NASA TECHNICAL NOTE

NASA TN D-7067

NASA TN D-7067

AN ACTUATOR EXTENSION TRANSFORMATION FOR A MOTION SIMULATOR AND AN INVERSE TRANSFORMATION APPLYING NEWTON-RAPHSON'S METHOD

by James E. Dieudonne, Russell V. Parrish,
and Richard E. Bardusch

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • NOVEMBER 1972

1. Report No. NASA TN D-7067	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN ACTUATOR EXTENSION TRANSFORMATION FOR A MOTION SIMULATOR AND AN INVERSE TRANSFORMATION APPLYING NEWTON-RAPHSON'S METHOD		5. Report Date November 1972
		6. Performing Organization Code
7. Author(s) James E. Dieudonne, Russell V. Parrish, and Richard E. Bardusch		8. Performing Organization Report No. L-8505
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		10. Work Unit No. 501-39-11-02
		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A set of equations which transform position and angular orientation of the centroid of the payload platform of the six-degree-of-freedom motion simulator at the Langley Research Center into extensions of the simulator's actuators has been derived and is based on a geometrical representation of the system. An iterative scheme, Newton-Raphson's method, has been successfully used in a real-time environment in the calculation of the position and angular orientation of the centroid of the payload platform when the magnitude of the actuator extensions is known. Sufficient accuracy is obtained by using only one Newton-Raphson iteration per integration step of the real-time environment.

17. Key Words (Suggested by Author(s)) Motion simulator Newton-Raphson method Real-time simulation	18. Distribution Statement Unclassified – Unlimited

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 20	22. Price* $3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

AN ACTUATOR EXTENSION TRANSFORMATION FOR A MOTION SIMULATOR AND AN INVERSE TRANSFORMATION APPLYING NEWTON-RAPHSON'S METHOD

By James E. Dieudonne, Russell V. Parrish,
and Richard E. Bardusch
Langley Research Center

SUMMARY

A set of equations which transform position and angular orientation of the centroid of the payload platform of the six-degree-of-freedom motion simulator at the Langley Research Center into extensions of the simulator's actuators has been derived and is based on a geometrical representation of the system. An iterative scheme, Newton-Raphson's method, has been successfully used in a real-time environment in the calculation of the position and angular orientation of the centroid of the payload platform when the magnitude of the actuator extensions is known. Sufficient accuracy is obtained by using only one Newton-Raphson iteration per integration step of the real-time environment.

INTRODUCTION

To enhance the capability of producing realistic aircraft simulations, in December 1971, NASA Langley Research Center acquired a Singer Simulation Products Division six-degree-of-freedom motion simulator. This paper describes two problems inherent to the design of this particular motion simulator and the methods used to solve these problems. First, since the motion simulator does not have independent drive systems for each degree of freedom but achieves motion in all degrees of freedom by a combination of actuator extensions, the base cannot be driven with the usual inputs of position and angular orientation of the centroid of the payload platform. This fact requires that a transformation be developed which will convert motion cues into actuator extensions. This transformation will be discussed first in this paper.

For the purpose of evaluating the performance of the simulator hardware, an inverse transformation is desirable. The actual position and angular orientation of the centroid of the payload platform would be obtained from the transformation of the actual magnitudes of the actuators. A comparison of desired position and orientation of the centroid with actual position and orientation would then yield the error in the base drive and provide a method of determining dynamic servo performance. This inverse transformation is also required for optimal washout filter design; therefore, the development of a method of computing this inverse transformation that will operate in a real-time environment is presented.

SYMBOLS

Values are given in both SI and U.S. Customary Units. The measurements and calculations were made in U.S. Customary Units.

$\bar{A}_i$ vector in fixed coordinate system from origin of moving coordinate system to upper attachment point of actuator i

$\bar{A}_{i,m}$ known vector in moving coordinate system from origin of moving coordinate system to upper attachment point of actuator i

$a_{i,m_1},a_{i,m_2},a_{i,m_3}$ elements of $\bar{A}_{i,m}$

$\bar{B}_i$ known vector in fixed coordinate system from origin of fixed coordinate system to lower attachment point of actuator i

b_{i_1},b_{i_2},b_{i_3} elements of $\bar{B}_i$

E_i extension of actuator i

$\bar{f}(\ )$ vector function

H height above lower bearing plane

i_m,j_m,k_m components of unit vectors defined in moving coordinate system

i_o,j_o,k_o components of unit vectors defined in fixed coordinate system

$\bar{\ell}_i$ vector in fixed coordinate system from lower attachment point to upper attachment point of actuator i

$\ell_{i,x},\ell_{i,y},\ell_{i,z}$ components of $\bar{\ell}_i$

$|\bar{\ell}_i|_a$ actual magnitude of actuator i obtained from instrumentation

$\lvert\bar{\ell}_i\rvert_{neut}$	magnitude of actuator i when payload platform is in its neutral position
$\bar{O}$	null vector
$\bar{R}$	vector in fixed coordinate system from origin of fixed coordinate system to origin of moving coordinate system
$\bar{r}_i$	vector in fixed coordinate system from origin of fixed coordinate system to upper attachment point of actuator i
$[T]$	Euler angle of transformation matrix
T_{ij}	elements of T matrix
x,y,z	elements of $\bar{R}$ which determine the inertial position of payload platform
$\bar{\alpha}$	unknown parameter vector which is vector root of $\bar{f}(\ )$
ψ,θ,ϕ	inertial angular orientation of payload platform

Subscripts:

m	moving coordinate system
max	maximum
min	minimum
n	past value
n+1	new value
neut	neutral

Superscript:

T	transpose

A bar over a symbol denotes a vector.

MOTION BASE DESCRIPTION

The motion base (fig. 1) consists of a payload platform upon which a cockpit and visual display will be mounted. The payload platform is driven by six hydraulically powered position servomechanisms forming a synergistic six-degree-of-freedom motion system. As shown in figure 2, the points where the actuators connect to the payload platform form an equilateral triangle approximately 3.66 m (144 in.) on a side. The actuators have a minimum length of 2.62 m (103 in.) and a maximum length of 4.14 m (163 in.) giving a fully settled height of the platform of approximately 2.05 m (81 in.) and a fully raised height of 3.99 m (157 in.) measured from the floor. The system design allows for a 6804 kg (15 000 lb) payload and provides the limits of performance shown in table I from the neutral position.

ACTUATOR EXTENSION TRANSFORMATION

Because of the design concept of the hardware, activation of all six hydraulic actuators is generally required for motion in each degree of freedom. Thus, the simulator must be driven with actuator extensions, but since aircraft simulations provide position and angular orientation of the centroid of the payload platform, a transformation to actuator extensions is necessary.

Two coordinate systems are defined as shown in figure 3; one has its origin at the centroid of the lower fixed platform of the simulator and the other, at the centroid of the moving payload platform. The coordinate systems are both right-hand systems and their axes are alined when the payload platform is at its neutral position. When one actuator at a time is considered, figure 4 defines the vector relationships between the origins of each coordinate system and the actuator attachment point of each platform. Figure 5 shows these relationships for a general orientation of the motion simulator. These relationships yield the vector equations

$$\left.\begin{aligned}\bar{r}_i &= \overline{A}_i + \overline{R} \\ \bar{r}_i &= \overline{B}_i + \bar{\ell}_i\end{aligned}\right\} \tag{1}$$

which after subtraction become

$$\bar{\ell}_i = \overline{A}_i + \overline{R} - \overline{B}_i \tag{2}$$

This equation is defined with respect to the fixed reference frame. Since the known vectors $\overline{A}_{i,m}$ are defined with respect to the moving frame, one must apply an Euler angle transformation in order to determine the corresponding vectors $\overline{A}_i$ in the fixed reference frame. By using a ψ,θ,ϕ order of rotation, the transformation is

$$\begin{bmatrix} i_m \\ j_m \\ k_m \end{bmatrix} = \begin{bmatrix} T \end{bmatrix} \begin{bmatrix} i_o \\ j_o \\ k_o \end{bmatrix}$$

where

$$\begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}$$

and

$$T_{11} = \cos\psi \cos\theta \qquad T_{12} = \sin\psi \cos\theta \qquad T_{13} = -\sin\theta$$

$$T_{21} = \cos\psi \sin\theta \sin\phi - \sin\psi \cos\phi \qquad T_{22} = \sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi \qquad T_{23} = \cos\theta \sin\phi$$

$$T_{31} = \cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi \qquad T_{32} = \sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi \qquad T_{33} = \cos\theta \cos\phi$$

Applying this transformation to the $\bar{A}_{i,m}$ vectors yields

$$\bar{A}_i = \begin{bmatrix} T \end{bmatrix}^T \bar{A}_{i,m}$$

Substituting this relation into equation (2) yields

$$\bar{\ell}_i = \begin{bmatrix} T \end{bmatrix}^T \bar{A}_{i,m} + \bar{R} - \bar{B}_i \tag{3}$$

which can now be solved since $\bar{A}_{i,m}$ and $\bar{B}_i$ are known constant vectors, and $\bar{R}$ and $\begin{bmatrix} T \end{bmatrix}^T$ are calculated from the given values of x, y, z, ψ, θ, and ϕ. The length of each actuator is then

$$\left| \bar{\ell}_i \right| = \sqrt{\ell_{i,x}^2 + \ell_{i,y}^2 + \ell_{i,z}^2}$$

and the actuator extensions E_i are defined by

$$E_i = |\bar{\ell}_i| - |\bar{\ell}_i|_{neut}$$

where $|\bar{\ell}_i|_{neut}$ is the known neutral position value of actuator i.

INVERSE TRANSFORMATION

The actuator extension transformation permits the driving of the motion simulator with position and angular orientation inputs, but a means for determining the simulator's response to these signals and its positioning accuracy was unavailable. A solution would be to take the actual actuator extensions, available from potentiometers mounted on each actuator, and transform them into the position and angular orientation of the centroid of the payload platform. Actual error, phase lag, and so forth, could then be determined from comparisons of the commanded and actual position and orientation. This transformation is simply the inverse of the actuator extension transformation. This inverse could be obtained if the vectors $\bar{\ell}_i$ were available, but the potentiometer reading yields only the magnitude of the corresponding actuator $|\bar{\ell}_i|_a$ and not the required vector. With only this information available, the problem becomes that of solving six simultaneous nonlinear equations for six unknowns (x, y, z, ψ, θ, and ϕ).

The approach was to apply an iterative numerical method known as the Newton-Raphson method. (See ref. 1 (pp. 447-453) and ref. 2 (pp. 19-26).) For a vector-matrix equation, this method is a general method of computing the vector root $\bar{\alpha}$ of equation (4)

$$\bar{f}(\bar{\alpha}) = \bar{O} \tag{4}$$

The iteration formula has the form

$$\bar{\alpha}_{n+1} = \bar{\alpha}_n - \left[\frac{\partial\bar{f}(\bar{\alpha}_n)}{\partial\bar{\alpha}_n}\right]^{-1} \bar{f}(\bar{\alpha}_n) \tag{5}$$

In order to apply this method to the problem, a function satisfying equation (4) must first be defined. Since the actual magnitudes of the actuators are known, a function $f_i(\bar{\alpha})$ can be defined as

$$f_i(\bar{\alpha}) = \bar{\ell}_i^T\bar{\ell}_i - |\bar{\ell}_i|_a^2 \tag{6}$$

where

$$\bar{\alpha} = \begin{bmatrix} x \\ y \\ z \\ \psi \\ \theta \\ \phi \end{bmatrix}$$

$$\bar{f}(\bar{\alpha}) = \begin{bmatrix} f_1(\bar{\alpha}) \\ f_2(\bar{\alpha}) \\ \cdot \\ \cdot \\ \cdot \\ f_6(\bar{\alpha}) \end{bmatrix}$$

and $|\bar{\ell}_i|_a$ is the actual magnitude of actuator i obtained from instrumentation. Substitution of equation (3) into equation (6) and performing the vector multiplication yields equation (7). Note that each vector multiplication results in a scalar and permits the grouping of terms as follows:

$$\begin{aligned} f_i(\bar{\alpha}) &= \left([T]^T\bar{A}_{i,m} + \bar{R} - \bar{B}_i\right)^T\left([T]^T\bar{A}_{i,m} + \bar{R} - \bar{B}_i\right) - |\bar{\ell}_i|_a^2 \\ &= \bar{A}_{i,m}^T[T][T]^T\bar{A}_{i,m} + \bar{A}_{i,m}^T[T]\bar{R} - \bar{A}_{i,m}^T[T]\bar{B}_i + \bar{R}^T[T]^T\bar{A}_{i,m} \\ &\quad + \bar{R}^T\bar{R} - \bar{R}^T\bar{B}_i - \bar{B}_i^T[T]^T\bar{A}_{i,m} + \bar{B}_i^T\bar{B}_i - \bar{B}_i^T\bar{R} - |\bar{\ell}_i|_a^2 \\ &= \bar{A}_{i,m}^T\bar{A}_{i,m} + 2\bar{R}^T[T]^T\bar{A}_{i,m} - 2\bar{B}_i^T[T]^T\bar{A}_{i,m} - 2\bar{R}^T\bar{B}_i \\ &\quad + \bar{R}^T\bar{R} + \bar{B}_i^T\bar{B}_i - |\bar{\ell}_i|_a^2 \end{aligned} \tag{7}$$

By expanding in terms of elements, equation (7) becomes

$$
\begin{aligned}
f_i(\overline{\alpha}) = {} & a^2_{i,m_1} + a^2_{i,m_2} + a^2_{i,m_3} + b^2_{i1} + b^2_{i_2} + b^2_{i_3} + x^2 + y^2 + z^2 - \left|\bar{\ell}_i\right|^2_a \\
& + 2\left(x - b_{i_1}\right)\left(a_{i,m_1}T_{11} + a_{i,m_2}T_{21} + a_{i,m_3}T_{31}\right) \\
& + 2\left(y - b_{i_2}\right)\left(a_{i,m_1}T_{12} + a_{i,m_2}T_{22} + a_{i,m_3}T_{32}\right) \\
& + 2\left(z - b_{i_3}\right)\left(a_{i,m_1}T_{13} + a_{i,m_2}T_{23} + a_{i,m_3}T_{33}\right) \\
& - 2\left(xb_{i_1} + yb_{i_2} + zb_{i_3}\right)
\end{aligned}
\tag{8}
$$

Taking the partial derivatives of $f_i(\overline{\alpha})$ with respect to the elements of $\overline{\alpha}$ yields

$$
\frac{\partial f_i(\overline{\alpha})}{\partial x} = 2\left(x + a_{i,m_1}T_{11} + a_{i,m_2}T_{21} + a_{i,m_3}T_{31} - b_{i_1}\right) \tag{9a}
$$

$$
\frac{\partial f_i(\overline{\alpha})}{\partial y} = 2\left(y + a_{i,m_1}T_{12} + a_{i,m_2}T_{22} + a_{i,m_3}T_{32} - b_{i_2}\right) \tag{9b}
$$

$$
\frac{\partial f_i(\overline{\alpha})}{\partial z} = 2\left(z + a_{i,m_1}T_{13} + a_{i,m_2}T_{23} + a_{i,m_3}T_{33} - b_{i_3}\right) \tag{9c}
$$

$$
\begin{aligned}
\frac{\partial f_i(\overline{\alpha})}{\partial \psi} = {} & -2\left(x - b_{i_1}\right)\left(a_{i,m_1}T_{12} + a_{i,m_2}T_{22} + a_{i,m_3}T_{32}\right) \\
& + 2\left(y - b_{i_2}\right)\left(a_{i,m_1}T_{11} + a_{i,m_2}T_{21} + a_{i,m_3}T_{31}\right)
\end{aligned}
\tag{9d}
$$

$$\frac{\partial f_i\left(\bar{\alpha}\right)}{\partial \theta} = 2\left(x - b_{i_1}\right)\left(-a_{i,m_1} \sin\theta \cos\psi + a_{i,m_2} \sin\phi \cos\theta \cos\psi + a_{i,m_3} \cos\phi \cos\theta \cos\psi\right) + 2\left(y - b_{i_2}\right)\left(-a_{i,m_1} \sin\theta \sin\psi + a_{i,m_2} \sin\phi \cos\theta \sin\psi + a_{i,m_3} \cos\phi \cos\theta \sin\psi\right) -2\left(z - b_{i_3}\right)\left(a_{i,m_1} \cos\theta + a_{i,m_2} \sin\phi \sin\theta + a_{i,m_3} \cos\phi \sin\theta\right) \qquad (9e)$$

$$\frac{\partial f_i\left(\bar{\alpha}\right)}{\partial \phi} = 2\left(x - b_{i_1}\right)\left(a_{i,m_2} T_{31} - a_{i,m_3} T_{21}\right) + 2\left(y - b_{i_2}\right)\left(a_{i,m_2} T_{32} - a_{i,m_3} T_{22}\right) + 2\left(z - b_{i_3}\right)\left(a_{i,m_2} T_{33} - a_{i,m_3} T_{23}\right) \qquad (9f)$$

The solution to the problem is now complete, since for a given set of initial conditions of $\bar{\alpha}$ and corresponding solutions to equations (8) and (9), equation (5) can be solved. The final form of the solution is

$$\begin{bmatrix} x \\ y \\ z \\ \psi \\ \theta \\ \phi \end{bmatrix}_{n+1} = \begin{bmatrix} x \\ y \\ z \\ \psi \\ \theta \\ \phi \end{bmatrix}_{n} - \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \psi} & \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \phi} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \cdot & \cdot & \cdot & \cdot \end{bmatrix}_{n} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}_{n} \qquad (10)$$

where $\left[\ \right]_n$ is defined as a function of past value of $\overline{\alpha}$, $\overline{\alpha}_n$; $\left[\ \right]_{n+1}$ is defined as the new value of $\overline{\alpha}$, $\overline{\alpha}_{n+1}$; and f_i implies $f_i(\overline{\alpha})$. With an appropriate set of initial conditions, equation (10) is repeated until some predetermined convergence criterion (desired accuracy) is met.

The fact that the inverse transformation must operate efficiently in a real-time environment places additional constraints on the solution of the inverse problem. Besides assisting in the validation of hardware performance, the inverse transformation is to be used in the feedback loop for optimal washout techniques. Figure 6 illustrates the major parts of a real-time motion simulation by using a washout technique requiring feedback. As shown in the figure, the motion simulation requires the addition of all the starred blocks to a normal fixed-base simulation. These additions will increase the computational time required per integration step for all motion simulations. The inverse transformation, being an iterative method, requires a variable amount of computational time based on the number of Newton-Raphson iterations; therefore, the minimum number of iterations required to give the desired accuracy had to be determined.

RESULTS

Actuator Extension Transformation

The actuator extension transformation was shown to be valid by the use of several independent checks.

Inverse Transformation

To operate efficiently in a real-time digital environment, the inverse transformation must yield the desired accuracy in the minimum amount of computational time possible. Since most of the real-time digital simulation programs at the Langley Research Center utilize an integration step size of 1/32 second, the inputs to the inverse transformation $|\bar{\ell}_i|_a$ will be changing every 1/32 second. Based on the maximum servo drive rates of the motion base and the sampling rate of the actuator magnitudes, the minimum number of Newton-Raphson iterations per sample required to give the desired accuracy had to be determined. The desired accuracy was defined to be anything within the position specifications of the servo drives of the motion base.

The data necessary to determine the minimum number of Newton-Raphson iterations acceptable were obtained in the following manner: The actuator extension transformation was initialized at a particular starting point and an input to one degree of freedom was swept through the allowable range of that degree of freedom with the maximum allowable rate (determined from the maximum servo drive rates). The resulting actuator extensions were used to drive the inverse transformation; thus, a motion base with perfect response was simulated. Any differences that existed between the inputs to the actuator extension transformation and the outputs of the inverse transformation would then be attributable to the number of Newton-Raphson iterations.

Although many different starting points were tried, no attempt was made to conduct a Monte Carlo simulation of the infinite number of starting points available. The results to be presented represent the worst case encountered.

Table II presents the maximum error obtained in any degree of freedom by using one, two, and three iterations for an input in each individual degree of freedom. For example, with only the x-input changing (at its maximum allowable rate), the maximum error that occurred during the sweep of the x-range was in z, 6.667×10^{-5} m (2.624×10^{-3} in.) for one iteration. The results demonstrate that with one iteration the maximum errors obtained, 9.287×10^{-5} m (3.655×10^{-3} in.) in translation and 2.719×10^{-5} rad (1.558×10^{-3} deg) in rotation, are within the specifications of the servo drives, 3.074×10^{-3} m (0.121 in.) in translation and 8.726×10^{-4} rad (0.05°) in rotation.

Table III shows the maximum error obtained in each degree of freedom, by using one iteration, for maximum rate inputs in individual degrees of freedom. Maximum errors for one-half and one-fourth maximum rates are also shown along with maximum error obtained from maximum rate in all six degrees of freedom simultaneously.

It is interesting to note that the maximum error in the translational degrees of freedom, regardless of the input, always occurs in z. From a comparison of these runs, it can be determined that the error is directly proportional to rate of drive and that even the error incurred from the physically unobtainable case of maximum rate in all degrees of freedom simultaneously is still within the required servo drive specifications.

It should be noted that the addition of the inverse transformation with one Newton-Raphson iteration to a real-time digital simulation program will increase the total computational time per integration step by 0.0018 second and the total memory by 1264 octal core locations.

CONCLUDING REMARKS

A transformation which provides actuator extensions for a six-degree-of-freedom motion base at the Langley Research Center from the motion cues of a real-time digital aircraft simulation has been developed and implemented.

An inverse transformation which provides position and angular orientation of the centroid of the payload platform from the measured magnitude of the actuator extensions has been developed by using a Newton-Raphson technique. The transformation has been shown to yield the required accuracy with one Newton-Raphson iteration for successful operation in a real-time environment, with only a slight increase in memory and computational time.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., October 19, 1972.

REFERENCES

1. Hildebrand, F. B.: Introduction to Numerical Analysis. McGraw-Hill Book Co., Inc., 1956.

2. Fröberg, Carl-Erik: Introduction to Numerical Analysis. Addison-Wesley Pub. Co., Inc., c.1965.

TABLE I.- PERFORMANCE LIMITS

Degree of freedom	Performance limits		
	Position	Velocity	Acceleration
Longitudinal (x)	Forward 1.245 m Aft 1.219 m	±0.610 m/sec	±0.6g
Lateral (y)	Left 1.219 m Right 1.219 m	±0.610 m/sec	±0.6g
Vertical (z)	Up 0.991 m Down .762 m	±0.610 m/sec	±0.8g
Yaw (ψ)	±0.559 rad	±0.262 rad/sec	±0.873 rad/sec^2
Pitch (θ)	±0.524 rad -.349 rad	±0.262 rad/sec	±0.873 rad/sec^2
Roll (ϕ)	±0.384 rad	±0.262 rad/sec	±0.873 rad/sec^2

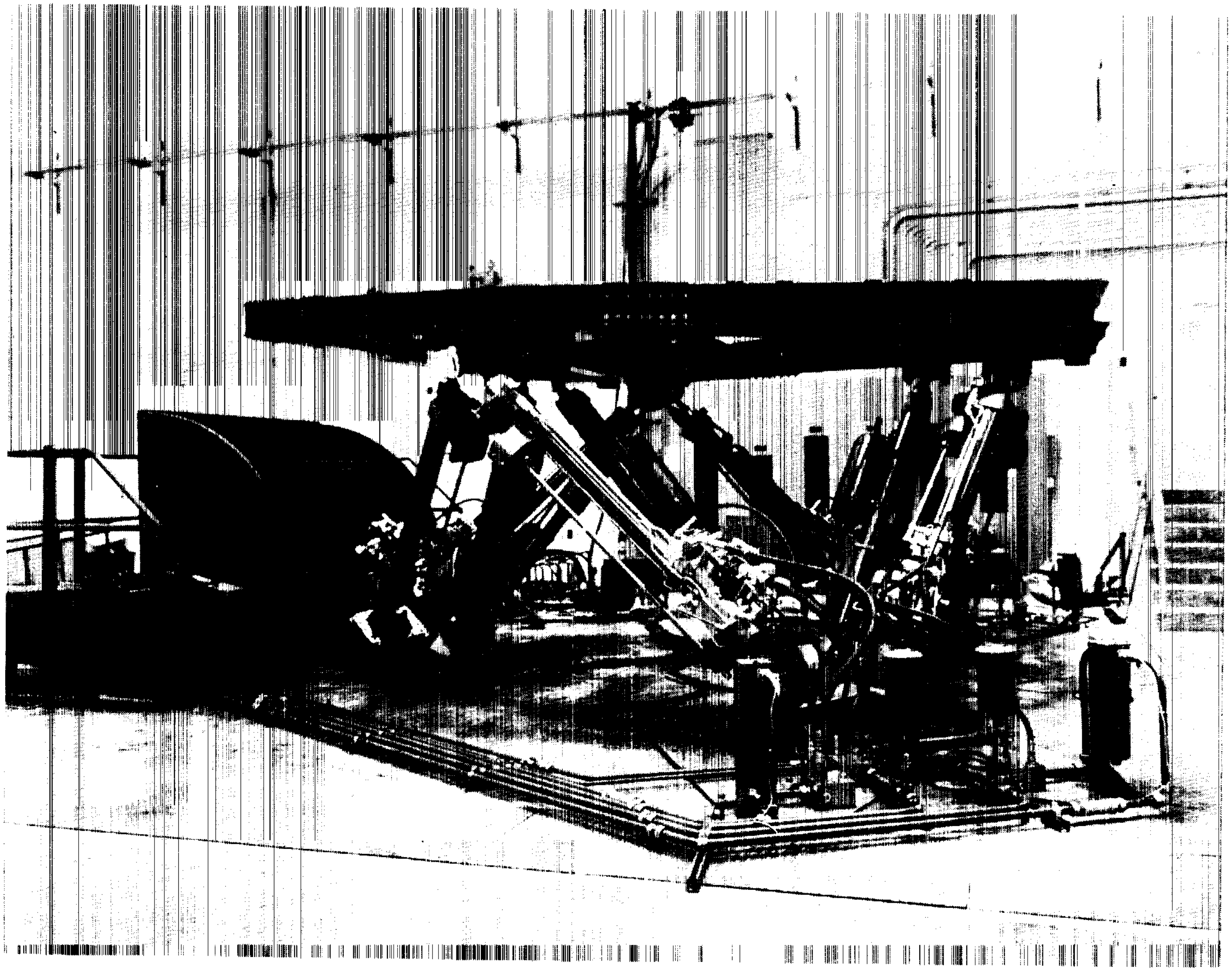
TABLE II.- MAXIMUM ERROR OBTAINED FROM NEWTON-RAPHSON'S ALGORITHM

Degree of freedom of input	Maximum error after –		
	1st iteration (*)	2d iteration (*)	3d iteration (*)
x	(z) 6.667×10^{-5} m	(z) 8.160×10^{-10} m	(z) 2.925×10^{-14} m
y	(z) 6.667×10^{-5} m	(z) 8.160×10^{-10} m	(z) 2.415×10^{-14} m
z	(z) 9.287×10^{-5} m	(z) 3.183×10^{-10} m	(z) 4.621×10^{-14} m
ψ	(ψ) 1.926×10^{-5} rad	(ψ) 1.109×10^{-10} rad	(ψ) 9.920×10^{-15} rad
θ	(θ) 2.119×10^{-5} rad	(θ) 8.980×10^{-11} rad	(θ) 1.984×10^{-15} rad
ϕ	(ϕ) 2.719×10^{-5} rad	(ϕ) 9.245×10^{-10} rad	(ϕ) 1.745×10^{-17} rad

*Quantity in parentheses denotes the degree of freedom in which maximum error occurred.

TABLE III.- MAXIMUM ERROR IN EACH DEGREE OF FREEDOM FOR INPUT OF MAXIMUM ALLOWABLE RATE

Degree of freedom of input	Maximum error in each degree of freedom (1st iteration)					
	x, m	y, m	z, m	ψ, rad	θ, rad	ϕ, rad
x (Maximum rate)	4.621×10^{-14}	1.030×10^{-14}	6.667×10^{-5}	3.633×10^{-15}	1.326×10^{-14}	4.992×10^{-15}
y (Maximum rate)	2.425×10^{-14}	2.166×10^{-14}	6.667×10^{-5}	7.642×10^{-15}	8.774×10^{-15}	9.160×10^{-15}
z (Maximum rate)	5.353×10^{-14}	1.271×10^{-14}	9.287×10^{-5}	3.208×10^{-15}	9.229×10^{-15}	9.171×10^{-15}
ψ (Maximum rate)	5.516×10^{-11}	3.202×10^{-11}	4.725×10^{-5}	1.926×10^{-5}	2.218×10^{-11}	7.171×10^{-13}
θ (Maximum rate)	6.422×10^{-5}	1.015×10^{-14}	4.089×10^{-5}	2.566×10^{-15}	2.119×10^{-5}	3.860×10^{-15}
ϕ (Maximum rate)	6.561×10^{-5}	1.850×10^{-5}	2.043×10^{-5}	1.753×10^{-6}	6.070×10^{-6}	2.719×10^{-5}
All (Maximum rate)	1.471×10^{-4}	1.212×10^{-4}	4.172×10^{-4}	1.440×10^{-4}	1.912×10^{-4}	1.964×10^{-4}
All (1/2 rate)	3.609×10^{-5}	3.017×10^{-5}	1.041×10^{-4}	3.530×10^{-5}	4.702×10^{-5}	4.821×10^{-5}
All (1/4 rate)	9.044×10^{-6}	7.550×10^{-6}	2.619×10^{-5}	8.841×10^{-6}	1.178×10^{-5}	1.205×10^{-5}



L-71-9223

Figure 1.- Six-degree-of-freedom motion simulator.

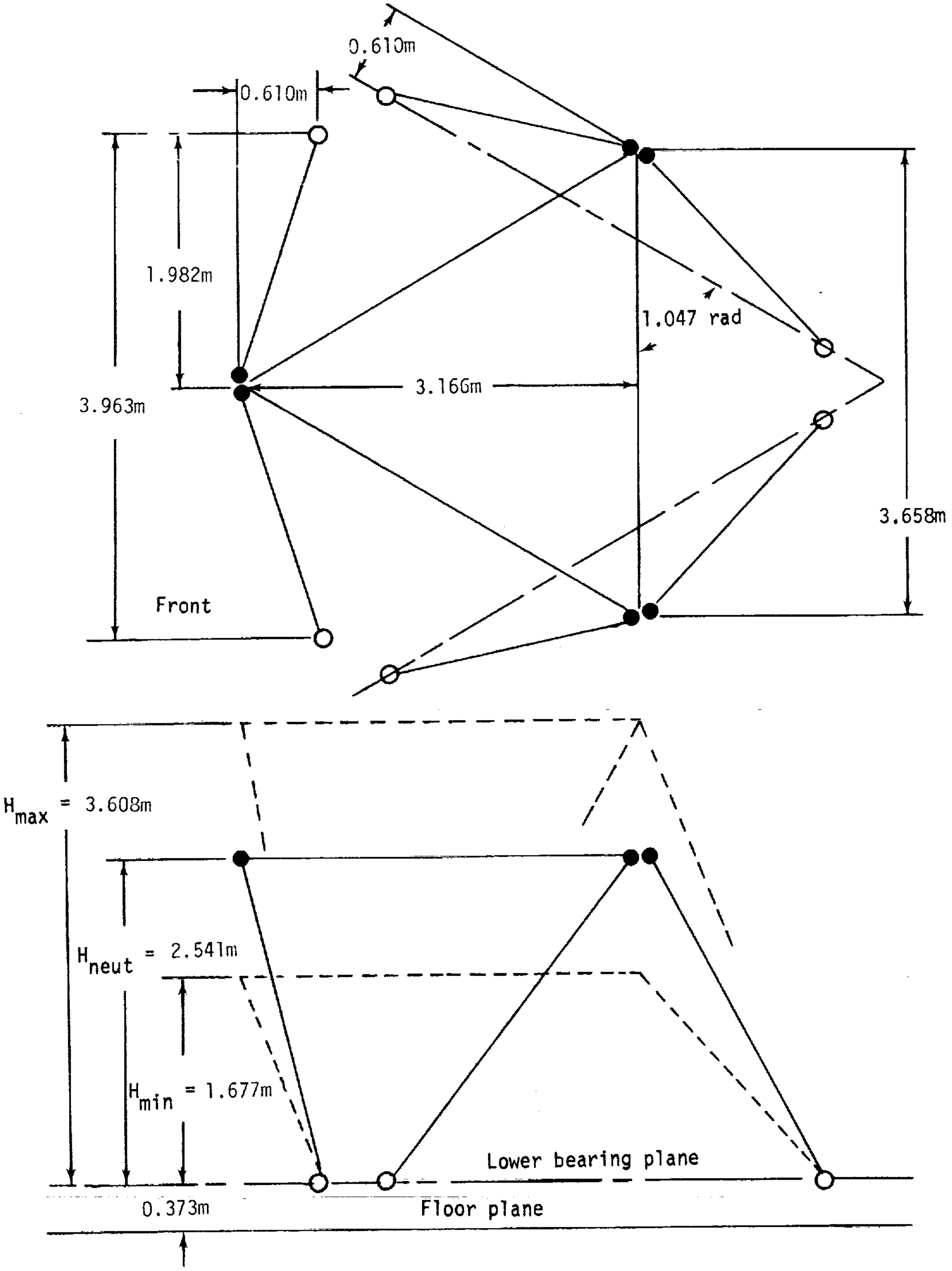


Figure 2.- Motion system in neutral, settled, and raised positions. Actuator dimensions: Minimum length, 2.62 m; maximum length, 4.14 m.

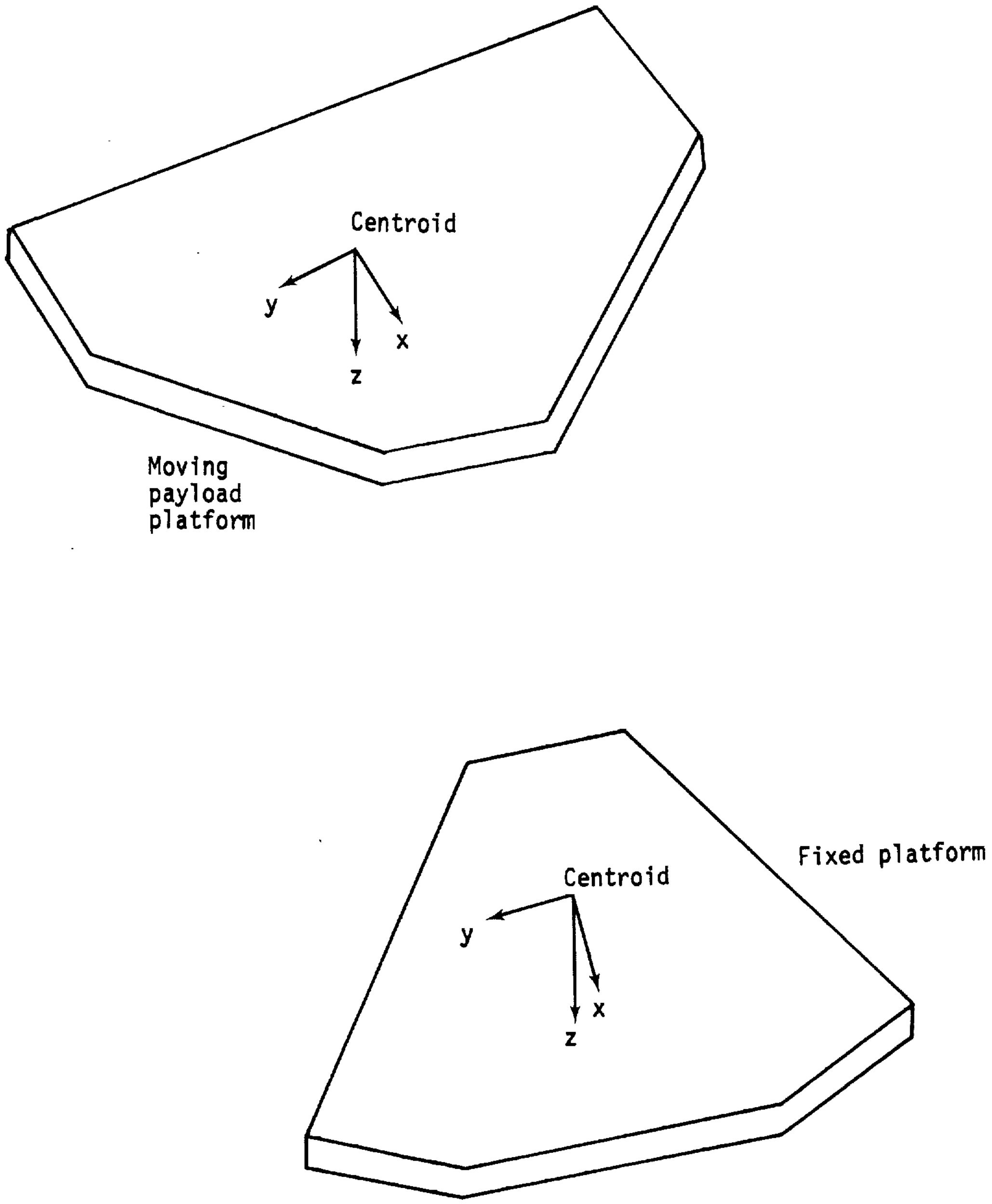


Figure 3.- Coordinate systems.

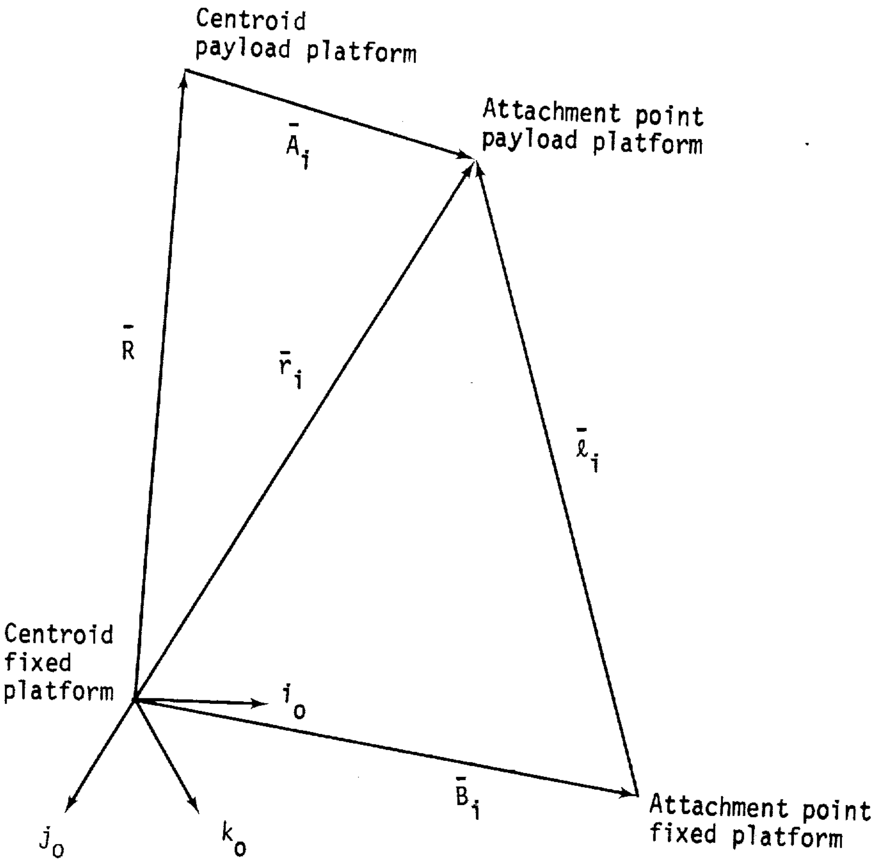


Figure 4.- Vector relationships for actuator i.

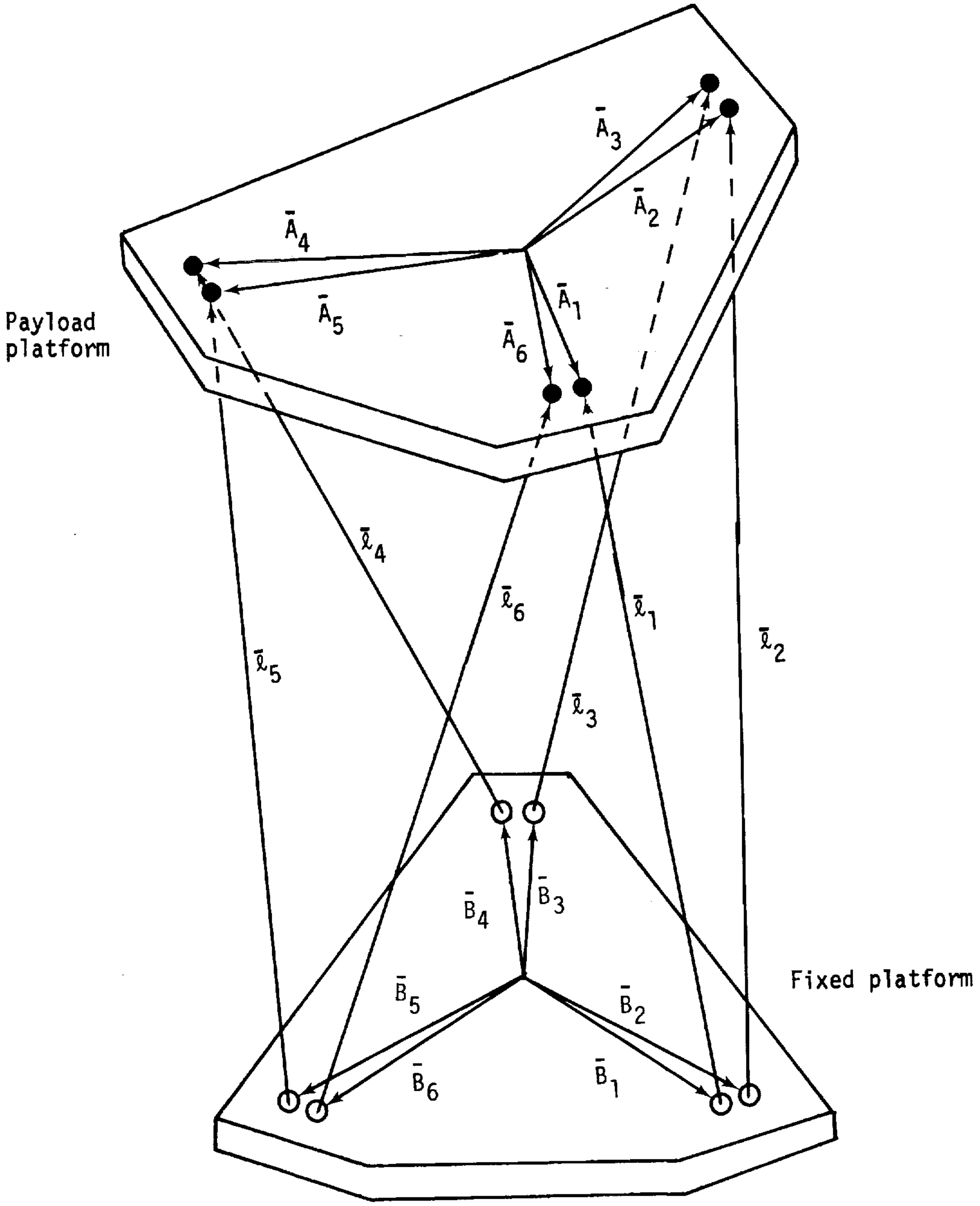


Figure 5.- Particular orientation of motion simulator.

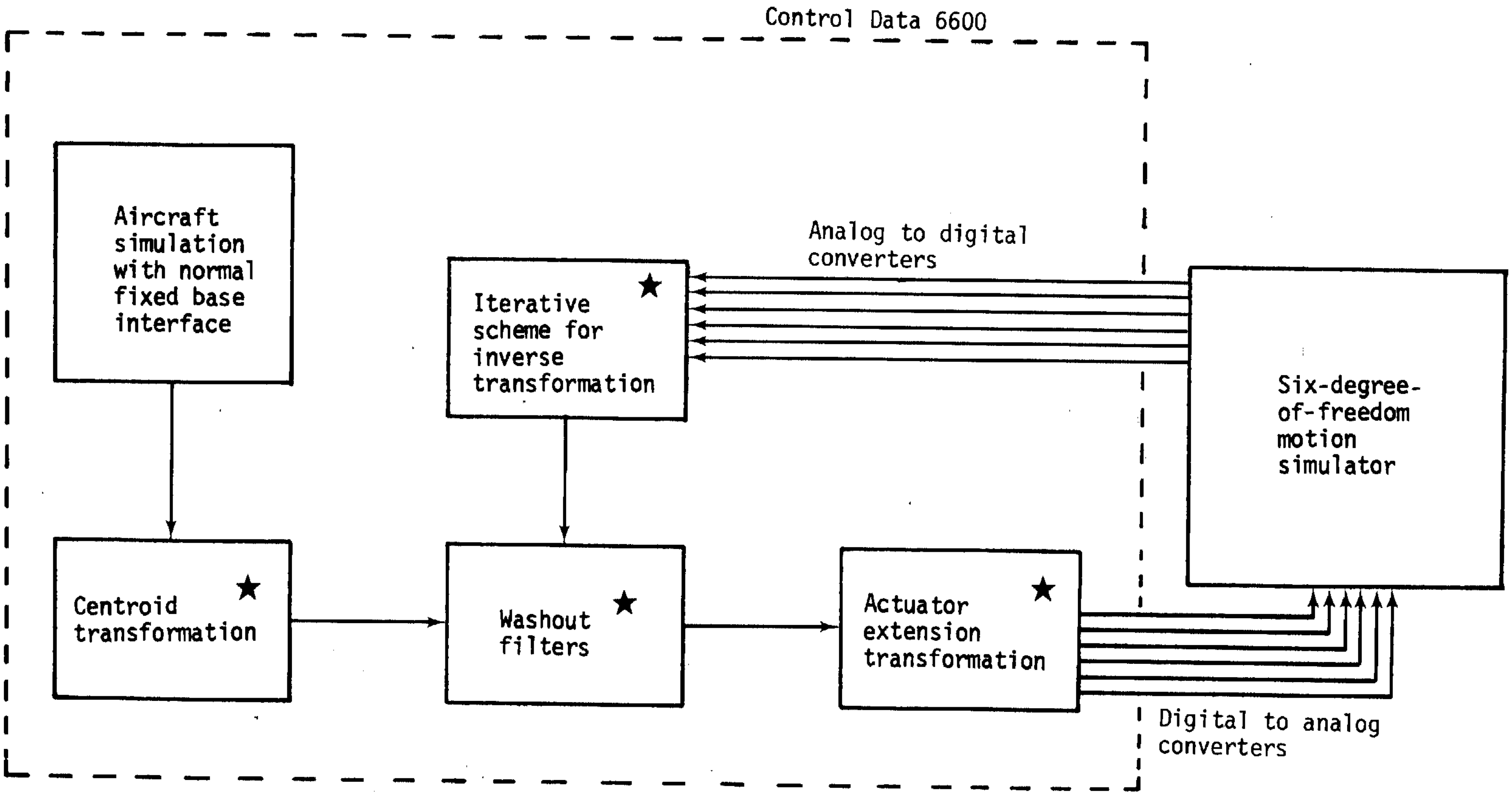

Control Data 6600
Aircraft simulation with normal fixed base interface
Centroid transformation ★
Iterative scheme for inverse transformation ★
Washout filters ★
Actuator extension transformation ★
Analog to digital converters
Digital to analog converters
Six-degree-of-freedom motion simulator

Figure 6.- Major blocks of a real-time motion simulation. Starred blocks indicate additional software required.

NASA-Langley, 1972 —— 8 L-8505