension for a Space Manipulator", in Proc. 1991 Intl. Conf. on Robotics and Automation, Sacramento: IEEE, pp. 1893-1900.

[2] Xu, Y., Brown, B., Aoki, S., Kanade, T., "Mobility and Manipulation of a Light-Weight Space Robot", in Proc. 1992 Intl. Conf. on Intelligent Robots and Systems, Raleigh: IEEE, pp.11-19.

[3] Armstrong, Brian, "Friction: Experimental Determination, Modeling and Compensation", in Proc. 1988 Intl. Conf. on Robotics and Automation, Philadelphia: IEEE, pp.1422-1427.

[4] Lee, C.C., "Fuzzy Logic in Control Systems", IEEE Trans. Systems, Man, and Cybernetics , vol. 20, no. 2, p.404-35, 1990.

[5] Franklin, G., J.D. Powell, A. Emami-Naeini, Feedback Control of Dynamic Systems, Second Edition. Addison Wesley, 1991.

[6] Virk, G.S., Digital Computer Control Systems, McGraw Hill, 1991.