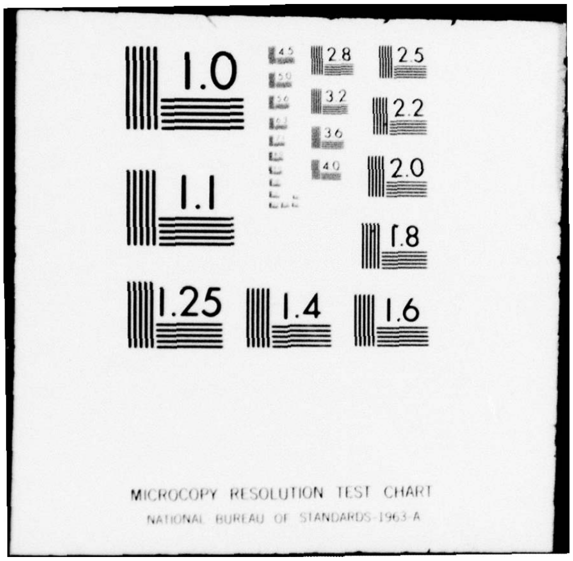
AD-A059 013 MOORE SCHOOL OF ELECTRICAL ENGINEERING PHILADELPHIA P--ETC F/G 9/2
ANALYSIS AND VALIDATION OF HUMAN BIODYNAMIC MODELS.(U)
AUG 78 N BADLER
N00014-78-C-0102
UNCLASSIFIED
NL

| OF |
AD
A059013

END
DATE
FILMED
11-78
DDC


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

LEGES SINE MORIBUS VANAE

UNIVERSITY of PENNSYLVANIA

The Moore School of Electrical Engineering

PHILADELPHIA, PENNSYLVANIA 19174

Office of Naval Research
⑮ Contract N00014-78-C-0102
Task No. NR 207-110
Final Report

⑥ Analysis and Validation of Human Biodynamic Models,

by

⑩ N. Badler

University of Pennsylvania
Department of Computer and Information Science
Philadelphia, PA 19104

⑪ 27 August 1978 ⑫ 33p.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

Distribution of this report is unlimited.

⑨ Final rept. 1 Dec 77-30 Jun 78,

410039

78 09 05 189

410 039

Analysis and Validation of Human Biodynamic Models

Table of Contents

I. Research Summary

Our research goal has been to investigate the integration of the Calspan (2), Bowman (3), and Linkman or Bubble-man (4) programs to provide a visual (graphic) facility for examining and evaluating these models. This research included transforming the Calspan body position data to the Linkman format, combining the body model with a given cockpit model, displaying both model and cockpit together in a readily visualizable manner, and detecting collisions between the body and the cockpit. Each of these areas is summarized below; further details are provided in the Appendices.

The Calspan-to-Bubble-man conversion program interfaces these two systems by converting data from one of the Calspan output files into a suitable input file for the Bubble-man program. By using files, no direct interaction or modification of either program is required. The significant portion of this effort is based on locating the appropriate data in the Calspan file and converting coordinate data from one model into the (different) model in Bubble-man. We used the body segment angular displacements to compute the joint angles required in Bubble-man. (Details are given in Appendix 1.) Figure 1 shows four positions of the body model which were produced by the Calspan simulator and displayed with the Bubble-man program.

Although the Calspan program is capable of writing out its variable information at as small a time step as desired (but of course not below the integration step size), it still may be desirable to interpolate between the Calspan results. Interpolation might be needed for producing an animated film sequence, for instance. For such uses, an APL program was developed. This program reads a Bubble-man angle file, such as that produced by the Calspan-Linkman interface program, and creates a new file of angles at a shorter time interval. How much shorter is determined by an input parameter. The inter-

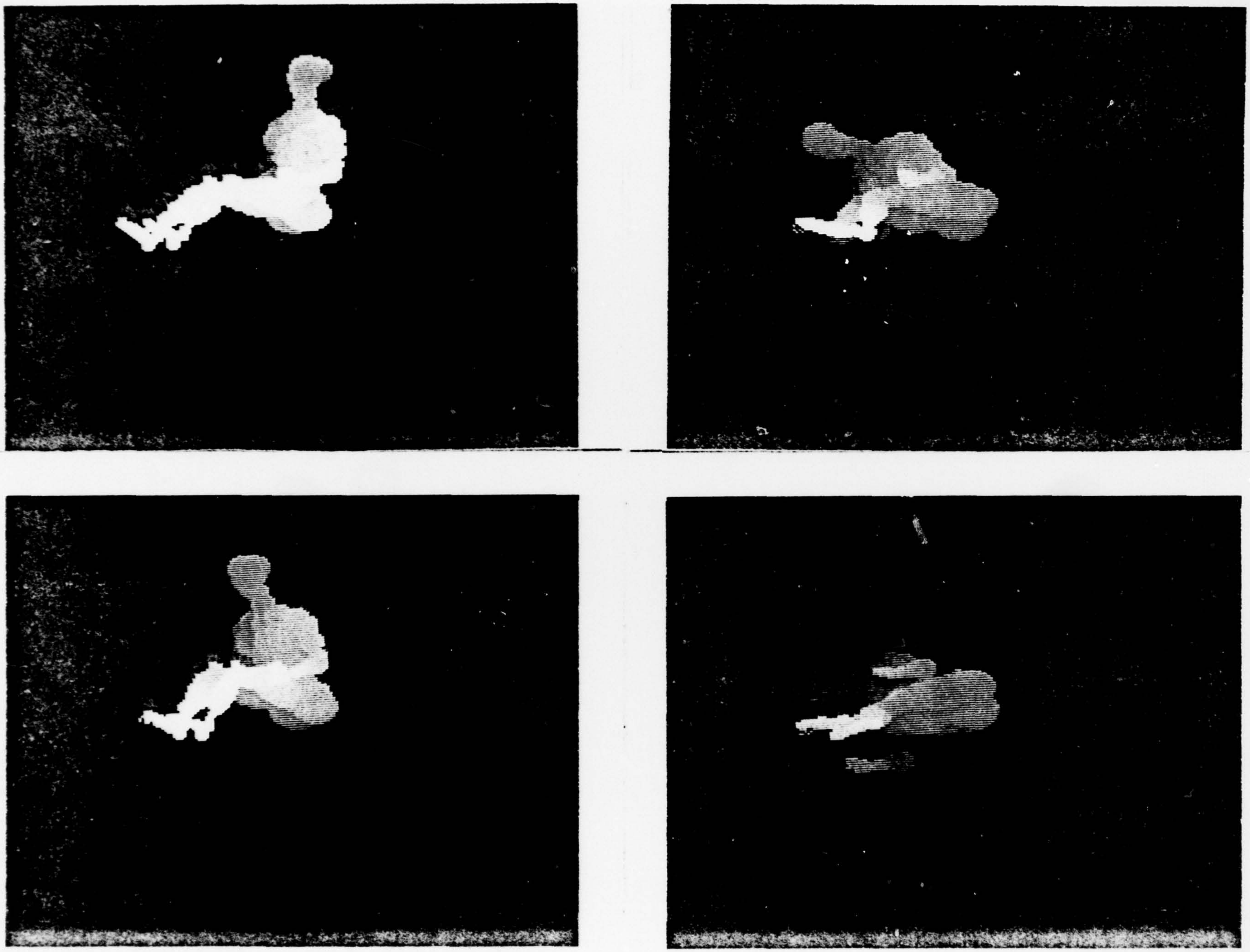


Figure 1. Crash sequence.

polation is a straight-forward linear interpolation. The APL listing is given in Appendix 2.

To produce understandable pictures of the human body model and cockpit data from the Bowman program, techniques were developed for combining Bubble-man with planar polygon cockpit surfaces (Figure 2). Since Bubble-man consists of overlapping spheres, no existing techniques could be used directly. Instead, the cockpit polygons and the Bubble-man spheres are entered into two arrays of points: one representing the actual intensity of that point on a raster display; and the other representing the depth (distance from the observer) of the closest scene component at that point. As each Bubble-man sphere or planar polygon is processed, it is checked against the current depth values at each point it affects. If its depth is greater, then nothing is changed. Otherwise the depth value is changed to the closer point and the intensity array is updated as well. The result is a shaded, solid rendering of body and cockpit (Figure 3).

Given the body position and cockpit model, collisions (intersections) between them may be discovered by simply computing the intersection between each sphere and polygon. In practice (since there are many spheres and polygons) the number of comparisons is reduced by checking each body segment against each polygon, then checking each sphere in the segment if an intersection were possible by the first test. Further details and an example of the collision detection process are given in Appendix 3.

Further information on the Bubble-man model and its properties may be found in Reference (1). We believe that the proposed integration of the modelling and simulation systems can be readily achieved as demonstrated by our results, and that such an integration is desirable in order to better utilize and visualize human biodynamic information.

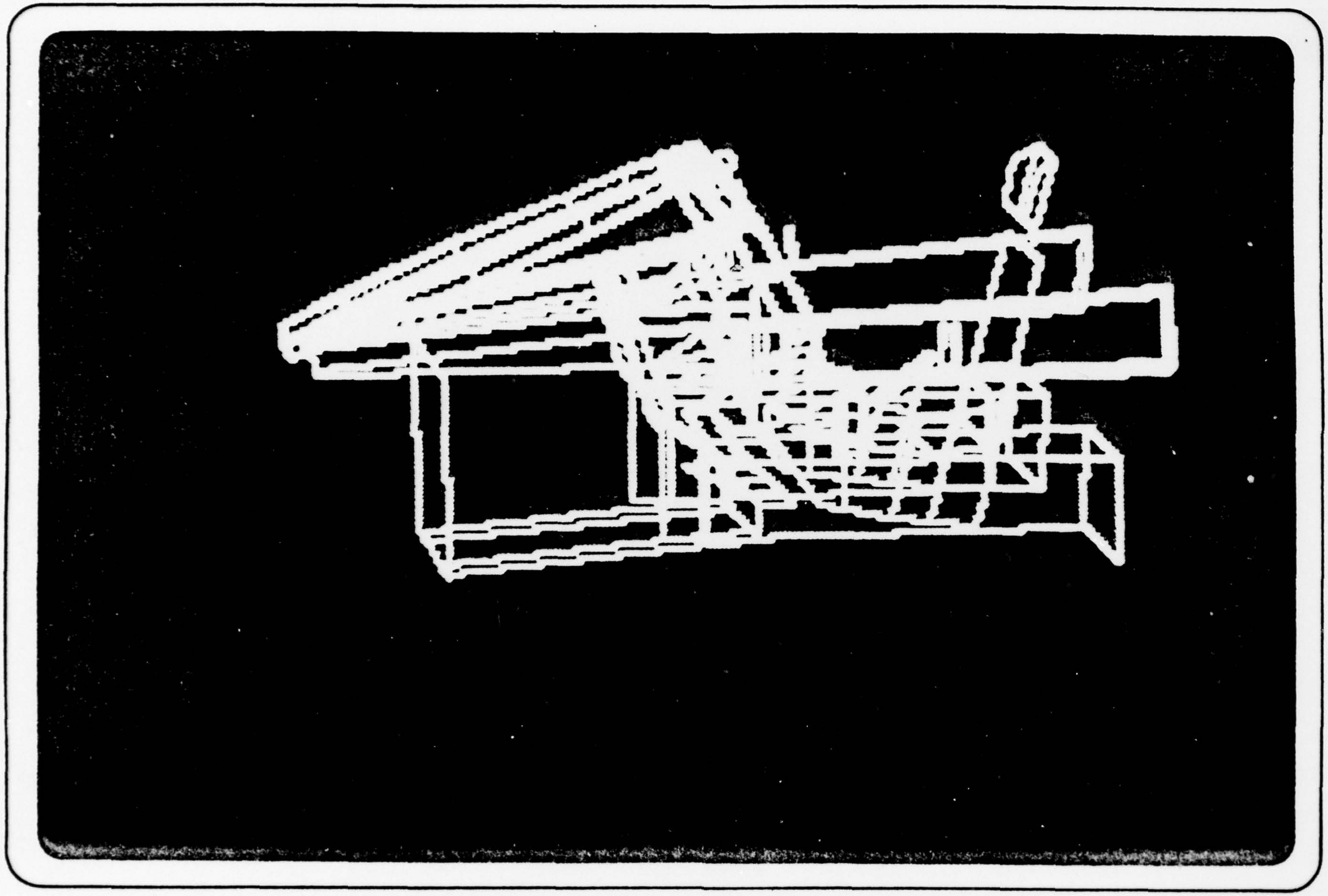


Figure 2a. Cockpit drawn with lines on raster display.

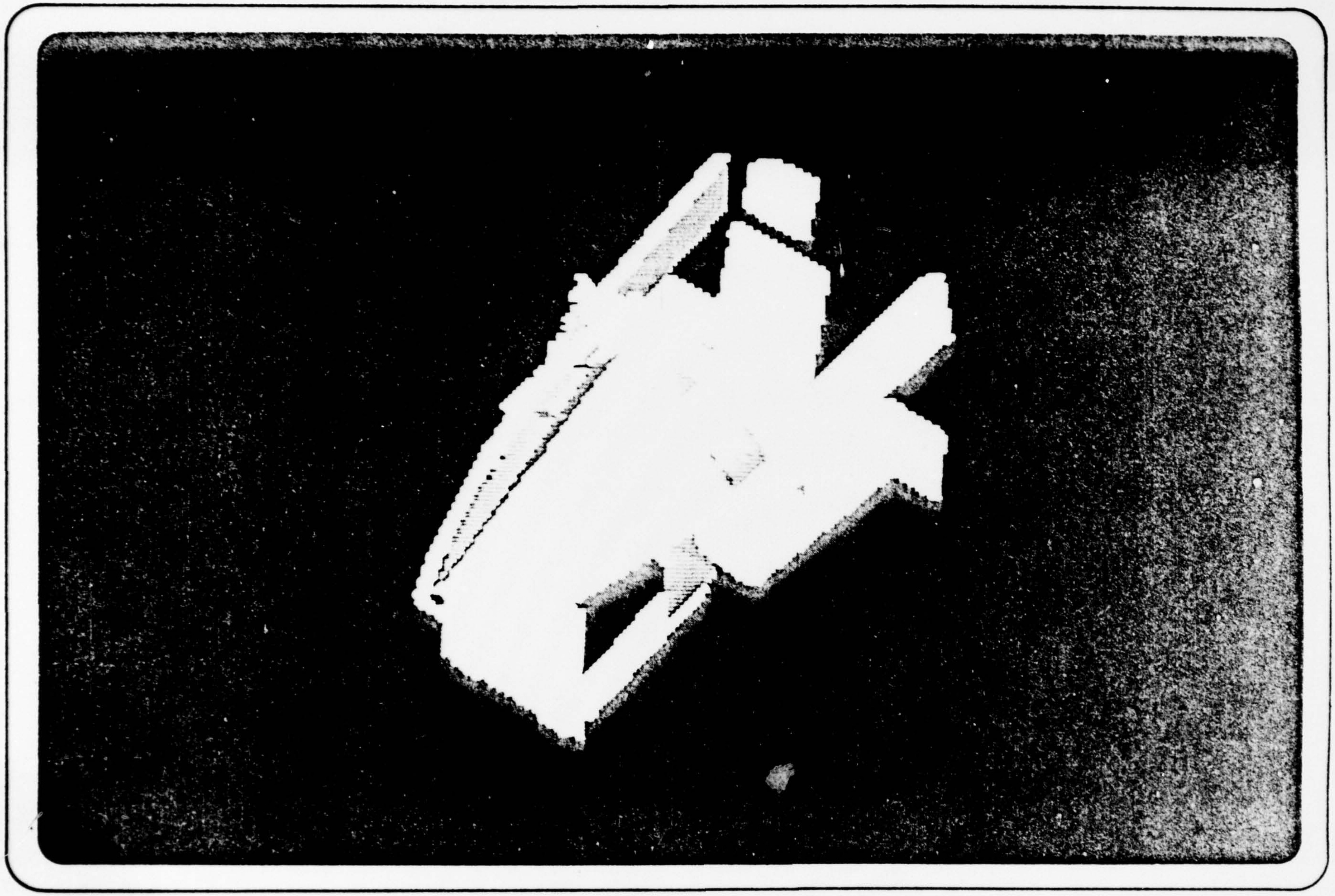


Figure 2b. Cockpit drawn with hidden surfaces removed.

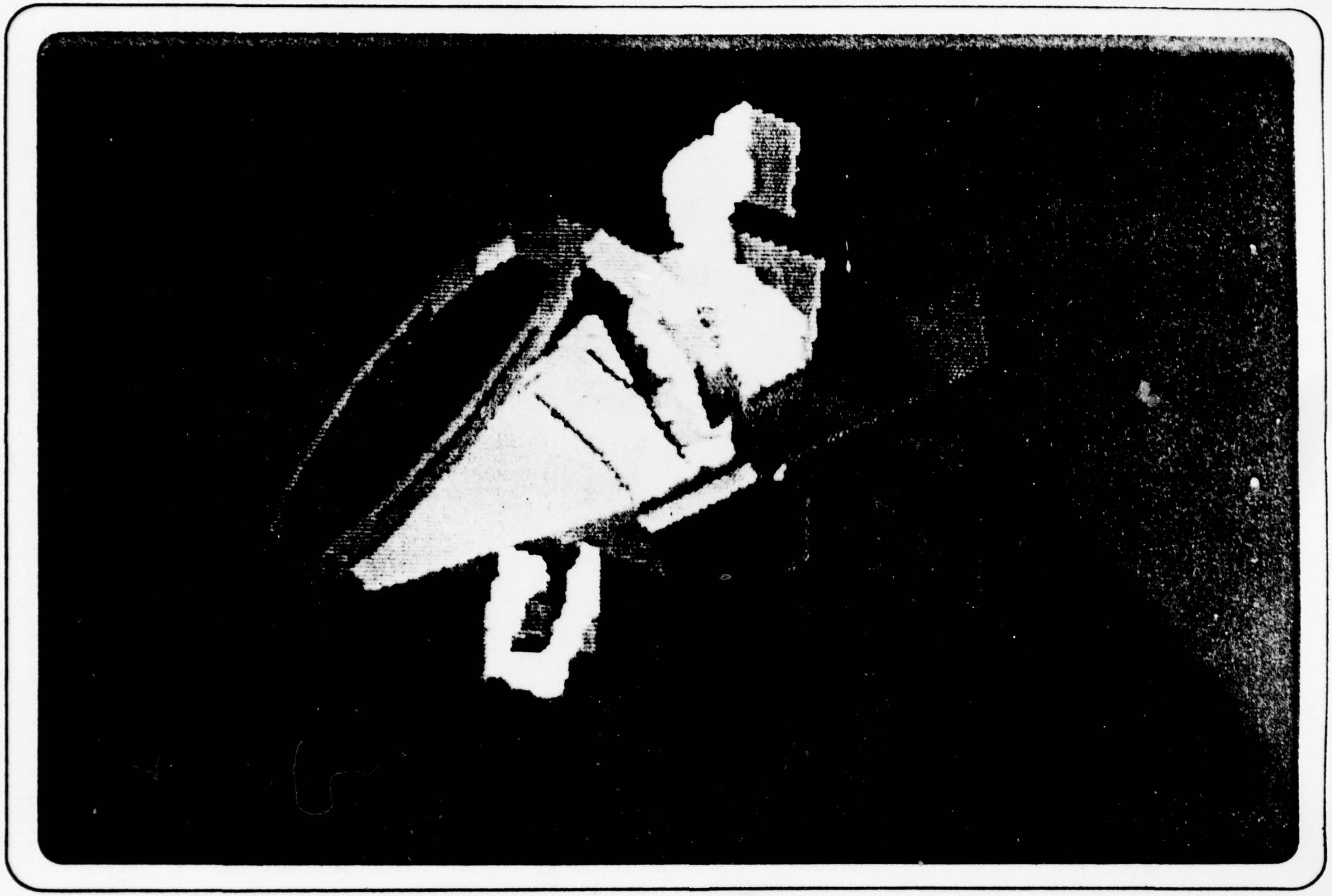


Figure 3. Human body model in cockpit.

References

(1) N. Badler, J. O'Rourke, and H. Toltzis, "A Human Body Model for Motion Studies," Department of Computer and Information Science Report, University of Pennsylvania, July 1978.

(2) J.T. Fleck, F.E. Butler, and S.L. Vogel, "An Improved Three Dimensional Computer Simulation of Vehicle Crash Victims," Volumes I,II,III, and IV, Calspan Corp., prepared for NHTSA, DOT, ZQ-5180-L-1, July 1974.

(3) Patrick W. Ryan, "Cockpit Geometry Evaluation - Phase III Final Report: Volume I. Program Description and Summary," Boeing under ONR contract N00014-71-C-0170, Boeing report number D162-10125-3, 1971.

(4) Joseph O'Rourke, "Representation and Display of Three Dimensional Objects with Spheres," MSc Thesis, University of Pennsylvania, Moore School of Electrical Engineering, Aug. 1977.

II. Technical Reports

Reference (1) above shall serve as a manuscript description of the overall thrust of this project. Other relevant details are provided as Appendices to this final report.

III. Publications

Reference (1) has been submitted for publication.* The cognizant Program Director for the Office of Naval Research will be notified and supplied with reprints upon acceptance.

IV. Conclusions

We have demonstrated that a solid model of the human body (Bubble-man) may be positioned by joint and segment data obtained from an existing movement simulator without modification of the latter. We have further demonstrated that the body, once positioned, may be placed in a planar polygon environment such as used for cockpit design and evaluation. Besides efficiently creating a graphics display of the composite (body plus cockpit), collisions between the body and itself or the cockpit may be easily computed, yielding concise (textual) descriptions of contact or intersection points. This information is essential to the evaluation of the cockpit design on the human occupant.

By integrating these three systems we open the possibility for video recording and preservation of real or simulation data: movements of the body may be examined as they occur in the cockpit or vehicle. Computer graphics permits the placement of the "observer" at any position, thus multiple views of the same movement sequence are easily produced. The full three-dimensionality and solidity of the model also provide a degree of realism unequalled in other

---

* IEEE Proceedings.

models. Such animations could be used to provide long-term archives of movement data, independent of computers or programs. By suitable video mixing, real and simulated data could be overlaid for validation of the simulator itself.

V. Major Accomplishments

We may itemize our conclusions as follows:

1. Bubble-man may be positioned by data obtained from existing movement simulators.
2. The Bubble-man model effectively portrays a body position on a raster graphics display: hidden parts are removed; the depth relationships are apparent; and joint deformations are eliminated.
3. Acceptable graphic images of Bubble-man may be produced on vector drawing as well as raster displays.
4. Joint positions obtained from a simulator may be interpolated to achieve additional (intermediate) positions for an animated display.
5. A movement simulator may be used to generate body positions, and that body may then be placed in a planar polygon environment (such as an airplane cockpit). Both may be displayed with hidden parts removed and visible parts shaded.
6. Collision detection between Bubble-man and a planar polygon environment is simple and efficient, yielding concise (textual) descriptions of contact or intersection points.
7. Demonstration of the feasibility of transferring simulation data to video tapes of the body model in a given environment, useful for evaluation and archival storage of experimental data.

VI. Acknowledgements

The research described in this report was due in large part to the efforts of Joseph O'Rourke in elucidating the Calspan data files and Hasida Toltzis in manipulating the Bowman data to create the cockpit displays, finding an efficient display algorithm, and programming the collision detector. The support of the Office of Naval Research under Contract N00014-78-C-0102 is gratefully acknowledged.

Appendix 1

Introduction

This appendix is concerned with the details of interfacing three different computer programs: The Calspan Crash Victim Simulator, The Bowman Cockpit Geometry Program, and Linkman or Bubble-man. Each program is useful in its own right, but used together they can become powerful tools for the analysis and validation of biodynamic models.

Calspan to Linkman Conversion

The main purpose of the Calspan to Linkman Conversion program is to act as an interface between two existing programs: The Calspan Crash Victim Simulation and The Linkman or Bubble-man program. The interface program (called CALBUB) accomplishes this by reading data from one of the Calspan output files and preparing an input file for the Linkman program. It works entirely from files: There is no direct interaction with either the Calspan or Linkman programs.

It may seem at first sight that the interfacing task is trivial, since the Linkman needs joint angles as input, and the Calspan program outputs joint angles (if requested). However, the joint angles output by the Calspan program are intended to explicate the joint torques, not describe the position of the body. Therefore, they are given as flexural and twist angles from the "principle axes" defined for the joints. It is not possible to convert these flexural and twist angles into the yaw, pitch, and roll angles needed by the Linkman program. This is because the flexural and twist angles only represent two degrees of freedom, the yaw, pitch, and roll system expresses the full three degrees of freedom.

In contrast to these difficulties, the segment angular displacements represent a reliable source of data for the joint angles. Normally the segment angular displacements for all segments are written out by the Calspan program, and the conversion into joint angles is conceptually clean, but by no means trivial. The program CALBUB uses these segment angular displacements.

The program will be described in three parts: the reading of the Calspan output file, the angle conversion, and the writing of the Linkman input file.

Reading of Calspan File

The Calspan Simulator creates a number of output files. One of them, TAPE4, is designed for time-curve plotting, and contains the angular displacements of the segments (if requested). In order to read this file and extract only the segment angular displacements, it is necessary to understand the precise file structure and positions of the data.

TAPE4 consists of a number of records, each record of which consists of all the variables at a particular time. Thus the overall structure of the file is as in Figure 1.

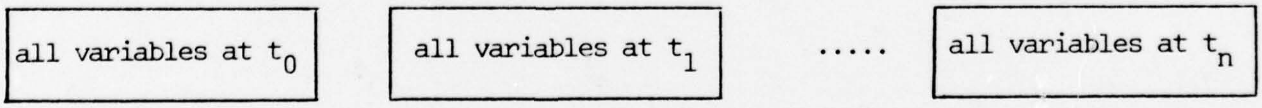


Figure 1

Within each time record, the variables are divided into ten groups, named as follows:

LA	linear acceleration
LV	linear velocity
LD	linear displacement
AA	angular acceleration
AV	angular velocity
AD	angular displacement
JP	joint parameters
PS	plane-segment contact faces
BS	belt-segment contact faces
SS	segment-segment

We will call each of these groups a *code*, and will refer to them by the above two letter designations. For each code, there may be a number of different *data sets*. These data sets can usually be identified with segments, but not always. For example, for the code LA, there may be three data sets: neck c.g., head c.g., and head at neck pivot. In this example, there are two data sets for the same head segment. The last level of structure is represented by *components*. Each data set has a number of components, for example, x,y,z, and resultant. The number of components is determined by the code, not the data set. Thus all LA data sets always have 4 components each, all JP data sets have 6 components, etc.

To summarize, the structure of the Calspan output file can be characterized by 4 different levels:

1. records
2. codes
3. data sets
4. components

In order for the interface program CALBUB to read the Calspan file, it must be told the particular file structure to expect. For this purpose, the CALBUB program reads file definition information from TAPE5. This file (TAPE5) is the only input to CALBUB other than TAPE4, the Calspan output file.

We have now set enough background to discuss the precise form of TAPE5, the input definition file.

CARD1. either DEBUG or NO DEBUG (format A10)

This controls the printing of debugging information in TAPE2.

CARD2. The number of data sets per code (free format)

CARD3.1
⋮
CARD 3. NANGDS — The names of the data sets (i.e. segments) for angular displacement data (format A10)

CARD4. number of joints

CARD5.1
⋮
CARD5.JMAX — The names of each joint and its two associated segments (format 3A10)

Card2 specifies the file structure by listing the number of data sets per code. Cards 3 give the names of the data sets for the angular displacement data which we are interested in, and Cards 5 associate these segment or data set names with the joint names. A sample 14 segment input deck is shown below.

```
NO DEBUG
11,1,12,2,2,15,4,0,0,0
LT
CT
UT
N
H
RUL
RLL
RF
LUL
LLL
LF
RUA
RLA
LUA
LLA
14
P         LT        CT        *
W         CT        UT        *
NP        UT        N         *
HP        N         H         *
RH        LT        RUL       *
RK        RUL       RLL       *
RA        RLL       RF        *
LH        LT        LUL       *
LK        LUL       LLL       *
LA        LLL       LF        *
RS        UT        RUA       *
RE        RUA       RLL       *
LS        UT        LUA       *
LE        LUA       LLL       *
```

After the input file TAPE5 is read in by subroutine RDEF, a routine PREPRT is called to prepare for the reading of the Calspan tape. This routine uses the definition information to compute a mapping vector POINTER which specifies exactly which positions within each time record of the Calspan file should be stored for future use. Then subroutine RTAPE is called, and this reads in each record and stores away the appropriate information according to the mapping vector. When RTAPE is finished, all the Calspan angles are stored within the array CALANG as follows:

$t_o\ y_1\ p_1\ r_1\ \ y_2\ p_2\ r_2\ \cdots\ y_n\ p_n\ r_n\qquad t_1\ y_1\ p_1\ r_1\ \cdots$

The angles are stored in the order they appear on the original Calspan tape, i.e., the order represented by cards 3 of the input definition file.

This represents the completion of the first phase of the CALBUB interface program: the Calspan tape has been read and the segment angular displacement data stored in the array CALANG.

Angle Conversion

Before we can discuss the conversion between the Calspan and Linkman angles, it is necessary to make clear the definition of the angles. Both the Calspan and Linkman coordinate systems are all right handed, with the three angles defined as:

yaw: + x to y

pitch: + z to x

roll: + y to z

However, all the Calspan coordinate systems have their z-axis pointing downwards, (with gravity) whereas all the Linkman coordinate systems have their z-axis pointing upwards (against gravity). The Calspan convention is responsible for an illusion that the coordinate systems in the upper and lower body parts are different. They are not. It is just that the segments sometimes "hang" under the coordinate system, and sometimes sit on top of the coordinate system.

In order to change the angles relative to the Calspan inverted coordinate system into angles relative to an upright coordinate system, the following transformation is performed:

1. change sign of yaw
2. change sign of pitch
3. leave roll unchanged

Once this transformation is performed, we no longer need worry about differences between coordinate system definitions.

All the Calspan segment angular displacements are expressed as rotations from the vehicle to the segment. The angles needed for the Linkman program are the joint angles: rotation from one segment to the adjacent segment connected by the joint. The first segment is more proximal to the body center, the second segment more distal. The two segments which together define a joint angle are specified by the card group 5 in the input definition file.

The method of computing the Linkman joint angles given the two associated Calspan segment angles (expressed in upright coordinate systems) is as follows. From the segment 1 angles, a rotation matrix is computed which transforms from segment 1 to the vehicle. Likewise a matrix is found which transforms from the vehicle to segment 2. The composition of these two rotations gives a transformation from segment 1 to segment 2 (See Figure 2).

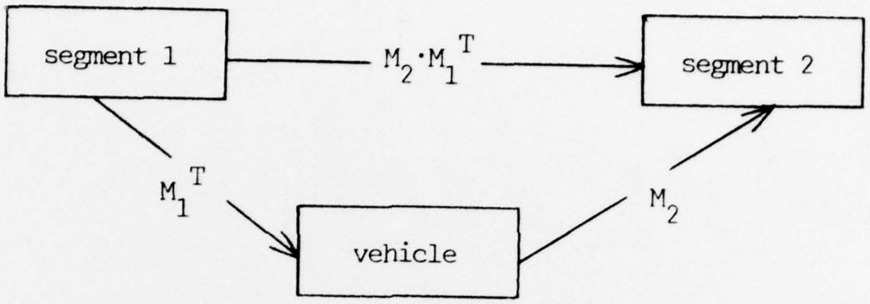


Figure 2

The rotation matrices are direction cosine matrices, and are computed as:

cos p cos y	cos p sin y	-sin p
-sin y cos r +sin r sin p cos y	cos y cos r +sin r sin p sin y	cos p sin r
+sin y sin r +cos r sin p cos y	-sin r cos y +cos r sin p sin y	cos p cos r

The combined transformation $M_2 \cdot M_1^T$ has this form. Once we have completed this matrix, our next task is to find three angles which could give rise to the matrix. These angles are the Linkman joint angles we are seeking. By inspection of the matrix elements we can derive the following formulas:

yaw = arctan (M(1,2) / M(1,1))

pitch = arcsin (M(1,3))

roll = arctan (M(2,3) / M(3,3))

By the nature of the inverse trigonometric functions, these formulas leave some ambiguity. In particular, we could also have:

yaw + 180

supplement (pitch)

roll + 180

This gives us eight possible solutions.

Although there may be a method of choosing among these 8 possibilities analytically, we have found it easier to simply substitute each angle set back into the definition of the direction cosine matrix and see if indeed the angles yield the original matrix. The result of this substitution is that **two** angle sets are always found which will give the original matrix. These two angle sets both represent the exact same angular orientation. The set which is closer to that used as the solution for the previous time step is chosen as the solution for the current time step.

At the completion of the angle conversion phase of the program, all the joint angles have been computed by the above process and stored into an array BUBANG, whose structure is exactly similar to the structure of CALANG.

<u>Writing of Angle File</u>

The last phase of the CALBUB program is to write the joint angles on a file in a form ready to be read in by the Linkman program. This is rather straightforward, but there are two further changes made to the data.

First, the Calspan and Linkman programs differ in their definition of ankle angles. In the Linkman program, all zeros for the ankle angles gives the normal relaxed standing position; in the Calspan program, all zeros means the feet are pointing straight downwards. Thus 90 degrees are added to the pitch of the Calspan ankle angles to be consistent with the Linkman conventions.

Second, it is normally the case that the Calspan output does not give sufficient information for determining every Linkman joint angle. The simulation may not have used as many segments as there are in the Linkman model, or not all of the information may have been directed to the Calspan output file. A list of all the joint names used in Linkman is therefore compared against the joints defined by the Calspan data (specified by card group5 on the input definition file), and for any joints which are not defined, default values are substituted. These default values are set in a data statement within subroutine WFILE.

With these two changes made, the joint angles are written out on a file (TAPE6) which can serve as the input file (TAPE4) for the Linkman program.

Using the Interface Program

There are three steps in any use of the CALBUB program:

1. Run the Calspan Simulator and save TAPE4.
2. Run CALBUB using job control cards equivalent to the following:

```
GET (TAPE4 = [file name of Calspan output  file])

GET (TAPE5 = [file name of input definition file])

GET (LGO + CALBUBB)
FILE (TAPE4, RT=S...)
LDSET(PRESET= ZERO, MAP=S, FILES=TAPE4)

LGO.

REPLACE (TAPE6 = [file name for Linkman angles])
```

3. Run the Linkman program using job control cards equivalent to:

```
GET (TAPE3 = [file name of man data])

GET (TAPE4 = [file name of angles])

GET (PL10LIB/ UN=SYSTEM)

GET (LNKMANB)

LDSET (PRESET=ZERO, MAP=S, LIB=PL10LIB)

LNKMANB.
```

Appendix 2

APL angle interpolation program.

```
∇ALL[⎕]∇
         ∇RES←I1 ALL I2
[1]       I←I1
[2]       RES←0 FOR M[I1;;]
[3]       I←I-1
[4]      LOOP:→((I2-1)<(I←I+1))/0
[5]       M[I;;]INTERP M[I+1;;]
[6]       I←I+1
[7]       RES←RES⍪0 FOR M[I;;]
[8]       I←I-1
[9]       →LOOP
         ∇
         ∇ASUB[⎕]∇
         ∇Z←A1 ASUB A2
[1]       Z←A1-A2
[2]       Z←NORM Z
         ∇
         ∇EXEC[⎕]∇
         ∇Z←EXEC CM
[1]       Z←⍳0
[2]       I←0
[3]      LOOP:→(95<I←I+1)/0
[4]       V←CM[I;]
[5]       V[(V='-')/⍳30]←'¯'
[6]       V←(~(V='.')∧(1⌽V=' '))/V
[7]       Z←Z,⍎V
[8]       →LOOP
         ∇

         ∇M1 INTERP M2;DELTA;M;J
[1]       ⍝
[2]       ⍝ THIS IS THE CONTROLLING FUNCTION FOR
[3]       ⍝ THE INTERPOLATION
[4]       ⍝
[5]       DELTA←M2 ASUB M1
[6]       DELTA←DELTA÷N
[7]       M←M1
[8]       J←0
[9]      LOOP:→((N-1)<(J←J+1))/0
[10]      M←M+DELTA
[11]      RES←RES⍪J FOR M
[12]      →LOOP
         ∇
```

```
          ∇RES←I1 ALL I2
    [1]     ⍝
    [2]     ⍝ THIS IS THE MAIN FUNCTION FOR THE
    [3]     ⍝ INTERPOLATION PROGRAM
    [4]     ⍝
    [5]     I←I1
    [6]     RES←0 FOR M[I1;;]
    [7]     I←I-1
    [8]    LOOP:→((I2-1)<(I←I+1))/0
    [9]     M[I;;]INTERP M[I+1;;]
    [10]    I←I+1
    [11]    RES←RES⍪0 FOR M[I;;]
    [12]    I←I-1
    [13]    →LOOP
          ∇

          ∇Z←A1 ASUB A2
    [1]     ⍝
    [2]     ⍝ THIS FUNCTION SUBTRACTS TWO ANGLES
    [3]     ⍝
    [4]     Z←A1-A2
    [5]     Z←NORM Z
          ∇

          ∇Z←EXEC CM
    [1]     ⍝
    [2]     ⍝ THIS FUNCTION EXECUTES A CHARACTER MATRIX
    [3]     ⍝
    [4]     Z←⍳0
    [5]     I←0
    [6]    LOOP:→(95<I←I+1)/0
    [7]     V←CM[I;]
    [8]     V[(V='-')/⍳30]←'¯'
    [9]     V←(~(V='.')∧(1⌽V=' '))/V
    [10]    Z←Z,⍎V
    [11]    →LOOP
          ∇

          ∇Z←J FOR M
    [1]     ⍝
    [2]     ⍝ THIS FUNCTION WILL FORMAT THE DATA IN
    [3]     ⍝ PREPARATION FOR OUTPUT
    [4]     ⍝
    [5]     Z←1 31⍴31↑TITLES[I;],'.',⍕J
    [6]     Z←Z⍪(10 2⍕M),'*'
          ∇
```

```
        ∇Z←NOR A
[1]      ⍝
[2]      ⍝ THIS FUNCTION NORMALIZES AN ANGLE
[3]      ⍝
[4]      Z←A
[5]     LOOP:→(Z>180)/SUB
[6]      →(Z<¯180)/ADD
[7]      →0
[8]     SUB:Z←Z-360
[9]      →LOOP
[10]    ADD:Z←Z+360
[11]     →LOOP
        ∇

        ∇Z←NORM M
[1]      ⍝
[2]      ⍝ THIS FUNCTION NORMALIZES A MATRIX
[3]      ⍝
[4]      Z←19 3⍴NORV,M
        ∇

        ∇Z←NORV V;A
[1]      ⍝
[2]      ⍝ THIS FUNCTION NORMALIZES A VECTOR
[3]      ⍝
[4]      →(0≠⍴V)/RECURSE
[5]      Z←V
[6]      →0
[7]     RECURSE:
[8]      A←,1↑V
[9]      V←1↓V
[10]     Z←(NOR A),NORV V
        ∇

        ∇Z←REA FN
[1]      ⍝
[2]      ⍝ THIS FUNCTION READS THE RAW
[3]      ⍝ UNINTERPOLATED DATA FROM A FILE
[4]      ⍝
[5]      0 CFPOS FN
[6]      Z←0 31⍴''
[7]     LOOP:
[8]      R←CFREAD FN
[9]      →(0=1↑0↑R)/END
[10]     Z←Z⍪31↑R
[11]     →LOOP
[12]    END:
[13]     'RETURNED ';R
        ∇
```

Appendix 3

Sphere - polygon collision algorithm.

1. Find plane containing polygon.
2. Find distance between sphere surface and polygon plane. If this distance is greater than zero, sphere and plane do not intersect; otherwise sphere intersects plane in a (possibly degenerate) circle.
3. Coordinatize plane of polygon, then check if circle intersects or contains enclosing box of polygon. If not, sphere and polygon do not intersect.

4.a If center of circle is inside polygon, then sphere and polygon intersect.

b. If center is outside polygon and any edge of polygon intersects circle, then sphere intersects polygon.

In case the circle in 4a is a single point the intersection may be called a "touch" (tangency). Otherwise the volumes overlap. Also, in case 4b, one may wish to distinguish the case in which the circle is tangent to the polygon, but not otherwise intersecting it.

An example of this algorithm on the body and cockpit shown in Figure 2 of the report yields the following collision description. (The polygon names are supplied from the cockpit data file, the body segment names from the Bubble-man program.)

```
COLLISION   LOWER TORSO                            RH CONSOLE
COLLISION   LOWER TORSO                            RIGHT HAND CONSOLE WALL SIDE
COLLISION   LOWER TORSO                            RIGHT HAND CONSOLE WALL REAR
COLLISION   LOWER TORSO                            SEAT BACK
COLLISION   LOWER TORSO                            SEAT PAN AFT
COLLISION   CENTRAL TORSO                          SEAT BACK
COLLISION   UPPER TORSO                            SEAT BACK
COLLISION   UPPER TORSO                            SEAT BACK
COLLISION   RIGHT SHOULDER MASS                    SEAT BACK
COLLISION   RIGHT UPPER ARM                        SEAT BACK
COLLISION   LEFT UPPER LEG                         RH CONSOLE
COLLISION   LEFT UPPER LEG                         RIGHT HAND CONSOLE WALL SIDE
COLLISION   LEFT UPPER LEG                         SEAT RIGHT SIDE AFT
COLLISION   LEFT UPPER LEG                         SEAT RIGHT SIDE CENTER AFT
COLLISION   LEFT UPPER LEG                         SEAT RIGHT SIDE CENTER FWD
COLLISION   LEFT UPPER LEG                         SEAT BACK
COLLISION   LEFT UPPER LEG                         SEAT BACK
COLLISION   LEFT UPPER LEG                         SEAT PAN MID
COLLISION   LEFT UPPER LEG                         SEAT PAN AFT
COLLISION   RIGHT UPPER LEG                        RH CONSOLE
COLLISION   RIGHT UPPER LEG                        RIGHT HAND CONSOLE WALL SIDE
COLLISION   RIGHT UPPER LEG                        LH CONSOLE
COLLISION   RIGHT UPPER LEG                        LEFT HAND CONSOLE WALL  SIDE
COLLISION   RIGHT UPPER LEG                        SEAT LEFT SIDE AFT
COLLISION   RIGHT UPPER LEG                        SEAT LEFT SIDE CENTER AFT
COLLISION   RIGHT UPPER LEG                        SEAT LEFT SIDE CENTER FWD
COLLISION   RIGHT UPPER LEG                        SEAT RIGHT SIDE AFT
COLLISION   RIGHT UPPER LEG                        SEAT RIGHT SIDE CENTER AFT
COLLISION   RIGHT UPPER LEG                        SEAT BACK
COLLISION   RIGHT UPPER LEG                        SEAT BACK
COLLISION   RIGHT UPPER LEG                        SEAT PAN MID
COLLISION   RIGHT UPPER LEG                        SEAT PAN AFT
COLLISION   LEFT LOWER LEG                         SEAT PAN FWD RIGHT
COLLISION   LEFT FOOT                              CONTROL STICK GRIP RIGHT LOWER
COLLISION   LEFT FOOT                              UPPER YOKE RH SIDE
COLLISION   LEFT FOOT                              UPPER YOKE RH SIDE
COLLISION   LEFT FOOT                              CONTROL STICK GRIP LEFT LOWER
COLLISION   LEFT FOOT                              UPPER YOKE LH SIDE
COLLISION   LEFT FOOT                              UPPER YOKE AFT
COLLISION   LEFT FOOT                              UPPER YOKE AFT
COLLISION   LEFT FOOT                              UPPER YOKE FWD
COLLISION   LEFT FOOT                              CONTROL STICK GRIP LOWER FWD
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP RIGHT LOWER
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP RIGHT SIDE
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP TOP RIGHT
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP KNOB RIGHT
COLLISION   RIGHT FOOT                             UPPER YOKE RH SIDE
COLLISION   RIGHT FOOT                             UPPER YOKE AFT
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP LOWER FWD
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP FORWARD
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP TOP FORWARD
COLLISION   RIGHT FOOT                             CONTROL STICK GRIP KNOB FORWAR
```

Appendix 4

Visits and Deliveries to the Naval Air Development Center, Warminster, PA

Date	Purpose of visit
3 Jan 78	discussion
25 Jan 78	discussion
28 Apr 78	discussion; delivery of 24 color slides
25 May 78	delivery of 54 slides, 3 computer listings, and 2 card decks
7 Jun 78	programming
9 Jun 78	programming
14 Jun 78	programming
6 Jul 78	delivery of final listings

OFFICE OF NAVAL RESEARCH
BIOLOGICAL SCIENCES DIVISION
BIOPHYSICS PROGRAM, Code 444
DISTRIBUTION LIST FOR TECHNICAL, ANNUAL AND FINAL REPORTS

Number of Copies	
(12)	Administrator, Defense Documentation Center Cameron Station Alexandria, Virginia 22314
(6)	Director, Naval Research Laboratory Attention: Technical Information Division Code 2627 Washington, D. C. 20375
(6)	Office of Naval Research Attention: Code 102IP (ONRL DOC) 800 N. Quincy Street Arlington, Virginia 22217
(3)	Office of Naval Research Biophysics Program Code 444 Arlington, Virginia 22217
(1)	Commanding Officer Naval Medical Research and Development Command National Naval Medical Center Bethesda, Maryland 20014
(1)	Chief, Bureau of Medicine and Surgery Department of the Navy Washington, D. C. 20375
(2)	Technical Reference Library Naval Medical Research Institute National Naval Medical Center Bethesda, Maryland 20014
(1)	Office of Naval Research Branch Office 495 Summer Street Boston, Massachusetts 02210
(1)	Office of Naval Research Branch Office 536 South Clark Street Chicago, Illinois 60605

Enclosure (3)

(1) Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91106

(1) Commanding Officer
Naval Medical Research Unit No. 2
Box 14
APO San Francisco 96263

(1) Commanding Officer
Naval Medical Research Unit No. 3
FPO New York 09527

(1) Officer in Charge
Submarine Medical Research Laboratory
Naval Submarine Base, New London
Groton, Connecticut 06342

(1) Scientific Library
Naval Medical Field Research Laboratory
Camp Lejeune, North Carolina 28542

(1) Scientific Library
Naval Aerospace Medical Research Institute
Naval Aerospace Medical Center
Pensacola, Florida 32512

(1) Commanding Officer
Naval Air Development Center
Attn: Aerospace Medical Research Department
Warminster, Pennsylvania 18974

(1) DIRECTOR
Naval Biosciences Laboratory
Building 844
Naval Supply Center
Oakland, California 94625

(1) Commander, Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

(1) DIRECTORATE OF LIFE SCIENCES
Air Force Office of Scientific Research
Bolling Air Force Base
Washington, D. C. 20332

(1) Commanding General
Army Medical Research and Development Command
Forrestal Building
Washington, D. C. 20314

(1) Department of the Army
U. S. Army Science and
Technology Center - Far East
APO San Francisco 96328

(1) Assistant Chief for Technology
Office of Naval Research, Code 200
800 N. Quincy Street
Arlington, Virginia 22217

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		**READ INSTRUCTIONS BEFORE COMPLETING FORM**
1. REPORT NUMBER (final)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) Analysis and Validation of Human Biodynamic Models		5. TYPE OF REPORT & PERIOD COVERED Final 12/1/77 to 6/30/78
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(*s*) N. Badler		8. CONTRACT OR GRANT NUMBER(*s*) N00014-78-C-0102
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Pennsylvania, Philadelphia, PA 19104 Dept. of Computer & Information Science		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task No. NR207-110
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Dept. of the Navy Arlington, VA 22217		12. REPORT DATE 27 August 1978
		13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*) Office of Naval Research Dept. of the Navy Arlington, VA 22217		15. SECURITY CLASS. (*of this report*) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

Distribution of this abstract is unlimited.

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

computer graphics, human body models, cockpit design, vehicle occupants, crash simulation.

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

We have integrated a cockpit design program, a crash victim simulator, and a new model of a human body to provide a visual facility for examining and evaluating these models. This research included transforming the motion simulator output to position the realistic body model, combining the body model and the cockpit data in a shaded graphics display, and detecting collisions between the body and the cockpit. Each of these processes is described and illustrated.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)