AN ACTIVE Z GRAVITY COMPENSATION SYSTEM

Greg White Yangsheng Xu

CMU-RI-TR-92-09

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

July 1992

Contents

List of Figures

List of Tables

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 GC system using a hybrid PI/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 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. This is especially true when one is working with a lightweight, low power manipulator [2]. 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.

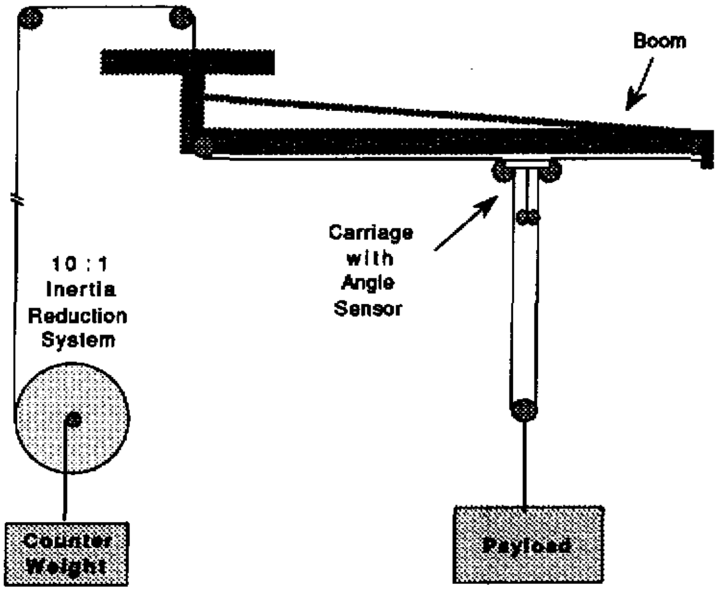


Fig. 1: CMU Gravity Compensation System (GCII)

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.

4) For the current experiments, the robotic payload is supported at a second point by a portion of space station truss. The effective mass to be supported by the GC system is therefore not constant, but changes as more mass is supported by the second (static) support point as the robot moves its center of mass. In general, there will be a need to compensate gravity force which varies during operation, which is not possible using the existing system.

We propose to control the z-axis (vertical direction) of the current GC system actively. Thus creating a hybrid of the passive counterweight and active motor control. Using a strain gauge to sense cable tension, and an encoder to sense payload position, a DC servo motor can be used to drive one of the system's pulleys to servo torque into the system, thus modulating cable tension to minimize the effects of friction and counterweight inertia, and to completely compensate for gravitational force. The advantages of this approach are that it can overcome the four 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).

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 the sum of the inertial moments of the individual pulleys in the real system.

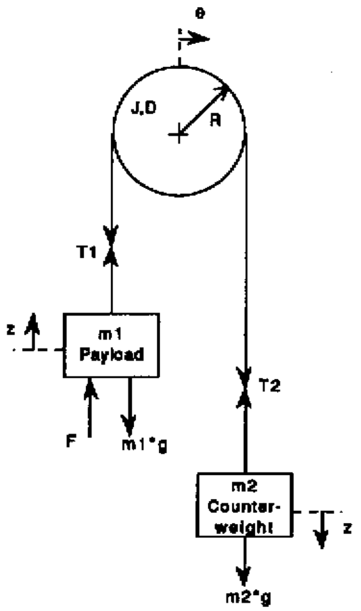


Fig 2: Simplified System

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 lower 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 so constantly change 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	m_1 = robot mass
T_2 = counterweight cable tension	m_2 = counterweight mass
z = 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

If we assume zero compliance in the cables, z=Rθ. Combining these three equations results in the system diagram shown in figure 3.

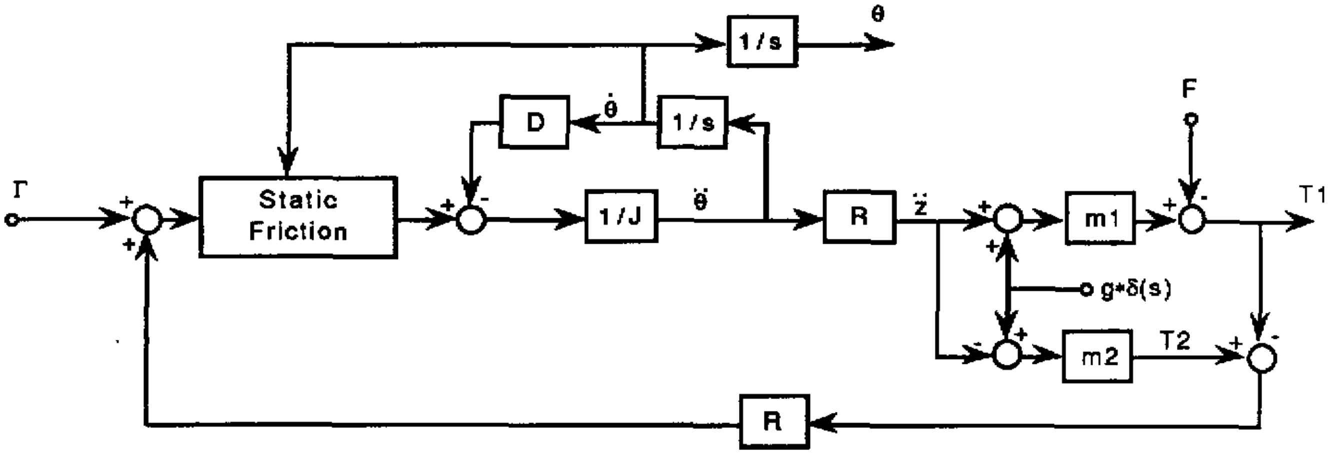
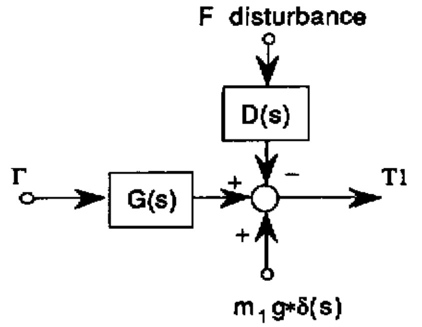


Fig. 3: System Block Diagram

Where the static friction will be described later.

In this system, T_1 is the parameter to be controlled, and can also be sensed directly. Other parameters such as θ and ω can be easily sensed by an encoder or tachometer, and their input can aid in the estimation of the system state if T_1 alone is not sufficient. For now, we will sense θ only so that the work space can be easily limited, we will assume that the system is modelled well enough that T_1 will give us a good indication of state.

This system can be simplified using standard block-diagram rules, as is illustrated in figure 4, below.



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

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

Fig. 4: Block Diagram Simplification 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 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 figure 2 yields:

$$\Gamma_{ms} = -\left[\left(\frac{R^2 m_2 + J}{R m_1}\right) * F + \left(\frac{D}{R m_1}\right) * \int F dt + Rg(m_1 - m_2)\right]$$

or

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

It is clear that at steady state, that is after all dynamics 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_0/m). The velocity of the body equals the time integral of the acceleration, and the position of the body equals the time integral of the velocity. Assuming we use an inertial frame of reference so that the initial position and velocity are zero, the resulting state vector is:

$$z = \frac{1}{m}\begin{bmatrix} F_o \\ \int F_o dt \\ \int\int F_o dt^2 \end{bmatrix}.$$

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:

$$\underline{\delta z} = \frac{1}{m}\begin{bmatrix} \delta T \\ \int \delta T dt \\ \int\int \delta T dt^2 \end{bmatrix}.$$

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 w.r.t. an equivalent mass in a zero (or partial) gravity environment. The first step in accomplishing this objective is to control the highest order state variable, namely the tension itself. Only by doing this can we monitor the other two state variables, and attempt to drive them as close to zero as possible.

The purpose of the control system is to ensure that the tension in the payload cable tracks a desired control input, regardless of any disturbance forces from the robot. As was just mentioned, the complete rejection of this disturbance is impossible, but through feedback control the effects can be minimized.

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 controller, which will completely reject impulse type disturbances, and track step type disturbances with a finite steady state error in this system configuration.

The rest of the controller dynamics are chosen to improve the speed of response. A single zero with a shorter time constant than the natural time constant of the system should be enough to bring the closed loop poles into a satisfactory location, but other dynamics can be added if needed. See the root locus for details, figure 5. The continuous time controller transfer function will be $C(s) = K^* \frac{s+b}{s}$, in other words, a PI controller

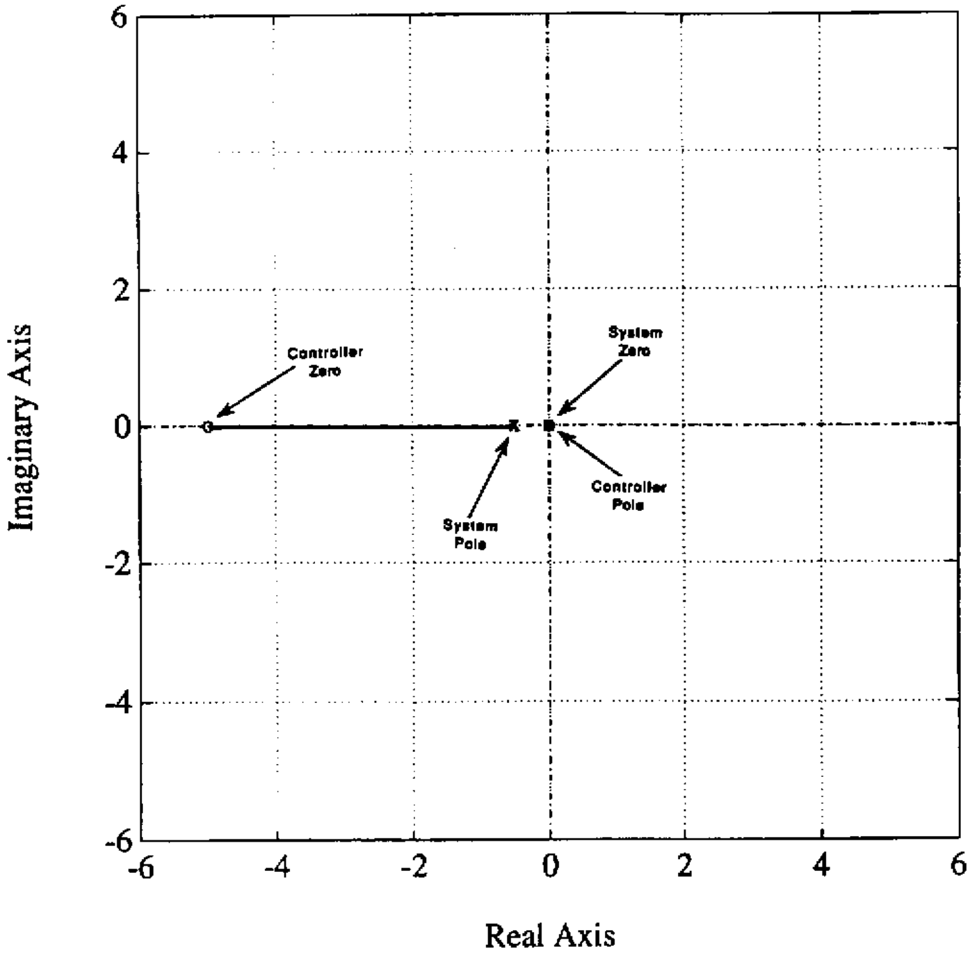


Fig. 5: S-Plane Root Locus

To implement this controller using a digital computer requires digitizing of the controller, and modifications to compensate for the effects of finite sampling rate, discrete output, etc.

Using the Bilinear Transformation, the continuous time controller becomes

$$C(z) = K^*\left(1 + \frac{bT}{2}\right)\left(\frac{z - \left(\frac{2 - bT}{2 + bT}\right)}{z - 1}\right) \qquad \text{or} \qquad C(z) = K_2 * \left(\frac{z - a}{z - 1}\right)$$

where T is the sampling period.

However, there is a delay of one time period between sensing and actuation, therefore an additional pole is unavoidable, resulting in:

$$C(z) = K_2 * \left(\frac{z - a}{z(z - 1)}\right)$$

The resulting controller time function is:

$$\Gamma(n) = K_2 {*}(\ T(n-1) - a {*} T(n-2)\ ) + \Gamma(n-1)$$

To verify the performance of the DT controller, and to set the feedback gain, it is best 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:

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

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

The DT root locus is shown in figure 6.

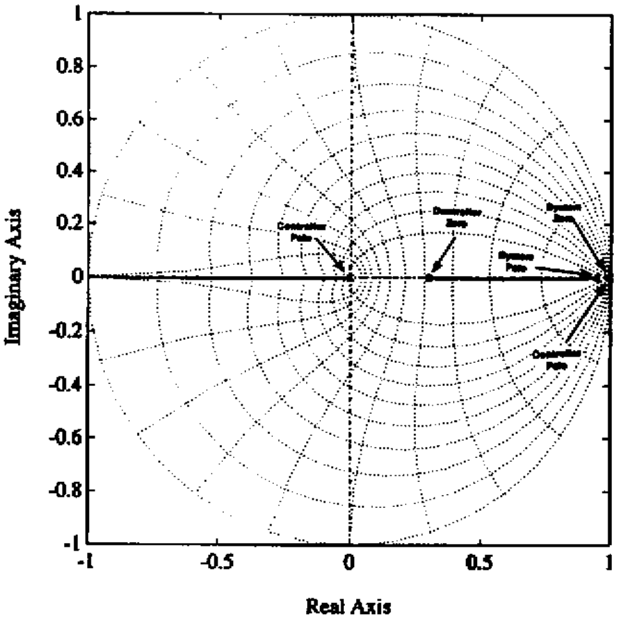


Fig. 6: Z-Plane Root Locus

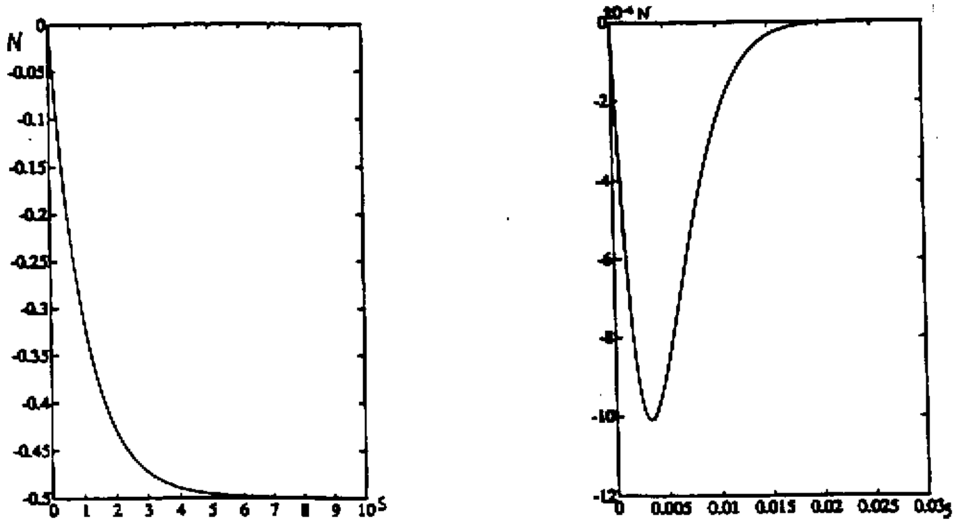
To simulate the operation of this system in the continuous time domain, it is necessary to convert the modified digital controller back into continuous time, assuming a zero-order hold for output, and using the Padé approximant to approximate the finite time delay (one sampling period) between sensing and actuation. To convert the DT transfer function into CT, the following function is employed;

$$C(s) = \left(\frac{1 - e^{-sT}}{s}\right) * C(z);\ \text{setting } z = e^{sT}.$$

The (2,2) Padé approximant [5] should be used to model the time delay terms, it is valid to within $5*10^{-3}\%$ for $s < 0.5/T$.

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

The time response for the uncontrolled and controlled system, due to a step disturbance of magnitude 1 Newton has been simulated on computer, and is shown in figures 7 & 8. Notice that the uncontrolled system exhibits a rather large steady state error to step disturbance, while this is completely eliminated in the controlled system.



Figs. 7 & 8: Uncontrolled & Controlled System Step Disturbance Rejection

It must be noted that the final objective is to control the tension so that the error state is minimized. The controller thus far described drives the first error state variable to zero, while keeping the second and third error states finite and small during all motions, as will be seen in the section on results.

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 type FCL pancake motor with quadrature encoder
Omega strain gauge and amplifier
Two masses (approx. equal)
Lightweight nylon string

i80286,287 computer system
Data Translation, DT2801 12 bit A/D converter
Technology 80, Model 5312 Quadrature decoder
PMI Torque servo drive

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

<u>Motor Parameters</u>

$K_T = 52 * 10^{-3}$ Nm/A

$I_{static\ friction} = 0.1$ A

$J = 65 * 10^{-6}$ Nms^2

$D = 6.5 * 10^{-6}$ Nms

<u>System Parameters</u>

$R = 0.0095$ m

$m_1 = 0.2615$ kg

$m_2 = 0.3272$ kg

servo gain = 1.482 A/V

Strain gauge output = $1.06\ T_1 + 0.438$ V

So the system transfer functions are:

$$G(s) = \frac{21s}{s + 0.055} \qquad D(s) = (0.8)\,\frac{s + 0.0688}{s + 0.055}$$

And, the controller transfer function must be:

$$C(z) = (K_2 / 77*10^{-3}) \frac{z - a}{z(z - 1)}$$

where K_2 and a have been chosen using root locus techniques to be $\mathbf{K_2} = 21.225$, and $\mathbf{a} = 0.83$. The time series that results from this transfer function is:

$$V_o(n) = 260*V_i(n-1) - 216*V_i(n-2) + V_o(n-1) - 19.41$$

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, see figure 9.

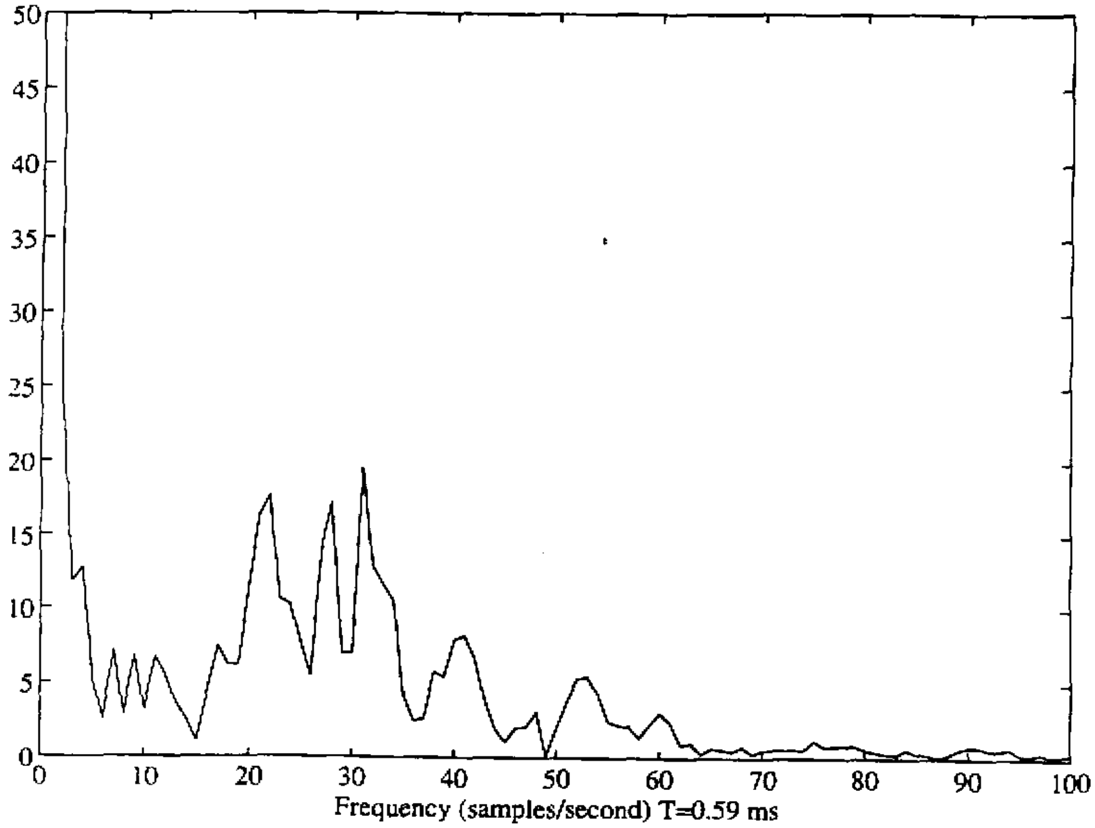


Fig. 9: FFT of System Step Response

Next, a Bode analysis was performed using a sinusoid generator between 10 - 50 Hz to try to gain more insight into the structure of these dynamics. Figure 10 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.

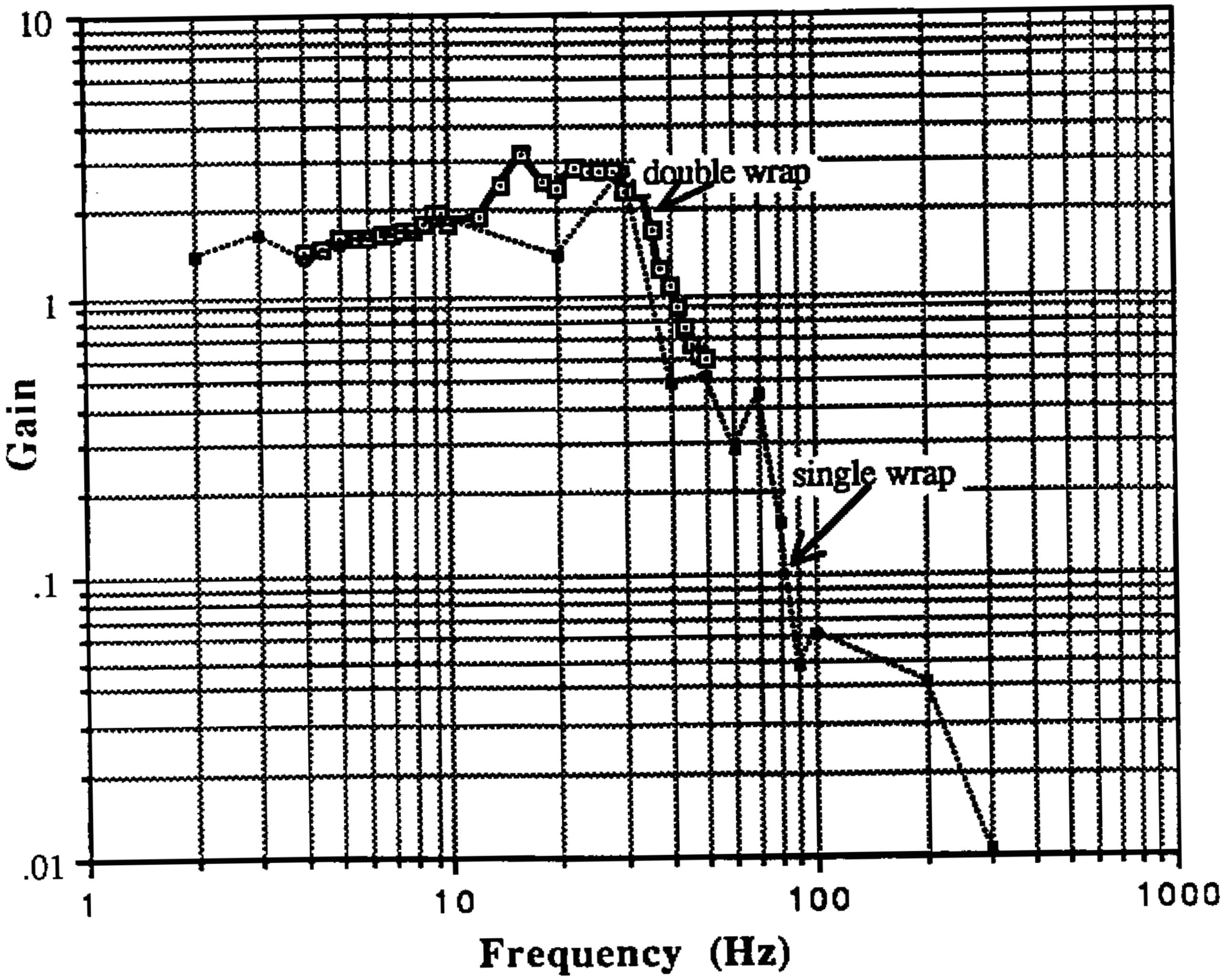


Fig. 10: Bode Plot

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. This results in a 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	Model Zeros
-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 11. 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 and zero location determined experimentally are: $8.3*10^{-4}$, -0.3.

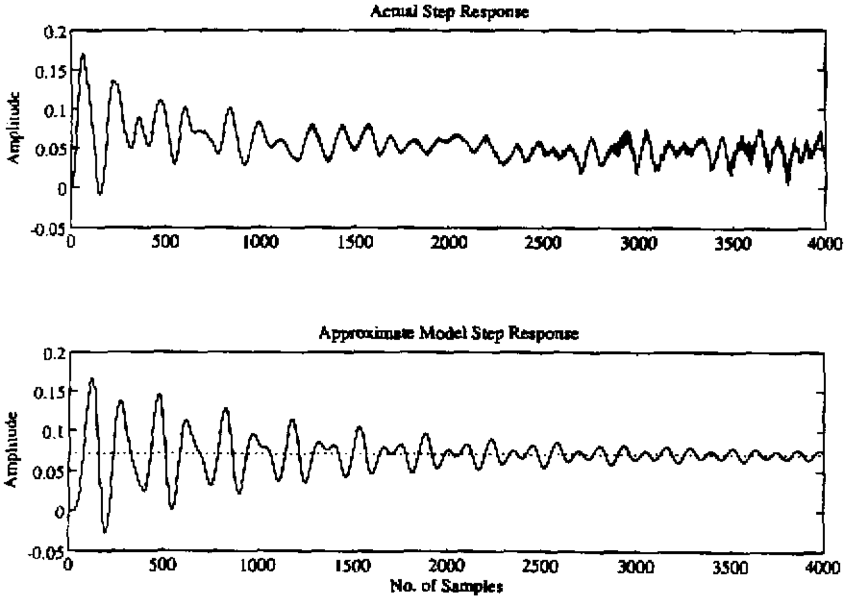


Fig. 11: Model Comparison

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, to add damping to the poles described above.

One way to minimize the effects of these dynamics is to filter them from the sensor output. A low pass 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 low pass 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 is applied while the system is at rest, the effects of static friction are 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 the expected torque being applied to the system.

When the system is in motion, however, there is still some frictional effect due to the coulomb kinetic friction. This appears as a constant deceleration torque regardless of the velocity of rotation or applied torque. This 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, therefore, 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 12.

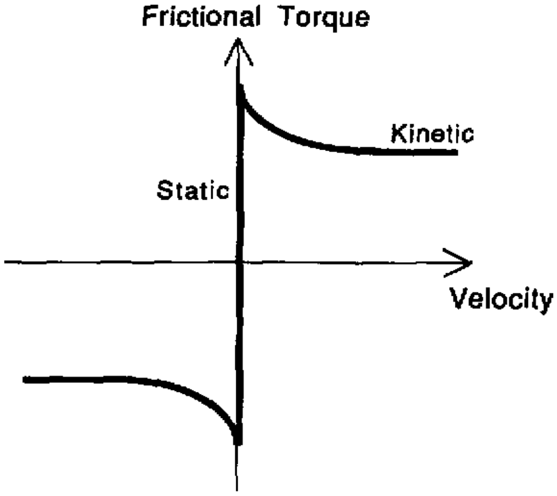


Fig. 12: Tustin's Friction Approximation

It should be noted that this friction does not only apply to torques generated by the motor, but rather to any torque that is applied to the rotational element of the system, so it affects performance even when the system is not actively controlled.

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 that 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 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 13.

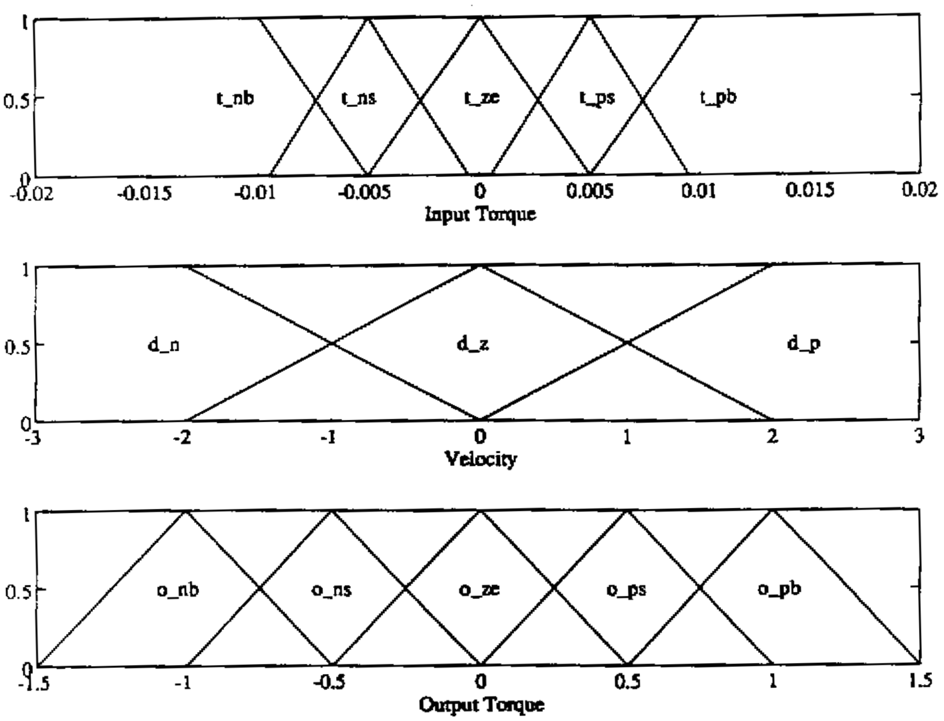


Fig. 13: Fuzzy Membership Functions

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 14.

	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. 14: Fuzzy Rule Base

To illustrate the response of this system for the entire range of inputs, the fuzzy gain surface is plotted in figure 15. This surface describes the input output relationship for the fuzzy system. The vertical axis gives the output magnitude, while the two horizontal axes give a range of input magnitudes.

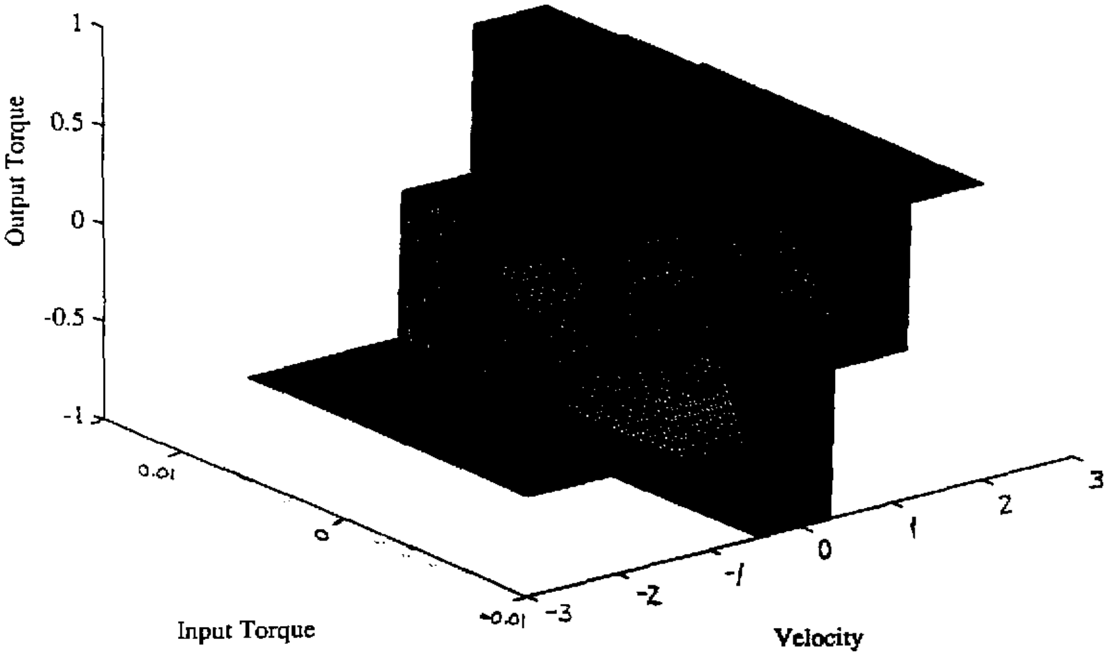


Fig. 15: Fuzzy Gain Surface

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 counterweight mass is not exactly equal to the robot mass, there will be a constant DC torque output required to balance the two masses. 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 determined experimentally 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.

If, through whatever means, a new offset is suddenly incurred that is large enough to incite the fuzzy friction compensation, this adaptation scheme is powerless to recover. This topic has not yet been addressed, but a keystroke has been implemented that toggles the fuzzy friction compensation. If such a situation should arise that the automatic offset adaptation cannot handle, the fuzzy control can be temporarily turned off, allowing the system to adapt to the new offset.

VII. Results

The complete system is shown in figure 16, the double boxed "system" is that shown in figure 3. The entire software control system is enclosed in a dashed box.

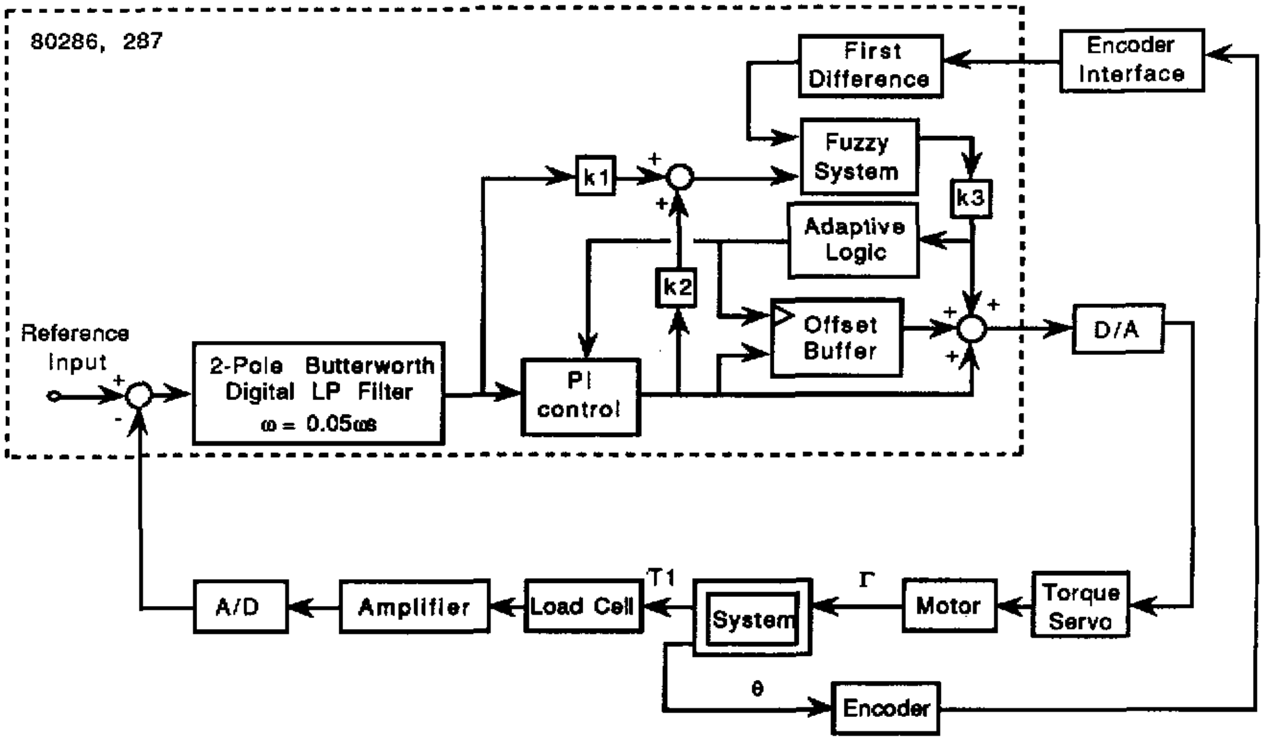
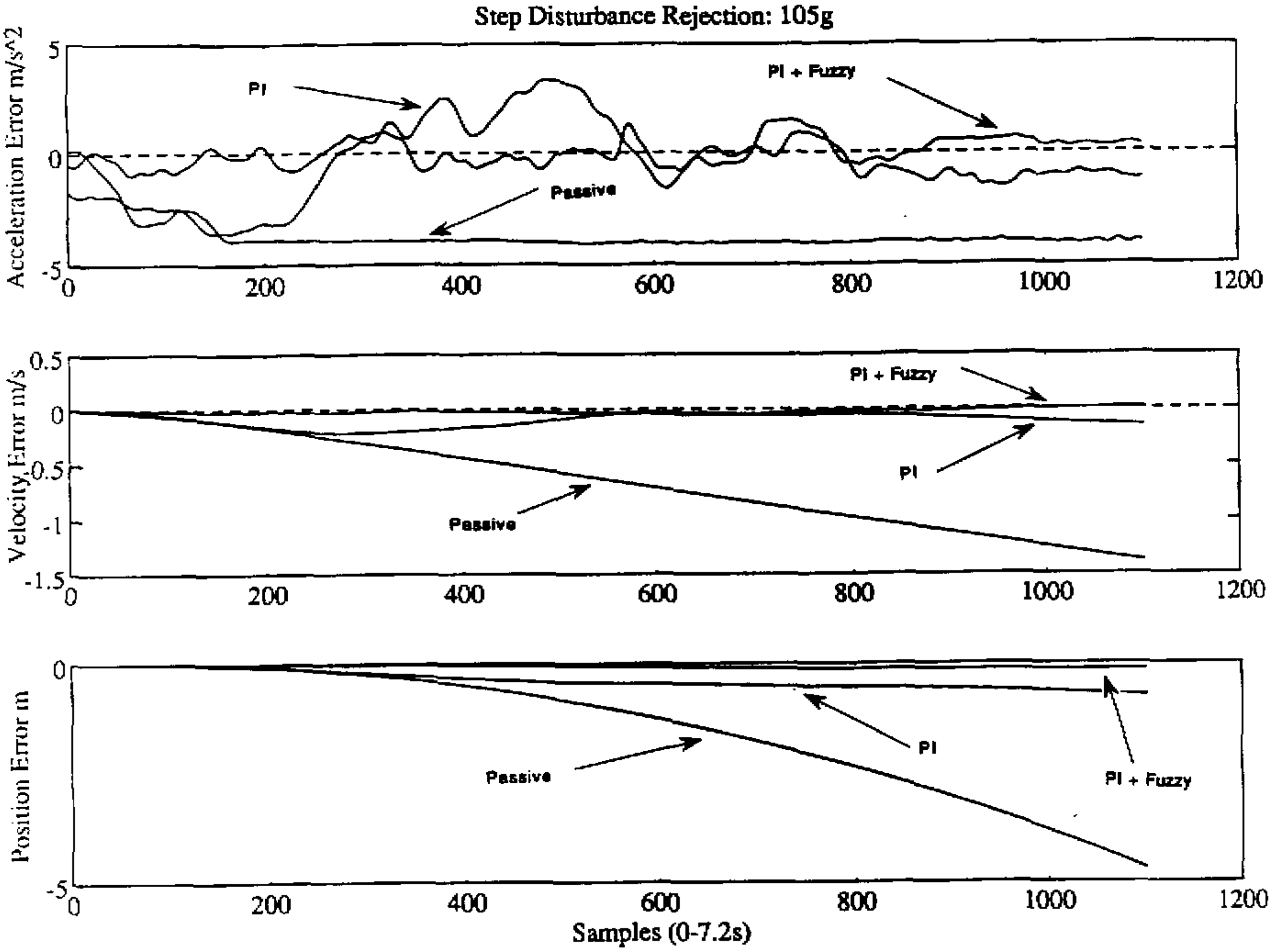


Fig. 16: Control System Block Diagram

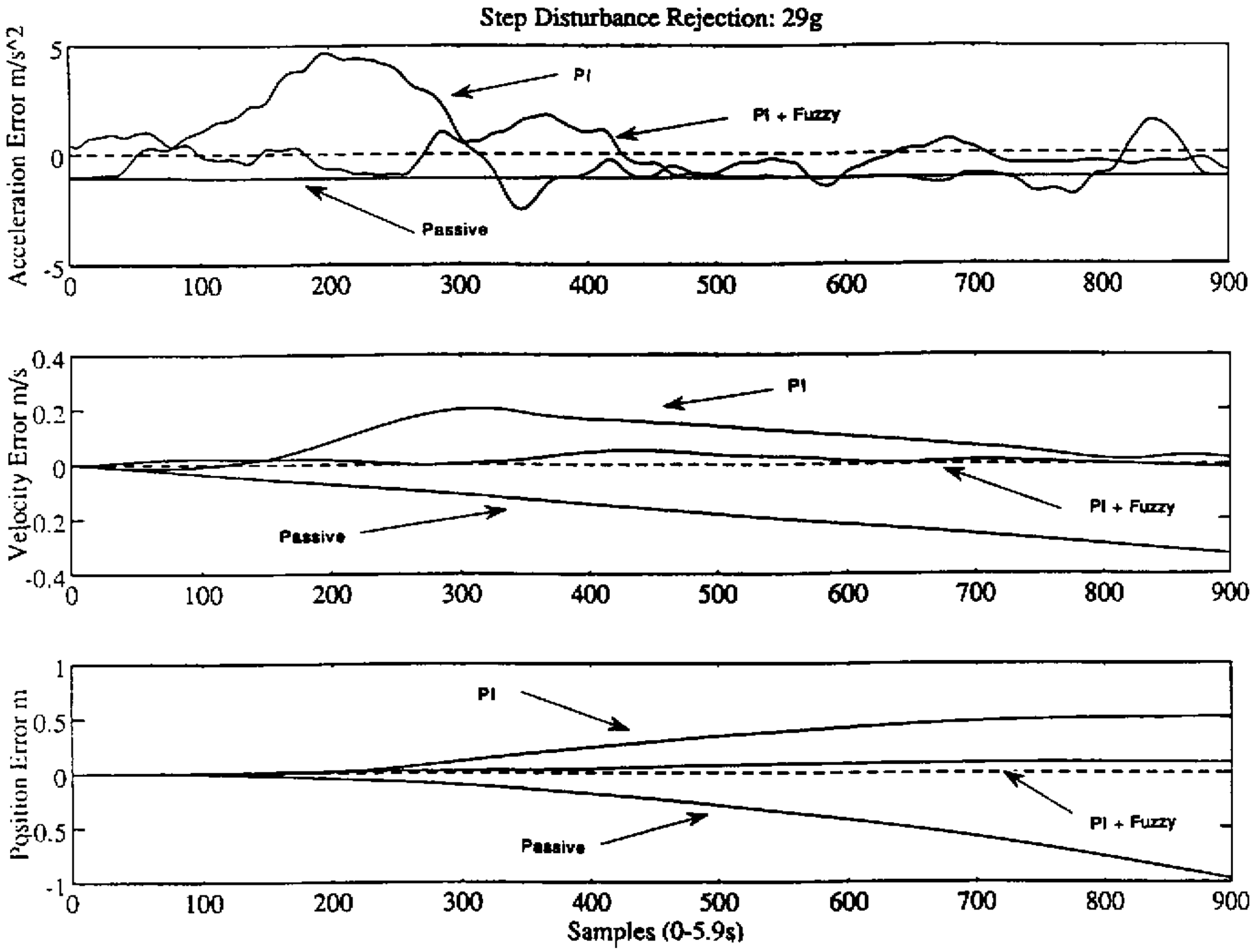
The results show the experimental system response to a step type disturbance input. Given are the acceleration error, velocity error and position error as functions of time. The first set of responses (figures 17-19) is due to an applied force of 1.03 N, while the second set of responses (figures 20-22) results from a much smaller force of 0.286 N. The smaller force was not enough to break the static friction in the uncontrolled system. 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: $8.3*10^{-4}$ and a time constant (a) of: -3. Notice that instantaneous acceleration error is reduced by over 65%, velocity error remains finite and very near zero, and position error remains finite and small. Static friction is also reduced by approximately 80%.

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. 17-19: Response to Large Step Disturbance



Figs. 20-22: Response to Small 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), Imprecise counterweight mass, No provision for variable gravity force compensation.

The proposed method of PI control and Fuzzy friction compensation has been shown through simulations and experiments to overcome the first three effects, and can be easily adaptable to include the fourth.

This work is significant in that it allows for a more realistic simulation of zero or partial gravity than has been achieved through passive control. 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 control algorithm could add damping to these dynamics.

To implement this system on the existing passive GC system, a motor must be chosen that can provide enough torque to fill the desired operating range which is defined by the maximum desired acceleration and velocity given in the equation on page 5. The motor can fairly easily be mounted on the existing inertia reduction system as long as the effects of this location are taken into consideration. A strain gauge tension sensor must be chosen that can operate within the expected range of cable tension. Some analysis (i.e. Bode) must be performed on the existing system to evaluate the oscillatory frequencies, so that a filter can be designed to minimize these effects. Or, if these frequencies are so low that a filter would interfere with the performance of the system to quick movements, a more complex compensation must be employed to add damping to these dynamics.

This then 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, *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, *Proc. 1992 Intl. Conf. on Intelligent Robots and Systems,* Raleigh: IEEE, pp.11-19.

[3] Armstrong, Brian, Friction: Experimental Determination, Modeling and Compensation, *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, 1990, p.404-35.

[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.