BALANCING OF A SMALL SATELLITE ATTITUDE CONTROL SIMULATOR ON AN AIR BEARING

Jeff S. Young
Utah State University
Logan, Utah

Abstract

This paper discusses the balancing of a ground-based satellite simulator. The simulator uses an air bearing as the primary method for the emulation of the frictionless environment of space. Of particular concern for accurate simulation is system balancing. 'Balancing' refers to the process of moving the center of mass(CM) of the simulator near to the center of rotation(CR) of the air-bearing to reduce the interference of gravitational torques.

This paper develops an automatic balancing system for the simulator. This system calculates the system center of mass by analyzing the dynamic sensor data along with the integrated equations of motion. The algorithm uses the method of least squares to estimate the vector from the center of rotation to the center of mass. The center of mass of the simulator is then moved near to the center of rotation by means of movable masses that are adjusted to the correct location. The algorithm is able move the center of mass to a location closer than two hundredths of a millimeter from the center of rotation. This adjustment increases the period of oscillation of the simulator to more than 60 seconds.

Nomenclature

$\mathbf{I}$	Mass moment of inertia tensor
ω	Angular velocity vector
$\mathbf{r}$	Vector: CR to CM
$\mathbf{M}_{THR}$	Applied moment vector from thrusters
$\mathbf{B}$	Aerodynamic drag coefficient
$\mathbf{\Theta}$	Euler angle vector
ϕ	X Euler angle
θ	Y Euler angle
ψ	Z Euler angle
m	System mass
g	Gravitational acceleration
t	time

Background

Accurate attitude determination and control (ADC) is crucial to the success of satellite missions. However, it is difficult to ensure proper in-flight operation of the ADC system. An option for satellite manufacturers is to test the ADC on the ground. However, the environment on earth is very different from the one in which the satellite will operate. This fact creates challenges in the functional ground testing of a satellite.

Satellite Simulators

A solution to this problem can be found in both hardware and software simulation of the satellite. Satellite simulators are useful to both the satellite industry and educational institutions. They are used for many purposes, including proof-of-concept design in the development of new satellites, pre-flight hardware and software integration and debugging, ADC algorithm testing, research into new ADC algorithms, and the teaching of satellite attitude control.

Exact duplication of the space environment is extremely difficult. The goal with ground-based satellite simulation is to emulate the space environment to the level necessary to appropriately test crucial subsystems. This testing can be performed in several ways. The most common simulation techniques are full software simulation, using simulated electronic inputs as 'sensor' input to the flight computer, and full dynamic hardware testing of the satellite computer and sensors (Kaplan, 1976).

Air Bearings

Air bearings have been found to be an advantageous method for full dynamic testing of the satellite's hardware/software integration. These low friction supports allow for three-axis freedom of motion, within a limited range, and allow accurate duplication of torque-free motion.

There are significant drawbacks with the use of air bearings. One drawback is the existence of gravitational torques. The center of mass of the simulator must be extremely close to the center of rotation of the bearing to minimize the gravitational torque about the center of rotation. Moving the center of mass toward the center of rotation is referred to as 'balancing' the table. The goal of balancing the

SSACS is to increase the period of oscillation as far as possible without causing system instability.

The Full Equations of Motion

The motion of a rigid body around a stationary point other than the CM of the body can be described by

$$\dot{\omega} = (\boldsymbol{A})^{-1}(\boldsymbol{M} - \boldsymbol{B}) \qquad (1)$$

(Young, 1998) where **A, M,** and **B** are defined as

$$\boldsymbol{A} = \begin{bmatrix} mr_y^2 + mr_z^2 + I_{xx} & -mr_x r_y + I_{xy} & -mr_x r_z + I_{xz} \\ -mr_x r_y + I_{xy} & mr_x^2 + mr_z^2 + I_{yy} & -mr_y r_z + I_{yz} \\ -mr_x r_z + I_{xz} & -mr_y r_z + I_{yz} & mr_x^2 + mr_y^2 + I_{zz} \end{bmatrix} \qquad (2)$$

$$\boldsymbol{M} = \begin{pmatrix} M_{Thr_x} \\ M_{Thr_x} \\ M_{Thr_x} \end{pmatrix} + \begin{pmatrix} -B_x \omega_x^2 \\ -B_y \omega_y^2 \\ -B_z \omega_z^2 \end{pmatrix} + \begin{pmatrix} -mgr_y \cos\phi\cos\theta + mgr_z \sin\phi\cos\theta \\ mgr_x \cos\phi\cos\theta + mgr_z \sin\theta \\ -mgr_x \sin\phi\cos\theta - mgr_y \sin\theta \end{pmatrix} \qquad (3)$$

$$\boldsymbol{B} = \begin{pmatrix} [(-2mr_y r_z + I_{zy})\omega_y^2 + (2mr_y r_z - I_{yz})\omega_z^2 \\ + (-mr_x r_z + I_{xz})\omega_x\omega_y + (mr_x r_y - I_{xy})\omega_x\omega_z \\ + (mr_y^2 - mr_z^2 - I_{yy} + I_{zz})\omega_y\omega_z] \\ [(2mr_x r_z - I_{zx})\omega_x^2 + (-2mr_x r_z + I_{xz})\omega_z^2 \\ + (mr_y r_z - I_{zy})\omega_x\omega_y \\ + (-mr_x^2 - mr_z^2 + I_{xx} - I_{zz})\omega_x\omega_z \\ + (-mr_x r_y + I_{xy})\omega_y\omega_z] \\ [(-2mr_x r_y + I_{xy})\omega_x^2 + (2mr_x r_y - I_{xy})\omega_y^2 \\ + (mr_x^2 - mr_y^2 - I_{xx} + I_{yy})\omega_x\omega_y \\ + (-mr_y r_z + I_{yz})\omega_x\omega_z + (-mr_x r_z + I_{xz})\omega_y\omega_z] \end{pmatrix} \qquad (4)$$

These equations can be solved by simultaneously solving equation 1 with

$$\dot{\boldsymbol{\Theta}} = \begin{bmatrix} 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \end{bmatrix} \omega \qquad (5)$$

Equation 5 gives the values for the Euler angles ϕ, θ, and ψ which describe the attitude of the body relative to an inertial frame (Wertz, 1978).

The Simplified Equations of Motion

The equations of motion for the SSACS can be greatly simplified by making a few assumptions. If ω, **r**, the products of inertia, and **B** are assumed to be small compared to the other terms, and $\mathbf{M}_{thr}$ is assumed to be zero, the equations of motion are given by equation 6. This equation, as with the full equations of motion, is completed by the inclusion of equation 5, the Euler rate equations (Wertz, 1978).

$$\dot{\omega} = \begin{bmatrix} \dfrac{mg}{I_{xx}}(-r_y\cos\phi\cos\theta + r_z\sin\phi\cos\theta) \\ \dfrac{mg}{I_{yy}}(r_x\cos\phi\cos\theta + r_z\sin\theta) \\ \dfrac{mg}{I_{zz}}(-r_x\sin\phi\cos\theta - r_y\sin\theta) \end{bmatrix} \qquad (6)$$

Manual Balancing

The common procedure used to balance air-bearing satellite simulators is known as manual balancing. This is a time-consuming, iterative process where lead weights of varying mass are placed on the simulator at various locations in an attempt to balance the table. The process is finalized by carefully adjusting several strategically placed set screws to 'zero-in' the CM to the CR.

This manual method of balancing has advantages and disadvantages. Manual balancing is relatively simple to conceive and execute. It requires only patience and a little skill in weight placement. On the other hand, the amount of time required to balance the SSACS is considerable, often several hours, and the results can be disappointing. After considerable time spent in the manual balancing process, the CM offset is still large enough to create oscillations around the CR with a period of approximately 20 seconds. Using the CM locator algorithm described below, the CM after manual balancing was found to be

$$r_{CM} = (\,-0.0096 \quad 0.0715 \quad -0.495\,)\ mm \tag{7}$$

The magnitude of this vector is 0.50 mm. Using manual balancing, it is extremely difficult to move the CM closer than a half of a millimeter to the CR.

DIAMC Balancing

An improvement over manual balancing is an automatic balancing system developed for the simulator. Known as Dynamic Identification and Adjustment of the Mass Center (DIAMC) balancing, this system automatically calculates the center of mass location and adjusts it a location very close to the center of rotation.

DIAMC balancing is very advantageous, especially when compared to manual balancing. Balancing can occur within ten minutes, move the CM closer than two hundredths of a millimeter to the CR, and requires minimal input from the operator. However, creating the automatic balancing system (including component design and creation, and algorithm derivation and coding) is a difficult and time consuming affair. However, once operational, DIAMC balancing is far more effective than other methods.

Center of Mass Location

The first step in DIAMC balancing is locating the center of mass of the SSACS. To do this, the simplified equations of motion for the angular velocity of the SSACS are used (see equation 6). These equations can be integrated individually over a short time period to give three equations for each time step. This can be done easily with the assumption that θ and ϕ remain relatively constant over a small time step. After integration, the three equations contain only three unknowns, the CM offset distance vector, **r**. The equations after integration can be written

$$(\omega_{y_{t2}} - \omega_{y_{t1}}) = \frac{mg\Delta t}{2I_{yy}}\Big[\big((\cos\phi\cos\theta)_{t2} + (\cos\phi\cos\theta)_{t1}\big)r_x + \big((\sin\theta)_{t2} + (\sin\theta)_{t1}\big)r_z\Big] \tag{8}$$

$$(\omega_{x_{t2}} - \omega_{x_{t1}}) = \frac{-mg\Delta t}{2I_{xx}}\Big[\big((\cos\phi\cos\theta)_{t2} + (\cos\phi\cos\theta)_{t1}\big)r_y - \big((\sin\phi\cos\theta)_{t2} + (\sin\phi\cos\theta)_{t1}\big)r_z\Big] \tag{9}$$

$$(\omega_{z_{t2}} - \omega_{z_{t1}}) = \frac{-mg\Delta t}{2I_{zz}}\Big[\big((\sin\phi\cos\theta)_{t2} + (\sin\phi\cos\theta)_{t1}\big)r_x + \big((\sin\theta)_{t2} + (\sin\theta)_{t1}\big)r_y\Big] \tag{10}$$

Placing these equations into matrix form gives

$$\begin{pmatrix} \Delta\omega_x \\ \Delta\omega_y \\ \Delta\omega_z \end{pmatrix} = \begin{bmatrix} 0 & \Phi_{12} & \Phi_{12} \\ \Phi_{21} & 0 & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & 0 \end{bmatrix} \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \tag{11}$$

The six values for Φ are assumed constant for a given time step. These values can be found by manipulating the expressions found in equations 8, 9, and 10 into the form given in equation 12.

$$\begin{aligned}
\Phi_{12} &= \frac{-mg\Delta t}{2I_{xx}}\big((\cos\phi\cos\theta)_{t2} + (\cos\phi\cos\theta)_{t1}\big) \\
\Phi_{13} &= \frac{mg\Delta t}{2I_{xx}}\big((\sin\phi\cos\theta)_{t2} + (\sin\phi\cos\theta)_{t1}\big) \\
\Phi_{21} &= \frac{mg\Delta t}{2I_{yy}}\big((\cos\phi\cos\theta)_{t2} + (\cos\phi\cos\theta)_{t1}\big) \\
\Phi_{23} &= \frac{mg\Delta t}{2I_{yy}}\big((\sin\theta)_{t2} + (\sin\theta)_{t1}\big) \\
\Phi_{31} &= \frac{-mg\Delta t}{2I_{zz}}\big((\sin\phi\cos\theta)_{t2} + (\sin\phi\cos\theta)_{t1}\big) \\
\Phi_{32} &= \frac{-mg\Delta t}{2I_{zz}}\big((\sin\theta)_{t2} + (\sin\theta)_{t1}\big)
\end{aligned} \tag{12}$$

Equation 11 can also be written in condensed form as,

$$\Delta\Omega = \Phi\, r \tag{13}$$

These equations can be solved for **r** using the method of least squares. Collecting these equations into matrix form, and expanding the matrices for many time steps gives

$$\begin{pmatrix} (\Delta\omega_x)_{t0} \\ (\Delta\omega_y)_{t0} \\ (\Delta\omega_z)_{t0} \\ (\Delta\omega_x)_{t1} \\ (\Delta\omega_y)_{t1} \\ (\Delta\omega_z)_{t1} \\ etc. \end{pmatrix} = \begin{bmatrix} 0 & (\Phi_{12})_{t0} & (\Phi_{12})_{t0} \\ (\Phi_{21})_{t0} & 0 & (\Phi_{23})_{t0} \\ (\Phi_{31})_{t0} & (\Phi_{32})_{t0} & 0 \\ 0 & (\Phi_{12})_{t1} & (\Phi_{12})_{t1} \\ (\Phi_{21})_{t1} & 0 & (\Phi_{23})_{t1} \\ (\Phi_{31})_{t1} & (\Phi_{32})_{t1} & 0 \\ - & etc. & - \end{bmatrix} \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \tag{14}$$

Equation 14 can also be written

$$\Delta\Omega_L = \Phi_L\, r \tag{15}$$

The Ω_L and Φ_L matrices are filled by calculating the six components specified in equation 12 for each time step. The least squares approximation for **r** is given by

$$r = [\Phi_L{}^T\, \Phi_L]^{-1} \Phi_L{}^T\, \Delta\Omega_L \tag{16}$$

<u>Length of Test.</u> An important consideration is the amount of data required to accurately locate the CM. To answer this question, a data set of 60 seconds at 25 hertz was taken from the SSACS. The CM location was calculated using various lengths of data, corresponding to test length. Representative test results for only the Z-axis are shown in Figure 1.

This figure shows the calculated CM location, as well as the standard deviation for the calculation. The figures show that a longer test length results in the algorithm settling on a CM value, and the standard deviation becoming very small.

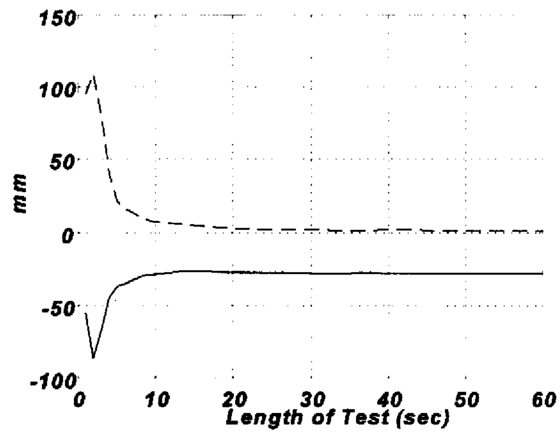

Fig. 1. Z-axis test length experiment. Calculated CM (solid), Corresponding standard deviation on the CM (dashed).

The plots also show that these improvements in accuracy are large at first, and begin to diminish rapidly. After a 60 second test length, changes in accuracy are extremely minimal. For this reason, a 60 second test length is used in the DIAMC balancing algorithm.

<u>Center of Mass Adjustment</u>

Once the center of mass offset vector, **r**, is known, the center of mass can be adjusted using Mass Moving Units (MMU). These MMU were designed and built by Walter Holeman for a previous satellite simulator and were adapted for use with the DIAMC balancing system (Holeman, 1997). The MMU are DC stepper motors mounted so that as the motor rotates, the motor itself is moved along the threaded rod. The motion of the stepper motor causes a change in the center of mass of the table. There are three MMU on the SSACS; each placed on a different axis. By combining the CM changes from each MMU, the CM offset can be minimized from any location as long as the initial offset is relatively close to the CR.

The distance to move the MMU to compensate for the CM offset can be calculated from the general equation that defines the center of mass (Greenwood, 1988). This equation is

$$r = \frac{1}{M} \sum_{i=1}^{n} m_i r_i \tag{22}$$

where $\mathbf{r}$ is as defined previously, M is the total mass of the system, m_i is the mass of each component, and r_i is the distance to the center of mass of each component.

The entire table can be assumed to be four components. The first component is the entire table minus the MMU. The other three components are the MMU. To simplify the derivation, assume that the CM offset is a result only of misplacement of the MMU, or that the table itself has no CM offset, and that the MMU are only misplaced along their respective axes. This allows the center of mass vectors to be written,

$$r_1 = [0,\ 0,\ 0]^T \tag{23}$$

$$r_2 = [r_{m_x},\ 0,\ 0]^T \tag{24}$$

$$r_3 = [0,\ r_{m_y},\ 0]^T \tag{25}$$

$$r_4 = [0,\ 0,\ r_{m_z}]^T \tag{26}$$

where $\mathbf{r}_m$ is the distance to the respective center of mass of the MMU. These assumptions allow equation 22 to be rewritten as

$$\boldsymbol{r} = \frac{1}{M}\begin{pmatrix} m_2 r_{m_x} \\ m_3 r_{m_y} \\ m_4 r_{m_z} \end{pmatrix} \tag{27}$$

The desired quantity is $\Delta\mathbf{r}_m$, the change in MMU position that brings the system CM to zero, or,

$$\Delta\boldsymbol{r}_m = -M\begin{pmatrix} r_x/m_2 \\ r_y/m_3 \\ r_z/m_4 \end{pmatrix} \tag{28}$$

This algorithm is only limited by the distance the MMU can travel. The maximum travel distance for the MMU is 0.076 meters. This correlates to a maximum correctable CM offset of 0.88 mm. This means that the SSACS needs to be manually balanced to within 0.88 mm before any automatic balancing can be effective. Because of the limited knowledge of the physical parameters of the SSACS, this algorithm can be repeated to iteratively place the CM closer to the CR.

The assumption can be made that all of the MMU masses are equal ($m_1=m_2=m_3=m$). Using this assumption, equation 27 becomes,

$$\boldsymbol{r} = \frac{m}{M}\boldsymbol{r}_m \tag{29}$$

and equation 28 becomes,

$$\Delta\boldsymbol{r}_m = -\frac{M}{m}\boldsymbol{r} \tag{30}$$

Balancing Stability

An important issue with the DIAMC balancing algorithm is balancing stability. By this is meant the tendency for the SSACS to stay within its range of motion, free from impact with the bearing or its support. In contrast, an unstable SSACS has the tendency to tip over and come to rest against the bearing support. When in this condition, no dynamic data can be taken, and the balancing process must be started again.

The cause of balancing instability is one of two things. Either the Z-axis CM is above the CR, or the CM is out farther along X-axis or Y-axis than down the negative Z-axis. The lowest energy state for both of these conditions is beyond the range of motion of the SSACS

The CM along the Z-axis may not cross into the positive range without a loss of stability. To insure that the Z center of mass remains in the stable area, the $\Delta\mathbf{r}_m$ value in multiplied by 0.6. This also minimizes the chance that the CM will be moved out farther along the X or Y-axis than the value of the current Z-axis.

DIAMC Balancing Effectiveness

The effectiveness of the DIAMC balancing algorithm can best be measured by the reduction in the oscillatory period of the motion of the SSACS. Because of the incomplete knowledge of several of the physical parameters, most notably the inertia tensor and MMU placement, the DIAMC balancing algorithm works best when repeated several times. Experience using the DIAMC balancing algorithm in concert with the actual SSACS shows that three iterations of the balancing algorithm yield the best results. This conclusion is supported by figure 2 and figure 3. These plots show that the CM distances shrink drastically through the two iterations shown. Additional iterations do not achieve any better results. In fact, additional iterations are likely to cause balancing stability problems.

Before starting the autobalancer, the table was balanced as well as possible manually. The CM was then located using the DIAMC balancing algorithm. The results can be seen in figure 2. The CM for each axis was within 0.5 mm of the CR. The period of oscillation for this CM is approximately 20 seconds, as shown in figure 3. The MMU were then moved according to the recommendations of the DIAMC balancing algorithm. The results can be seen in figure 2. By moving the MMU, the CM has been moved dramatically closer to the CR. The results of this CM change can be clearly seen in figure 3. The period of motion has been increased to approximately 35 seconds. To improve the response further, the process was repeated once again. This final iteration brought the CM for all three axes to within 0.1 mm of the CM and the period of motion has been increased to 60 seconds. This data is shown in table 1.

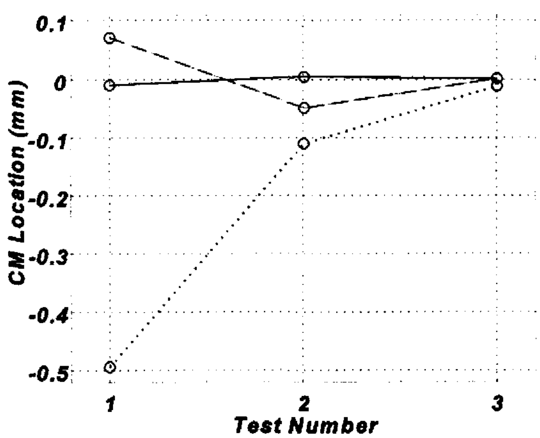

Fig. 2. Center of mass location measurements: Trial 1. X-axis (solid), Y-axis(dashed), and Z-axis (dotted). Test 1 is before any MMU movement, test 2 is after one iteration, and Test 3 is after two iterations.

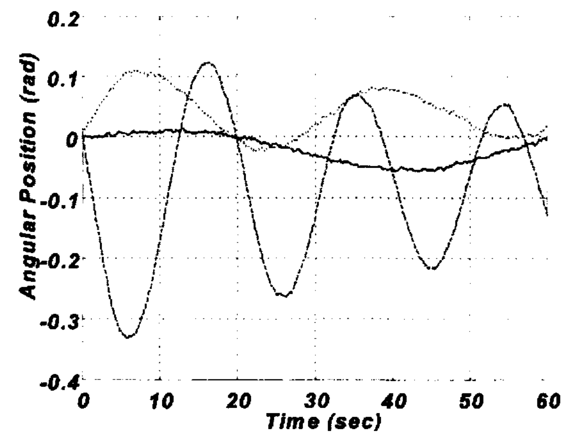

Fig. 3. X-axis tilt data for DIAMC balancing test: Trial 1. Test 1 (dashed), Test 2 (dotted), Test 3 (solid).

These figures show how well the DIAMC balancing algorithm works: The CM can be moved to within two hundredths of a millimeter from the CM, and the oscillations caused by the CM offset can be slowed to 3 times their original value (see table 1).

Table 1. Period of Oscillation during DIAMC Balancing

	Manual Balance (sec)	Iteration 1 (sec)	Iteration 2 (sec)	Percent Improve-ment
Trial 1	20	35	60	300
Trial 2	22	40	60	275

<u>Comparison to a Simple Pendulum.</u> The period of a simple one-degree of freedom pendulum with the small angle assumption can be found from

$$T = \frac{2\pi}{\sqrt{\frac{Mgr}{I_{XX}}}} \tag{31}$$

(Greenwood, 1988) where M, g, and I_{XX} are used as previously defined. The value of r is the distance from the point where the pendulum attaches to the ceiling to the oscillating mass. This period value can be compared to the period shown by the SSACS using the $\mathbf{r}_{CM}$ values from Trial 1 of the DIAMC balancing effectiveness test. The estimated simple pendulum values and the observed values from the SSACS can be seen in table 2. The values correspond well. The large difference between the simple pendulum and the actual SSACS at Iteration 3 can be explained by the fact that the SSACS is not a simple, frictionless pendulum. The SSACS is a complex three-axis 'pendulum' with the ability to transfer energy between the axes as well as non-linear friction at the air-bearing.

The periods from the initial test and from iteration 1 correspond very well. This comparison provides more evidence that the DIAMC balancing algorithm is able to find an accurate CM estimate and correct for the offset.

Table 2. Comparison of a Simple Pendulum to the SSACS X-Axis

	r_{CM} (mm)	SSACS X-Axis Period (Observed) (sec)	1-Axis Pendulum Period (Calculated) (sec)
Initial Test	0.500	20	21
Iteration 1	0.120	35	42
Iteration 2	0.014	60	132

References

Greenwood, Donald. Principles of Dynamics. 2nd ed., New Jersey: Prentice-Hall, Inc., 1988.

Holeman, Walter. "Determination and Control of the Orientation of the Principal Axes of Inertia of Spinning Bodies." Masters Thesis, Utah State University, 1997.

Kaplan, Marshall. Modern Spacecraft Dynamics & Control. New York: John Wiley & Sons, 1976.

Wertz, James, ed. Spacecraft Attitude Determination and Control. Dordrecht: Kluwer Academic Publishers, 1978.

Young, Jeff. "Development of an Automatic Balancing System for a Small Satellite Attitude Control Simulator." Masters Thesis, Utah State University, 1998.