

U N I V E R S I T Y O F M I C H I G A N

Technical Report 26

Computer Graphics in the  
Dynamic Analysis of Mechanical Networks

Milton A. Chace  
Michael E. Korybalski

CONCOMP: Research in Conversational Use of Computers  
F. H. Westervelt, Project Director  
ORA Project 07449

supported by:  
ADVANCED RESEARCH PROJECTS AGENCY  
DEPARTMENT OF DEFENSE  
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050  
ARPA ORDER NO. 716

administered through:  
OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

February 1970

engn  
UMR1042

## TABLE OF CONTENTS

ABSTRACT. . . . .	ix
1. Introduction. . . . .	1
2. Background of Program . . . . .	8
2.1 Constitutive Elements . . . . .	8
2.2 Definition of Constraint . . . . .	10
2.3 Dynamic Conditions - Lagrange's Equation . . . . .	12
2.4 Relation of Topological and Dimensional Problems . . . . .	23
3. Graphics Systems and Applications . . . . .	25
3.1 Systems Viewpoint. . . . .	25
3.2 The Graphic Terminal . . . . .	28
3.3 Description of DF Routines . . . . .	29
3.4 DAMN User Graphics Options . . . . .	29
3.5 Organization of Graphics Output Program. . . . .	31
3.6 Description and Examples of Output . . . . .	37
3.7 Man-Machine Interaction. . . . .	45
ACKNOWLEDGMENTS . . . . .	49
REFERENCES. . . . .	50



LIST OF TABLES

Table 1.	Form of the Generalized Force Terms in the Lagrange Equation. . . . .	19
Table 2.	Functions of the Supervisory Programs of the University of Michigan IBM 360/67 . . .	27
Table 3.	Definitions of DF Routines Used in the DAMN Graphics Output Program. . . . .	30
Table 4.	Graphic Options and Description of Output Available to DAMN User. . . . .	32
Table 5.	Description of Data Necessary for Genera- tion of DAMN Graphic Display. . . . .	33



## LIST OF FIGURES

Figure 1.	Flow Diagram of Operation of the DAML Program in Graphic Output Mode. . . . .	9
Figure 2.1.	Schematic of a Hypothetical Machine in Terms of Constitutive Elements . . . . .	11
Figure 2.2.	Schematic of a Kinematic Slider-crank Mechanism . . . . .	13
Figure 2.3.	A Chain of Keyed Assemblies from Ground to the Center of Mass of a Particular Part. . . . .	17
Figure 3.1.	Relation of U of M System Software in the Execution of Batch and Timeshared Jobs on the 360/67 Central Computer . . .	26
Figure 3.2.	Flow Chart of the Graphics Output Program . . . . .	34
Figure 3.3.	Example Graphics Output . . . . .	35
Figure 3.4a.	Three-dimensional View of a Two-link Compound Pendulum Moving in a Gravitational Field. . . . .	38
Figure 3.4b.	Two-dimensional Schematic of the Compound Pendulum Shown in Figure 3.4a . . .	39
Figure 3.5.	A Series of Photographs of the DEC 338 Graphic Display Screen Taken During a Program Run Simulating the Motion of the Compound Pendulum Shown in Figures 3.4a and 3.4b . . . . .	40
Figure 3.6a.	Automobile Hood Mechanism Used in Connecting the Hood to the Body . . . . .	42
Figure 3.6b.	Two-dimensional Schematic Representation of the Hood Mechanism of Figure 3.6a. . . . .	43
Figure 3.7.	A Series of Photographs of the DEC 338 Graphic Display Screen Taken During a Program Run Simulating the Motion of Automobile Hood Linkage Shown in Figures 3.6a and 3.6b . . . . .	44

List of Figures (Continued)

Figure 3.8	Photographs of the DEC 338 Graphic Display Screen Showing Other Systems Simulated by DAMN. . . . .	46
Figure 3.8a.	A Kinematic Two-slider Mechanism Driven by a Constant Velocity Input Motion. . . . .	46
Figure 3.8b.	A One-freedom Four-bar Linkage (No Input Motion). . . . .	46
Figure 3.8c.	A Simplified Two-dimensional Automobile Suspension System. . . . .	47
Figure 3.8d.	A Two-dimensional Simulation of a Washing Machine. . . . .	47



## ABSTRACT

Computer graphics is particularly appropriate for studying the time response of two-dimensional mechanical dynamic systems. This paper discusses the interactive use of a DEC 338 graphics terminal for input and display related to a generalized FORTRAN IV applications program named DAMN--Dynamic Analysis of Mechanical Networks. The DEC 338 is remotely timeshared to the University of Michigan IBM System 360/67 central computer, operating under MTS (Michigan Terminal System).

## 1. INTRODUCTION

This paper deals with the use of computer graphics in an attempt to apply computer-aided design to realistic machinery. Here the term machinery denotes mechanical machinery, including such devices as high-speed card punches, automotive suspensions, washing machines, tractor backhoes, and control linkages. Approximately one tenth of all engineers are concerned primarily with design of this class of machinery.<sup>1</sup> Characteristics of realistic machinery include the following:

1. Large-scale multidimensional movement. In contrast to the vibration of structural systems, the time response of machinery may involve translational movements comparable to the size of the machine itself. Angular coordinates may move through complete rotations. Typically these displacements occur in multiplanar two-dimensional space, although often machinery is designed for motion which can only be described three-dimensionally.
2. Detailed constraint. Most realistic machinery does not conform to the lumped mass-spring-damper model used to explain vibration. Typically parts are interconnected, not only by elastic and damping elements, but by other parts as well. These part-to-part contacts frequently

form complicated loop structures. Moreover the contacts themselves are varied. Some are essentially simple hinges or keyways and are fixed in location relative to the contacting parts. However, others shift location throughout the motion of the machine. An example is the contact between a cam and a follower.

3. Multifreedom dynamics. Despite the extensive constraint present in machinery, it is still necessary to regard it as having more than zero degrees of freedom. (The classical approach to machinery involves just this assumption--that machinery is kinematic in nature, with zero freedom once all input motions are specified.) Multifreedom behavior may be introduced by elastic elements such as timing belts or flexible beams, by motors which lack the power to bring the machine almost instantly up to speed, or by unusual interaction of a normally rigid system (automobile crash).
4. Nonlinear elements. Typically the elements of realistic machinery are not ideal springs and dampers but instead such items as bushings or pneumatic elements. For large-scale motion, these cannot be represented in a single linear model.

5. Intermittency. Often a machine will change its structural identity, perhaps irregularly, at times during its motion. Higher pair contacts may open because of inertial loads, or parts which are normally fixed in position may begin to move due to overload.
6. Multidisciplinary designs. Mechanical designers tend to actively couple electronic, optical, fluid, and other diverse elements in their machine design.
7. Additional. Other deviant characteristics of machinery will typically be observed according to the product with which a machine is associated. For example, thread manufacturing machinery has parts which, in effect, change mass with time as more thread is wound onto a spool.

Effective computer-aided design of any type of engineering system requires at its foundation a generalized program which will accept not only changes in the values of the parameters of a specific device but will also accept changes in the identity of the device itself. As suggested by the foregoing characterization, the development of a true generalized program for computer-aided design of realistic machinery is an imposing task. In fact, the program discussed in this paper is somewhat more limited than the ideal. It accommodates large-scale motion but only for two-dimensional

systems. This, however, sets it apart from most dynamics programs designed for the study of vibration. It accommodates the constraint of part loop closure and therefore applies to two-dimensional kinematic systems such as those covered by KAM<sup>2</sup> and the generalized matrix kinematic programs written at Northwestern University.<sup>3</sup> KAM and the Northwestern program cover three-dimensional systems also, but do not extend to dynamics. Because of problems of detail, the present program is limited in covering higher pairs and nonlinear elements, and has reasonable limits on problem size. But it does provide a working basis for a large fraction of computer-aided machine design. The program algorithms were aimed toward a system representation for machinery similar to that which has been so successful for electrical and structural systems. Because of the complexity previously cited, a complete, consistent identification with either network state variable techniques<sup>4</sup> or bond graph<sup>5</sup> techniques has not yet been made. However the identification is sufficiently close that the program has been named DAMN (Dynamic Analysis of Mechanical Networks).

Computer graphics is a most helpful adjunct in applying the DAMN program to machine design. Graphic output allows the designer a very rapid appreciation of the relation of the several problem coordinates as they perform in two-dimensional space. He can immediately sense the phase relations, accelerations, possible interference problems,

and other features which are obviously difficult to interpret from a collection of individual plots of the variables. Because of the effectiveness of graphic output there is an increased emphasis on the worth of an interactive capability to the designer. The program has been designed to allow immediate change in either the parameters defining the design of a specific system (masses, stiffness coefficients, geometric lengths, etc.) or the topological data which distinguish one system from another.

Figure 1 shows generally the relation between program, remote graphic input/output, the University of Michigan central computer facility, and the DAMN program itself. The DEC 338 display terminal of course includes a PDP-8 computer connected to a graphic display unit. This is connected in full-duplex mode over normal telephone lines to another computer, also a PDP-8, modified for communications use and termed the Data Concentrator. In effect, the graphic terminal can communicate with the DAMN program at any level for flexible simulation of a specific device or set of devices. The DAMN program itself is written entirely in FORTRAN IV. It is resident on disc storage at the University of Michigan Computing Center and is executed on the University's IBM 360/67 computing system<sup>6</sup>. The graphics output program also shown is written primarily in FORTRAN IV, but is importantly dependent on the DF routines developed under the Concomp Project for the support of the

# TECHNICAL ILLUSTRATION

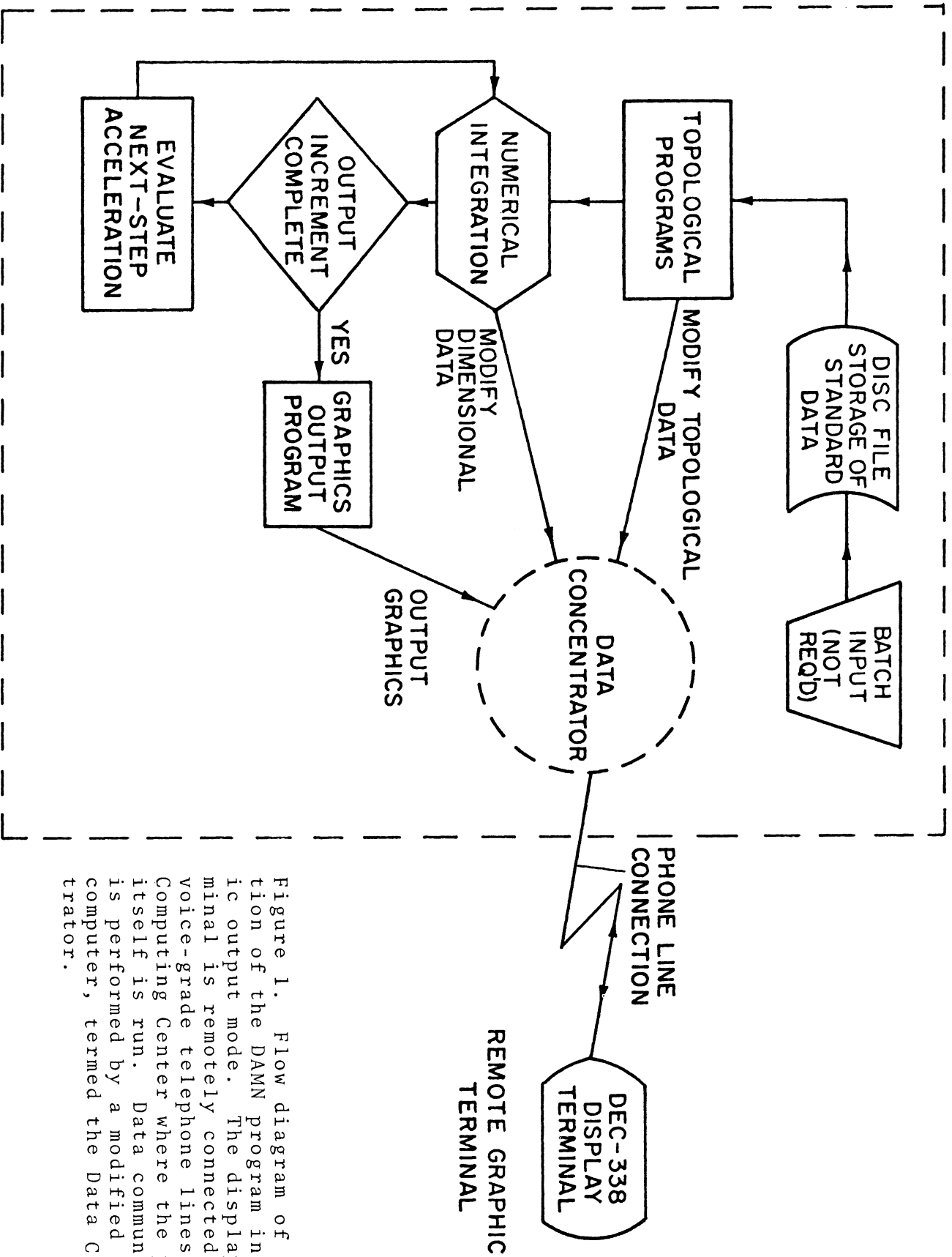


Figure 1. Flow diagram of operation of the DAMN program in graphic output mode. The display terminal is remotely connected by voice-grade telephone lines to the Computing Center where the program itself is run. Data communications is performed by a modified PDP-8 computer, termed the Data Concentrator.

remote graphic terminal.<sup>10</sup> At present, light-pen input is not used in a major way. Instead, desired parameter changes are typed in from a teletypewriter unit on-line to the DEC 338 display terminal.

The DAMN program was created as part of a project sponsored primarily by the Ford Motor Company for the computer analysis of mechanical dynamic systems. The project was substantially assisted also by the Concomp Project, itself sponsored by the Advanced Research Projects Agency of the U.S. Department of Defense. The DAMN program is viewed as a prototype program useful in giving students an appreciation of computer-aided design of mechanical systems and in revealing to them the worth of analytical methods and computer characteristics for subsequent programming of a more advanced, comprehensive system. However, experience with DAMN to date has been very favorable particularly the level of detail and speed with which the response of mechanical systems can be determined, and the characteristics of graphic display of mechanical systems.

In Section 2 the analytical background and program structure of DAMN will be reviewed more specifically. Section 3 will detail the graphic programming and system facilities and discuss experience with specific mechanical systems which have been simulated in graphic computer runs.



## 2. BACKGROUND OF PROGRAM

### 2.1 Constitutive Elements

The DAMN program is designed to represent machinery systems by a simultaneous set of second-order ordinary differential equations. The essential programming problem is the development of the coefficient terms in these equations regardless of the machinery system that has been defined to the program via the input. To facilitate this, a lumped parameter view of machinery is taken in which machinery is seen to consist of eight constitutive elements:

- (1) Parts. Rigid, inextensible objects of zero or finite size, mass, and moment of inertia.
- (2) Fields. Conservative or nonconservative forces applied between one part and another.
- (3) Generators. Motion sources, inputting either a rotational or translational motion.
- (4) Rotations. Contacts allowing a single degree of freedom of rotation of one part relative to another.
- (5) Translations. Contacts allowing a single degree of freedom of translation of one part relative to another.
- (6) Markers. Points, unit vectors, or the combination of a point and unit vector, fixed translationally or in orientation relative to a specified part.

- (7) Keyed Assemblies. Assemblies of one or more parts related entirely by translational contact.
- (8) Loops. Sequences of parts and contacts forming an independent closed chain.

All data input to the DAMN program is categorized under the first six elements. Categorization under the elements "Keyed Assembly" and "Loop" is made internally by the program, but the user is not expected to recognize these elements themselves. The user also specifies the output he desires in terms of the relative displacement, velocity, and acceleration of selected pairs of markers. The program is being developed also to compute reaction force at rotations and translations.

Based on the data input in the preceding categories, the program constructs a set of second-order ordinary differential equations and numerically integrates these through the range of time specified by the user. In general, numerical integration is a slower process than the direct, algebraic computation ordinarily possible for kinematic systems, but it is a practical necessity for large nonlinear systems. Most of the equations constructed correspond to Lagrange equations. One of these is constructed for the angle of each keyed assembly in the system, and one for the displacement of each translation. Thus, the coordinates for two-dimensional mechanical systems are chosen as keyed assembly angles

and translational displacements. Additional differential equations are created to represent constraint.

## 2.2 Definition of Constraint

Figure 2.1 shows a revealing though very academic example of a machine. The coordinates for this system are the angles of the two movable keyed assemblies 2 and 3, plus the three translational coordinates  $s_{11}$ ,  $s_{21}$ , and  $s_{31}$ . For any situation of forces acting between the parts in this example, the system is dynamic with three degrees of freedom. However, the fact that the parts are connected in a closed chain or loop means that the coordinates are interrelated by the following equation.

$$\vec{r}_{21} + \vec{s}_{21} + \vec{r}_{22} + \vec{r}_{31} + \vec{s}_{31} + \vec{r}_{32} + \vec{r}_{11} + \vec{s}_{11} + \vec{r}_{12} = 0 \quad (1)$$

Equation (1) simply states that the sum of part vectors  $\vec{r}$  and translation vectors  $\vec{s}$  around the closed loop is 0. The components of this equation in the directions  $\hat{i}$  and  $\hat{j}$  correspond to two scalar equations in the coordinates  $\theta_2$ ,  $\theta_3$ ,  $s_{21}$ ,  $s_{31}$ , and  $s_{11}$ . When the problem of Figure 2.1 is represented by DAMN, the second derivative of the components of Eq. (1) is included as a pair of constraint differential equations supplementing the Lagrange equations.

As a limiting case the constraint conditions are the conditions normally used to solve kinematic problems (zero

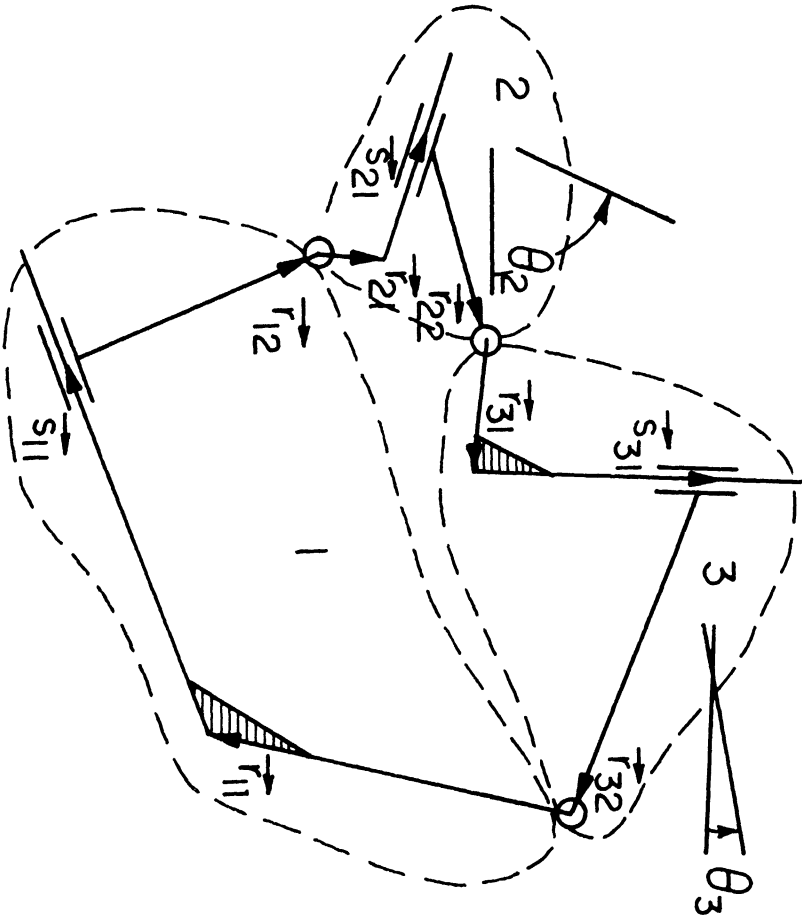


Figure 2.1. Schematic of a hypothetical machine in terms of constitutive elements. A closed loop is formed by three keyed assemblies, each circumscribed by a closed dashed line. The keyed assemblies are connected by rotations (pin joints). In this example within each keyed assembly there are two parts connected by a translation (keyway joint). In later schematics translations are denoted more conveniently by small triangles. The angles  $\theta_2$  and  $\theta_3$  shown are the angles of the keyed assemblies relative to ground.

degree of freedom). For example, in Figure 2.1, if the slips  $s_{21}$  and  $s_{31}$  are prevented and the angle of keyed assembly number 2 is specified as a function of time, the system becomes a kinematic slider crank mechanism, as shown in Figure 2.2. In this case only the constraint equations are necessary to solve the system, either by direct evaluation of algebraic equations as Eq. (1), or by numerical integration of a derivative of Eq. (1).

For a program to simulate three-dimensional machinery another basic constraint condition must be included-- specifically the condition of angular constraint around a closed loop of keyed assemblies.

### 2.3 Dynamic Conditions--Lagrange's Equation

The dynamic condition on which the DAMN program is based is the Principle of Least Action, or Hamilton's Principle. As applied to mechanical systems this states that a system will respond so as to minimize the integral over time, of the work due to all effects, by the surroundings on the system. In equation form this condition is presented as the variation of an integral equal to 0:

$$\delta \int_0^t W(\dot{q}_i, q_i, t) = 0 \quad (2)$$

By use of variational mathematics this condition is developed to the following set of differential equations, one

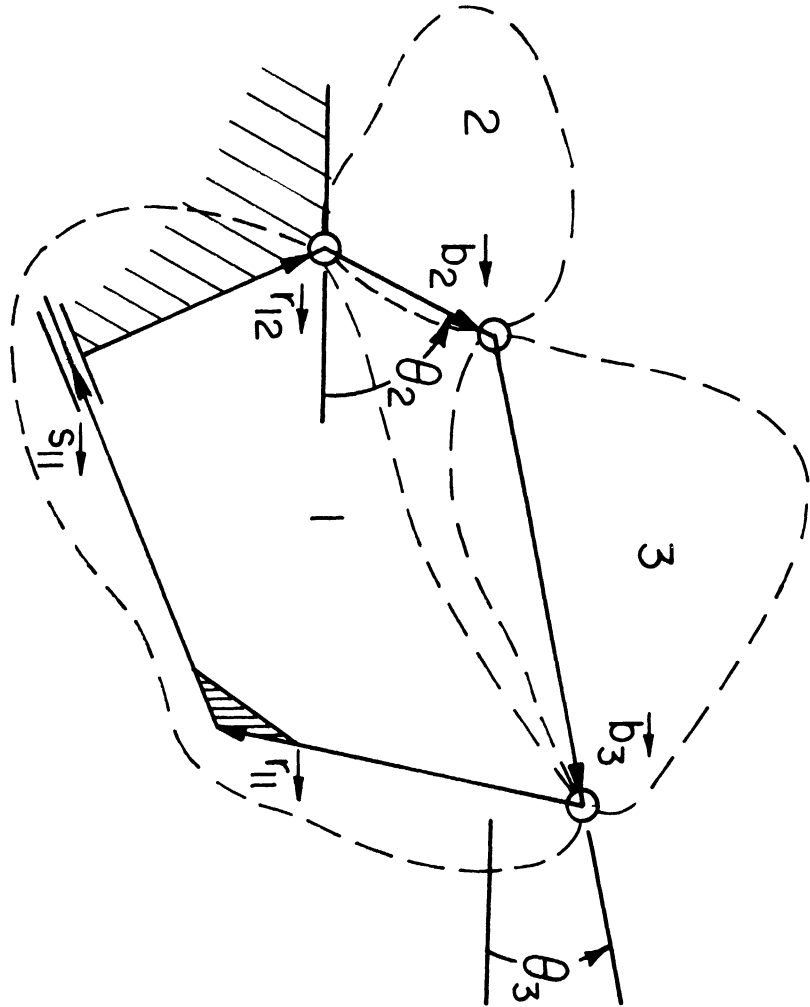


Figure 2.2. Schematic of a kinematic slider-crank mechanism. In this example enough freedom is removed from the loop originally shown in Figure 2.1, that the constraint condition itself (Eq. (1)) is sufficient for determination of system time response, given an input of  $\theta_2$  versus time.

for each coordinate  $q_i$ .

$$\frac{d}{dt} \left( \frac{\partial W}{\partial \dot{q}_i} \right) - \left( \frac{\partial W}{\partial q_i} \right) = 0 \quad i = 1, 2, 3, \dots, m \quad (3)$$

Here, the work  $W$  done by the surroundings on the system includes kinetic and potential energy effects, plus effects due to nonconservative forces such as those of damping. Ordinarily these are the only effects considered. However, a basic assumption of Eq. (3) is that the coordinates  $q_i$  are independent. Because of the loop constraints mentioned in Section 2.2 this condition is not normally met by the coordinates recognized by DAMN for machinery systems. To overcome this difficulty the loops are treated as broken for purposes of Lagrange's equation. Formally, the coordinates  $q_i$  are therefore independent as required. However, the Lagrange equations thereby produced are accompanied by second-order loop constraint equations which as a practical matter insure that the loops remain closed. Also the work terms in the Lagrange equations now include Lagrange multiplier terms for each component of loop constraint. These have the form  $\lambda_j \phi_j$  where the  $\lambda_j$  are reaction forces and the  $\phi_j$  are loop constraint functions. The ultimate form of Lagrange's equation with constraint is the following:

$$\frac{d}{dt} \left[ \frac{\partial (T-V - \sum_{j=1}^{2n} \lambda_j \phi_j)}{\partial \dot{q}_i} \right] - \frac{\partial (T-V - \sum_{j=1}^{2n} \lambda_j \phi_j)}{\partial q_i} = Q_i \quad (4)$$

As it stands Eq. (4) is difficult to program for general application. What is instead required is a set of relations directly of the form of second-order ordinary differential equations. The principal problem here is the complicated set of operations performed on the kinetic energy terms. By viewing a mechanical system in terms of its constitutive elements it is possible to determine general expressions for the Lagrange equation kinetic energy terms, for the mass of one specific part, part m. These expressions are the following:

$$\frac{d}{dt} \left[ \frac{\partial T_m}{\partial \dot{\theta}_\ell} \right] - \frac{\partial T_m}{\partial \theta_\ell} = \delta_{\ell m} I_m \ddot{\theta}_m + m_m \left\{ \left[ \sum_{i=1}^a \sum_{j=1}^{t_i} \theta_i \vec{b}_i \right. \right. \\ \left. \left. - \dot{\theta}_i^2 (\vec{b}_i \times \hat{k}) + \ddot{s}_{ij} (\hat{s}_{ij} \times \hat{k}) + 2 \dot{s}_{ij} \dot{\theta}_i (\hat{s}_{ij}) \right] \cdot \hat{b}_\ell \right\} \quad (5)$$

$$\frac{d}{dt} \left[ \frac{\partial T_m}{\partial \dot{s}_{\ell p}} \right] - \frac{\partial T_m}{\partial s_{\ell p}} = m_m \left\{ \left[ \sum_{i=1}^a \sum_{j=1}^{t_i} \theta_i (\hat{k} \times \vec{b}_i) - \dot{\theta}_i^2 \vec{b}_i \right. \right. \\ \left. \left. + \ddot{s}_{ij} \hat{s}_{ij} + 2 \dot{s}_{ij} \dot{\theta}_i (\hat{k} \times \hat{s}_{ij}) \right] \cdot \hat{s}_{\ell p} \right\} \quad (6)$$



These relations are especially interesting when the resemblance is noted to the general expression for the second derivative of a single vector,  $\ddot{\vec{r}}$ .

$$\ddot{\vec{r}} = \ddot{\theta}(\hat{k} \times \vec{r}) - \dot{\theta}^2 \vec{r} + \ddot{r} \hat{r} + 2\dot{r}\dot{\theta}(\hat{k} \times \hat{r}) \quad (7)$$

In Eqs. (5) and (6) the summation over  $i$  is a summation over all the keyed assemblies in a chain from ground to the part whose mass is being considered. The summations over  $j$  are summations of translational terms within a given keyed assembly  $i$ , included in the chain mentioned. Figure 2.3 illustrates the elements involved in this summation. The vectors  $\vec{b}_i$  are position vectors which span the  $i$ th keyed assembly, from the contact at which the chain enters to the contact at which it leaves. Two equations are required, (5) and (6), because the Lagrange equation must be written both for coordinates which are angles of keyed assemblies  $\theta_\ell$  and coordinates which are displacements of translational contacts,  $s_{\ell p}$ .

As far as the assembly of inertial terms is concerned, at each integrating step DAMN polls each part having finite mass and accumulates coefficient terms as defined by Eqs. (5) and (6).

The other terms in Eq. (4) are much easier to evaluate. The Lagrange multiplier terms  $\lambda_j \phi_j$  result in a matrix of coefficients which is exactly the transpose of the matrix of coefficients obtained in a second-order

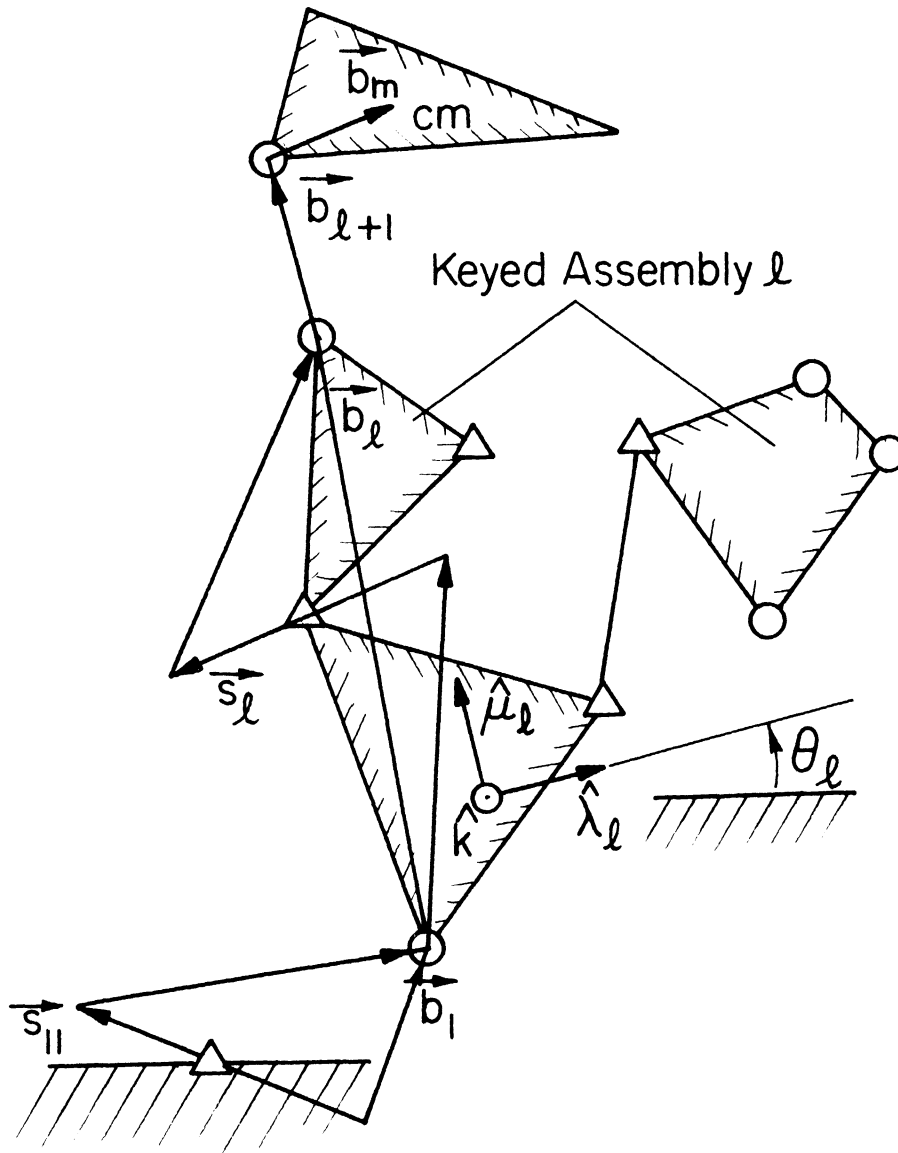


Figure 2.3. A chain of keyed assemblies from ground to the center of mass of a particular part. In constructing the Lagrange equation kinetic energy terms for each part  $m$  (Eqs. (5) and (6)), summations are made over the chain of keyed assemblies to the part center of mass, and over the individual translations within each keyed assembly in the chain.

constraint condition. Forces due to potential energy effects (operations on  $V$ ) are simply included in the collection of nonconservative force terms  $Q_i$ . Table 1 shows the construction of the force terms, differing according to whether the Lagrange equation coordinate is angular or translational and whether the actual applied force is a translational force or a rotational force (torque).

The forces included in the  $Q_i$  term are distinguished as being displacement- or velocity-dependent, not acceleration-dependent. This term therefore includes spring forces, damping forces, and any other force whether linearly or nonlinearly dependent on displacement and velocity. One standard force-type available in DAMN is a "wall force" which has magnitude zero until a specified pair of points, thought of as the terminals of a force field, move within a distance comparable to an integration step of each other. The force function then increases very rapidly for very small decrease in point separation. In effect the points are prevented from exact coincidence. This particular force type provides a basis for including a wide variety of intermittent mechanisms and impact situations within DAMN.

At present DAMN is not organized to accept a complete variety of one-of-a-kind forces. Such acceptance requires allowing the user to input force functions in either subroutine or tabulated form, certainly a feasible extension

TABLE 1

Form of the Generalized Force Terms  
in the Lagrange Equation

Type of Applied Force	Displacement Coordinate, $s_{\ell p}$	Angular Coordinate, $\theta_{\ell}$
Translational Force $\vec{F}_k$	$\vec{F}_k \cdot \hat{s}_{\ell p}$	$\vec{F}_k \cdot (\hat{k} \times \vec{b}_{\ell})$
Rotational Force (torque) $\vec{\tau}_k$	zero	$\vec{\tau}_k (\delta_{i\ell} - \delta_{j\ell})$

$\vec{F}_k \equiv k$ th actual applied translational force in the system

$\vec{\tau}_k \equiv k$ th actual applied rotational force (torque) in the system, applied between a pair of parts  $i$  and  $j$

$\theta_{\ell} \equiv$  angle of the  $\ell$ th keyed assembly

$s_{\ell p} \equiv$  the  $p$ th translation within the  $\ell$ th keyed assembly

of the program. A similar restriction now exists for one-of-a-kind motion for generators and part contacts dependent on surface profile. When the reaction force computation capability is completed DAMN will accept applied forces which are themselves dependent on reaction force (Coulomb friction force). Meanwhile this force type is excluded.

The essential form of the dynamic and constraint conditions necessary to DAMN can be represented as an Eq. (8):

$$\begin{array}{|c|c|c|c|}
 \hline
 \begin{array}{c} A_{11} \quad A_{12} \quad \dots \quad A_{1m} \\ \vdots \quad A_{22} \\ \text{motion} \\ \text{matrix} \end{array} & & \begin{array}{c} C_{11} \quad C_{21} \quad \dots \quad C_{n1} \\ C_{12} \\ \vdots \\ \text{Lagrange mult.} \\ \text{coefficients} \end{array} & \begin{array}{c} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_m \end{array} \begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_m \end{array} \\
 \hline
 \begin{array}{c} A_{m1} \quad \dots \quad A_{mm} \\ C_{11} \quad C_{12}' \quad C_{1m} \\ C_{21} \\ \vdots \\ \text{constraint} \\ \text{equation} \\ \text{coefficients} \end{array} & & \begin{array}{c} C_{1m} \quad \dots \quad C_{nm} \\ \text{all zeros} \\ C_{nm} \end{array} & \begin{array}{c} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{array} \begin{array}{c} g_1 \\ g_2 \\ \vdots \\ g_n \end{array} \\
 \hline
 \end{array} \quad (8)$$

Here the differential equations are represented as a set of simultaneous linear algebraic equations in the variables  $\ddot{q}_i$  and  $\lambda_j$ . The coefficient matrix is partitioned in four submatrices. The upper left submatrix contains the coefficients  $A_{ij}$  of the second-order acceleration terms from Lagrange's equation. Specifically these are radial and tangential acceleration coefficients. The main diagonal coefficients of this matrix are constants, and the submatrix is symmetric. However, the off-diagonal terms are nonlinearly dependent on problem coordinates, specifically the angles of keyed assemblies. The upper right and lower left submatrices happen to be the transpose of each other, although the lower left submatrix is formed directly from the second-order terms of the constraint conditions, whereas the upper right submatrix is composed of Lagrange multiplier coefficients from the Lagrange equations themselves. These coefficients are also nonlinearly dependent on the angles of the keyed assemblies. The column vector of dependent variables includes the second derivatives of the keyed assembly angles plus the second derivatives of the translational variables. The lower subvector includes the Lagrange multiplier terms which originate from Lagrange's equation. On the right is a column vector of elements which themselves are large summations of displacement- and velocity-dependent forces, applying to the particular Lagrange or

constraint equation identified with their row. None of these terms however is dependent on coordinate second derivatives.

The actual integration process takes place as outlined in the lower left of Figure 1. First, the initial conditions are used to evaluate all of the coefficients in Eq. (8) at time equal zero. Then Eq. (8) is solved as a simultaneous set of linear algebraic equations for the  $\ddot{q}$ 's and the  $\lambda$ 's. In concept the  $\ddot{q}$  terms are numerically integrated to determine corresponding velocity and displacement, then fed back to compute the coefficient terms in Eq. (8) for a next step. At present the integrating program used is HPCG (Hamming Predictor-Corrector Generalized) available from the IBM Scientific Subroutine Package<sup>7</sup>. The integration takes place in small steps with a simultaneous equation solution in each step until a user-specified output increment is completed. At that time current values of all problem coordinates are transmitted to the graphics output program and processed there for immediate display, or if a printed output has been elected, the results are stored on disc until the end of the entire run, at which time they are printed out.

The integration process outlined here is simple, but not the most efficient process possible. For example, it is probably feasible to integrate without having to solve Eq. (8) at every integrating step. Also, with this

particular set of variables and the Lagrange multiplier technique, the number of differential equations exceeds the degree of freedom of the system by the amount  $2c$ , where  $c$  is the number of scalar constraints. Alternate approaches may, in fact, yield a smaller number of equations in the set. However, the Lagrange equation approach has been especially easy to interpret and free from artificial singularity problems because of the close physical correspondence of each of the Lagrange and constraint equations. For the size of problems considered so far (order  $10 \times 10$ ), integration speeds have been more than adequate for analysis, as discussed in Section 3. However, use of the program for computer-controlled synthesis will require at least an order of magnitude increase in speed—probably feasible.

#### 2.4 Relation of Topological and Dimensional Programs

We have made careful distinction between topological and dimensional subroutines within DAMN. Here, the term "topological programs" denotes those which determine information which is independent of part dimensions, or any system displacements or forces. A good example of a topological program is the subroutine which determines the independent loops of parts and contacts in an arbitrary machine system. The results of this program are, of course, unaffected by the state of the machine displacement. As



shown in Figure 1, the program was designed so that all computations which are topological in nature are performed before the beginning of numerical integration. The results of these computations are stored as integer information in a set of program tables, termed plex tables. In essence, one such table exists for each of the constitutive elements defined in Section 2.1. These plex tables are held in memory throughout both the topological and dimensional programs. The information contained considerably increases the speed with which coefficient values can be determined for the numerical integration process.

The DAMN program is network-based in the sense that most of the topological routines use network mathematics corresponding closely to that used in electrical or structural analysis problems. For example, the minimum number of independent part-contact loops is determined by processing a branch-node incidence matrix where parts are interpreted as nodes and contacts as branches.

### 3. GRAPHICS SYSTEMS AND APPLICATIONS

#### 3.1 Systems Viewpoint

DAMN is a set of FORTRAN IV subroutines which are executed in either batch or timeshared mode on the University's IBM 360/67 central computer. All such programs are under the control of a supervisory program named UMMPS (University of Michigan Multiprogramming System) which supervises a timesharing monitor, MTS (Michigan Terminal System), and an auxiliary batch monitor HASP (Houston Automatic Spooling Priority System). HASP assigns priorities to batch jobs. In general, the user does not have access to UMMPS or HASP and must communicate to the central computer through MTS by means of the MTS command language.<sup>8</sup> Figure 3.1 shows the interrelation of these programs to the user. Table 2 outlines the important functions of UMMPS, HASP, and MTS.

Currently MTS can service about 75 users in a timeshared mode. For maximum efficiency, batch jobs are first assigned a priority by HASP. These are then passed to MTS, highest priority first, for execution. Batch jobs are processed by three IBM 2540 card read-punch units, one IBM 2501 card reader, and five IBM 1403 line printers. Terminal jobs utilize standard IBM and AT&T communications equipment and are serviced by one IBM 2702 and one IBM 2703 communications terminal.

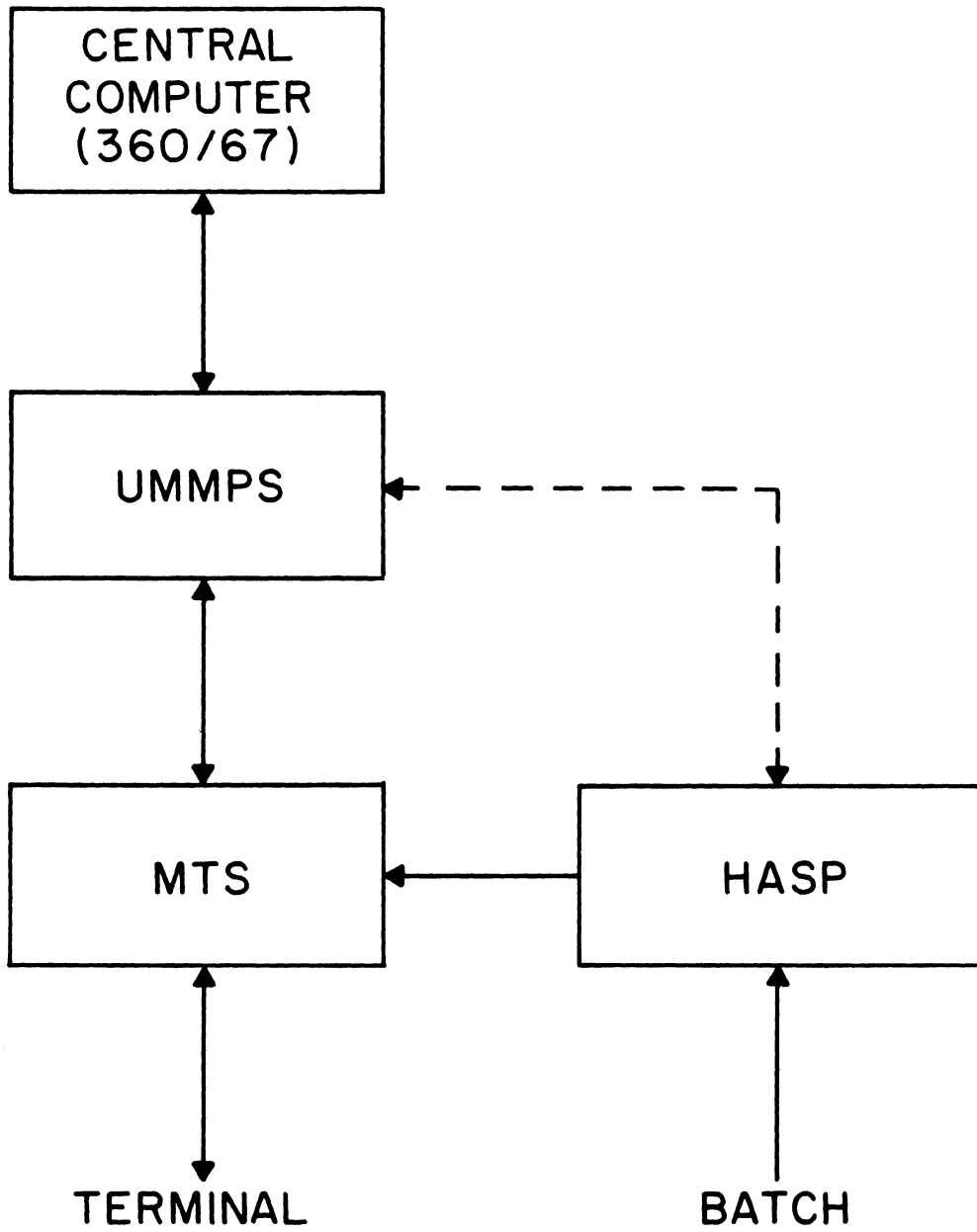


Figure 3.1. Relation of U of M system software in the execution of batch and timeshared jobs on the 360/67 central computer. Terminal jobs are executed according to MTS commands. Batch jobs are first assigned a priority by HASP, then executed in timeshared coordination with terminal jobs. Overall control is accomplished by UMMPS.

TABLE 2

Functions of the Supervisory Programs of  
The University of Michigan IBM 360/67

- UMMPS (University of Michigan Multiprogramming System). The system program which supervises the timesharing system. Inaccessible to the general user. Interfaces the 360 hardware to MTS. Controls the loading and execution of user's programs and I/O, and processes interrupts.
- MTS (Michigan Terminal System). User-oriented system program. Contains assemblers, translators, and system programs which the user accesses by issuing the appropriate command. Monitors both terminal and batch jobs.
- HASP (Houston Automatic Spooling Priority System). Interacts with UMMPS to assign priority to batch jobs for effective execution of programs. Control is eventually passed on to MTS to actually execute program. Inaccessible to general user.

In the U of M communications network, a special control unit named the Data Concentrator can be accessed by any terminal user. The Data Concentrator is itself a PDP-8 computer containing four 4096-word banks of storage and a 1.5-microsecond access core memory. It is controlled by a supervisory program called RAMP. The Data Concentrator was developed to adapt special terminal equipment to the timesharing system, and its applications include interfacing certain non-IBM special-purpose communications equipment to MTS.<sup>9</sup> For DAMN it interfaces a DEC 338 CRT display unit to the central computer, making possible the graphic display feature. Teletype communication to the central computer through the Data Concentrator is essentially the same as through the IBM equipment.

### 3.2 The Graphic Terminal

The graphic terminal used for DAMN output is a DEC 338 CRT display. The U of M Concomp Project developed the necessary support routines to interface the DEC 338 to the central computer. A user communicates with the central computer over voice-grade telephone lines by means of an AT&T 201 data phone. This configuration provides the user with some of the speed advantages of the central computer, coupled with the medium-cost graphics ability of the 338.

The 338 also serves as a remote graphics terminal time-shared to MTS at the Ford Motor Company in Dearborn, Michigan (25 miles distant), and the Whirlpool Corporation in Benton

Harbor, Michigan (160 miles distant). These terminals have the same capabilities as those located at the University and provide industry an unusually direct association and utility with programs of their interest, such as DAMN.

### 3.3 Description of DF Routines

Concomp personnel also developed a comprehensive set of FORTRAN-callable subroutines, named the DF routines<sup>10</sup> (Display File routines), which, when executed on the central computer, construct DEC 338 code to draw lines, display labels, and recognize figures through light pen detection. After construction, the code is transmitted through a 201 data set to the 338, which displays the file until a change is initiated by the user's control program. In constructing a display file, control program tasks include (1) creating a construction buffer, (2) constructing figures by use of lines and labels, and (3) transmitting the file with an appropriate name.

Other applications of the DF routines on the graphic terminal include construction of surfaces, design of electric circuits, and simulation of arterial fibrillation. The DF routines are also used in a number of computer-oriented courses at The University of Michigan. Table 3 gives a brief description of the DF routines used by the DAMN program.

### 3.4 DAMN User Graphics Options

The primary objective of the DAMN graphics output program is the construction and display of a schematic diagram

TABLE 3

Definitions of DF Routines Used in the DAMN  
Graphics Output Program

DFINI	Initializes construction buffer in central computer. All code generated by subsequent calls of DF routines will be placed in this buffer (display file).
DFXYC	Draws one line in display file.
DFLAB	Constructs one label in display file.
DFOCT	Inserts a user-specified four-digit octal number into construction buffer. Used by graphics output program to change intensities of lines.
DFPJM	Allows a program to push-jump or call on a display file which is defined as a subroutine. In the graphics output program, three symbols are defined in this way.
DF201	The subroutine to transmit the display file over the AT&T 201 data phone to the DEC 338.
DFSN	Erases the DEC 338 screen.

of a mechanical system. For this purpose the DAMN user specifies the lines, symbols, vectors, and title to be drawn, and the output program calls on the appropriate DF routines. The DAMN user has the option of defining whatever lines, vectors, or symbols he wishes. Other information which must be specified in the data are scale factors and a time increment which determines when the next frame will be shown. The options available to the graphics user and the information required to generate the display are summarized in Tables 4 and 5 respectively.

### 3.5 Organization of Graphics Output Program

Figure 3.2 is the flowchart for the graphics output program and Figure 3.3 shows example output. To construct the frames, the DAMN graphics output program uses two display files. Static labels such as the title and headings for velocity, acceleration, and time are placed in the first display file. As an example, labels are shown in Figure 3.3 in the upper right side of the frame. These labels are constructed, transmitted, and displayed on the first pass of the output program and are branched over on subsequent output steps. Also constructed in the first pass are the three symbols  $\bigcirc$ ,  $\oplus$ , and  $\triangle$ , and the labels for the vector heads. The vector heads are constructed by concatenating the EBCDIC marker identification number to



TABLE 4

Graphic Options and Description of Output  
Available to DAMN User

- Lines. There are two types of lines, dark and light intensity. By convention, dark lines or a polygon of dark lines represent rigid parts, and light lines represent axes, springs, paths of translation, and any line which is not descriptive of a rigid part. A line is input by specifying its two endpoint markers and its intensity.
- Symbols. The user has available three symbols: a 1/4" circle, a 1/4" circle crossed in the middle, and a 1/4" triangle. In this program the crossed circle represents a rotation which contacts a movable part to ground, a circle represents a rotation between two movable parts, and a triangle represents a relative translation. The symbol positions are defined by specifying the marker I.D. number for the marker at which the symbol is fixed.
- Vectors. Velocity and acceleration vectors of any marker relative to ground can be displayed.
- Title. A title of up to 48 characters can be displayed.

TABLE 5

Description of Data Necessary for Generation  
of DAMN Graphic Display

<u>Base.</u>	The user must specify the ground marker I.D. number defined in the analysis data and its position on the screen. The ground coordinates of all other markers are referenced to this.
<u>Scale Factors.</u>	Scale factors are required to predict the maximum dimensions of the system, and the velocity and acceleration vectors.
<u>Time Increment.</u>	The interval of time between display of each new frame must be specified.

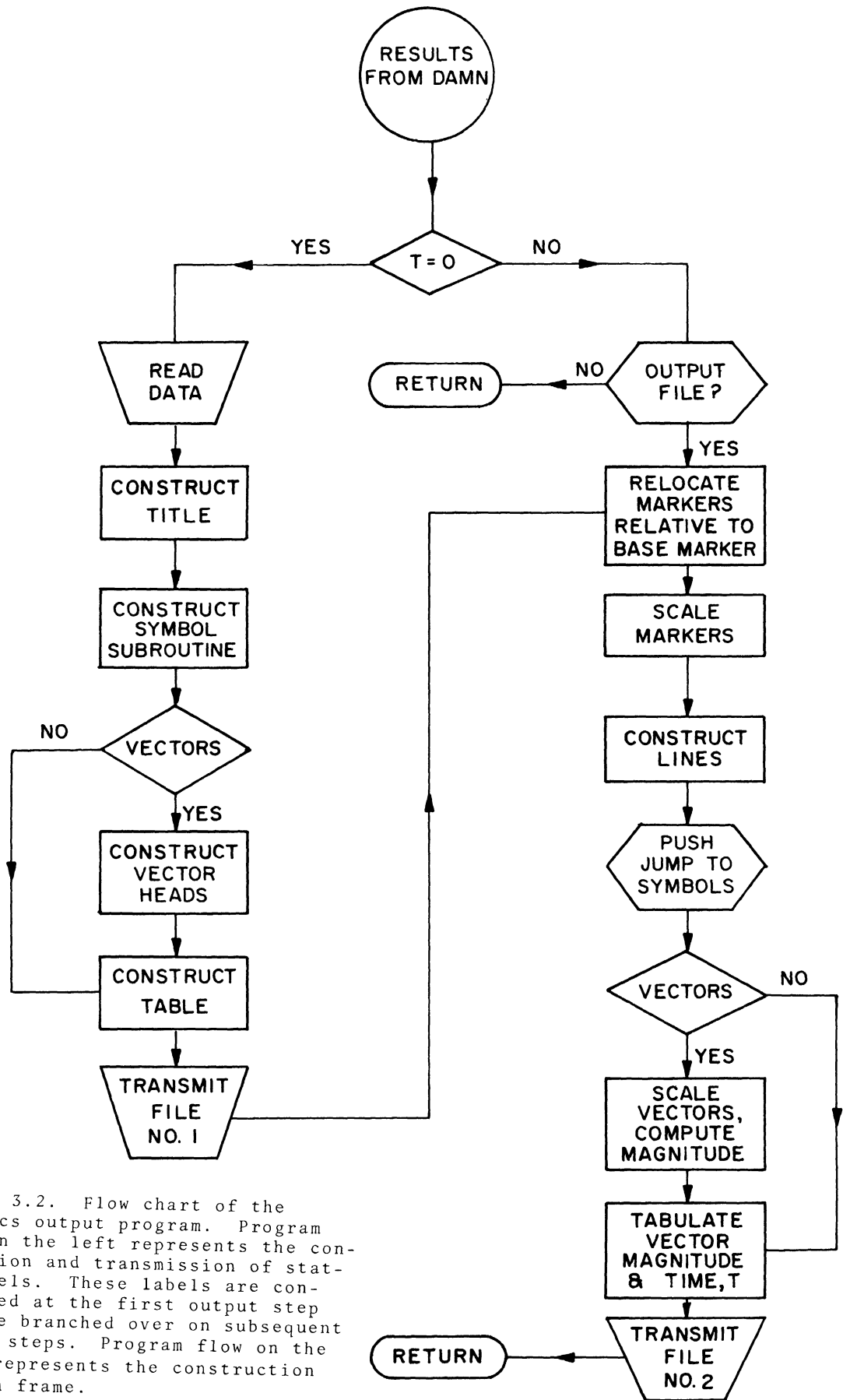


Figure 3.2. Flow chart of the graphics output program. Program flow on the left represents the construction and transmission of static labels. These labels are constructed at the first output step and are branched over on subsequent output steps. Program flow on the right represents the construction of each frame.

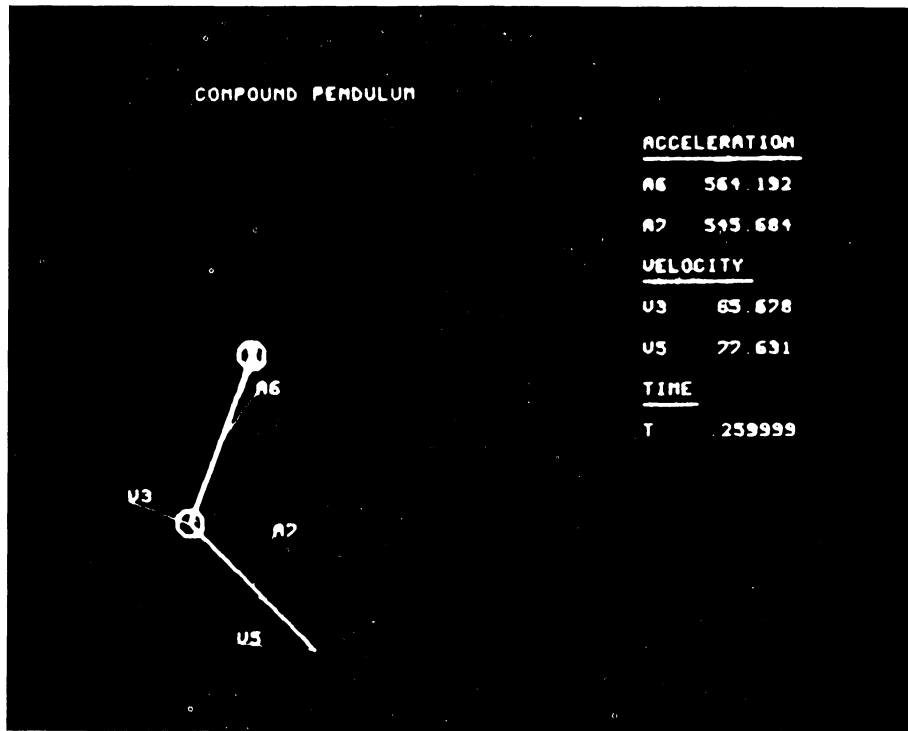


Figure 3.3. Example graphics output. Rigid parts are represented by straight lines or polygons. Any number of velocity and acceleration vectors can be shown. In this example two velocity vectors (symbols V3 and V5) and two acceleration vectors (symbols A6 and A7) are shown. Generators and fields are not displayed.

the EBCDIC letters A and V and placing the result in an array.

The second display file contains the lines which describe the parts in the system, symbols, and velocity and acceleration vectors including their respective magnitudes. The position of the lines (parts) and the direction and magnitude of vectors change with each frame and must be redrawn after each output from the main DAMN program. The output program constructs the system by recognizing key markers, such as endpoints of lines, relocating these markers relative to a base ground marker, and scaling all markers down to screen size. The output program then calls appropriate DF routines to draw the specified lines and symbols. As an example, Figure 3.3 is a schematic of a compound pendulum of two links. Two lines were drawn to describe the links and two symbols were drawn to represent rotations between one link and ground and between the two links, respectively.

The output program must also construct velocity and acceleration vectors if any are requested. Examples of velocity and acceleration vectors are shown in Figure 3.3. Again, the output program must recognize, relocate, and scale the motions of key markers. The magnitude and direction of the vectors are computed in the main DAMN program, and the output program has only to scale these vectors and call on the appropriate DF routines.

Additionally, the output program must access the EBCDIC label for vector heads and call a DF routine to add them to the display. Magnitudes of vectors must also be converted from floating point numbers to their EBCDIC equivalent and tabulated on the upper right-hand side of the screen. Finally the second display file is transmitted to complete the process. The output program then returns to the analysis program to await the coordinates for the next output step.

The timing and quality of the display are highly dependent on the number of options the user chooses, the number of people who are timesharing simultaneously, and the number of dependent variables that the analysis program must solve. Average timing experience is cited in the examples which follow.

### 3.6 Description and Examples of Output

Example 1. Two degrees of freedom, unconstrained, dynamic compound pendulum.

Figure 3.4a shows a sketch of a compound pendulum with two degrees of freedom. Each link has mass and moment of inertia. The schematic representation of this problem is shown in Figure 3.4b. Parts 2 and 3 initially have zero velocity and are acted upon by an unbalanced gravitational force. Figure 3.5 is a series of photographs of the compound pendulum as it is seen on the DEC 338. Labels V3 and V5 define the magnitudes and

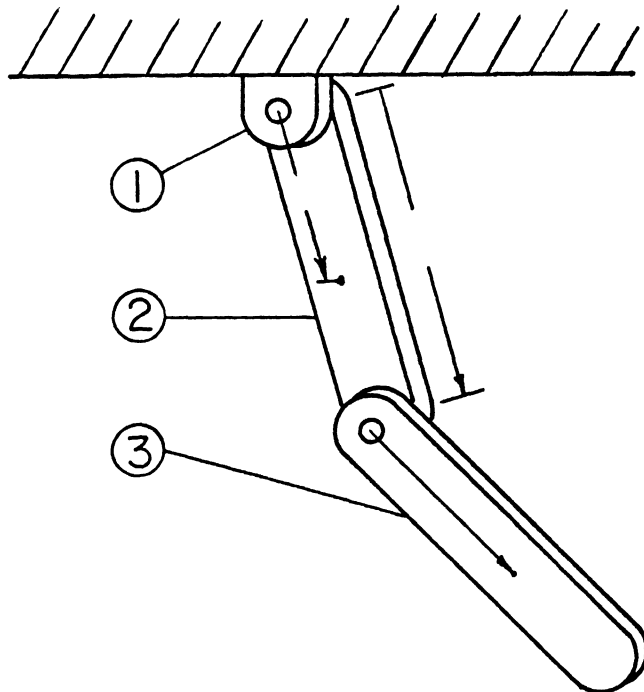


Figure 3.4a. Three-dimensional view of a two-link compound pendulum moving in a gravitational field. Part 1 is ground and parts 2 and 3 the two links. Both parts 2 and 3 have appreciable mass and moment of inertia, and undergo arbitrarily large displacements. This is a two-freedom unconstrained dynamic system.

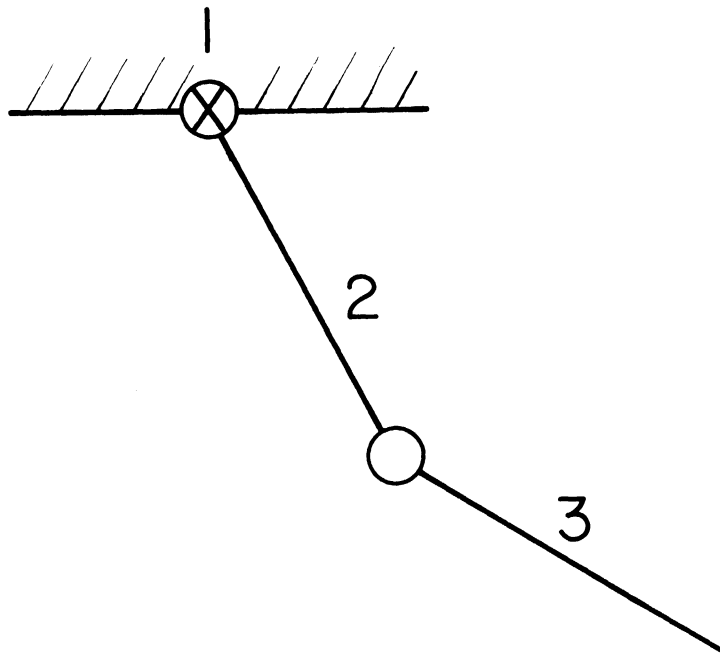


Figure 3.4b. Two-dimensional schematic of the compound pendulum shown in Figure 3.4a.



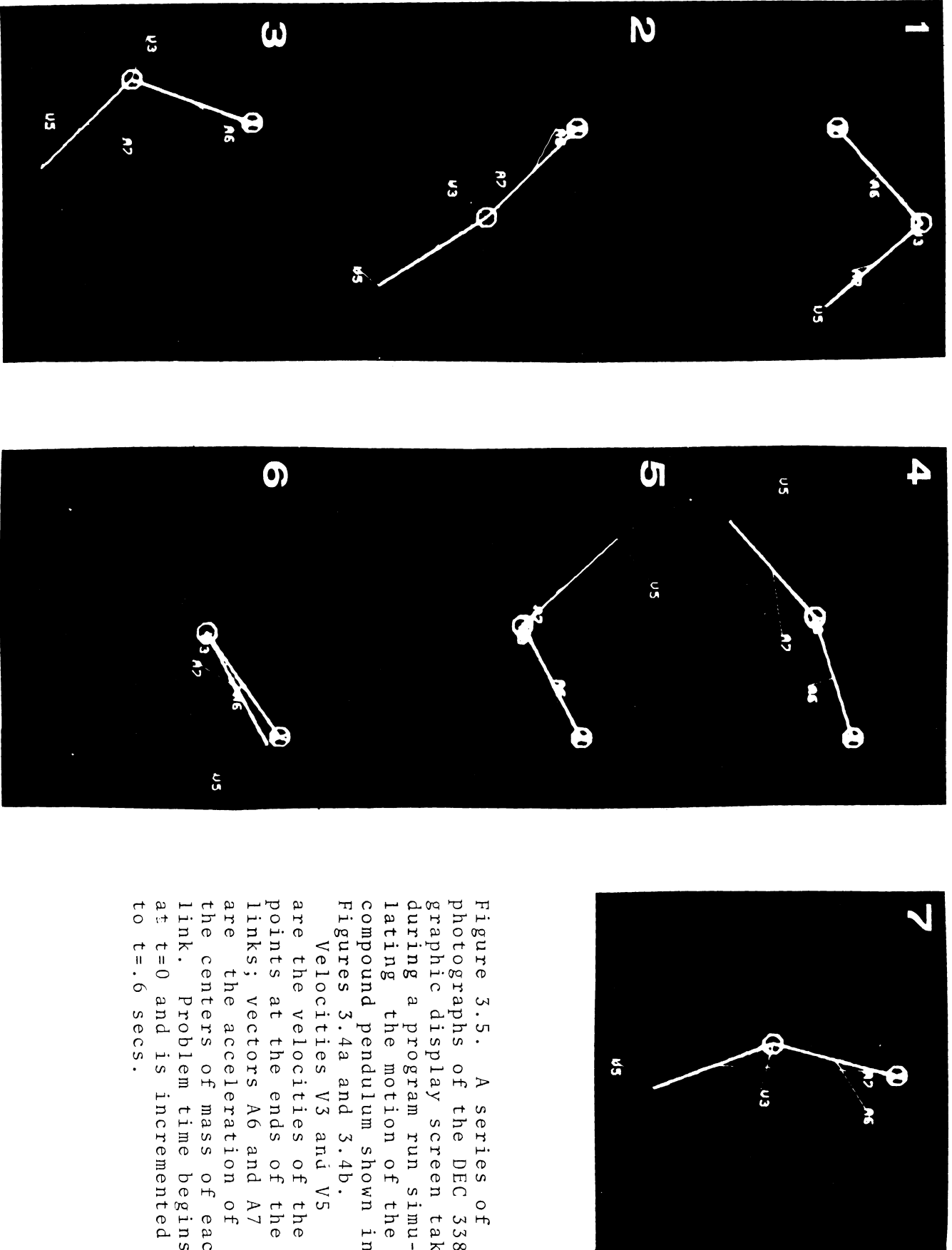


Figure 3.5. A series of photographs of the DEC 338 graphic display screen taken during a program run simulating the motion of the compound pendulum shown in Figures 3.4a and 3.4b. Velocities V3 and V5 are the velocities of the points at the ends of the links; vectors A6 and A7 are the acceleration of the centers of mass of each link. Problem time begins at  $t=0$  and is incremented to  $t=.6$  secs.

directions of the velocities of the ends of the links. Vectors  $A_6$  and  $A_7$  are the acceleration vectors of the mass centers of the links. For this problem DAMN must integrate a system of two second-order differential equations to obtain the velocity and position of each link. The DAMN output program draws two lines and two symbols, solves for four vectors and their magnitudes, and displays 37 frames per minute at approximately 25% MTS system load capacity.

#### Example 2 Automobile hood linkage

Figure 3.6a is a drawing of a simplified automobile hood linkage, and a schematic of this linkage is shown in Figure 3.6b. This linkage consists of five moving parts and ground (part 1). Parts 2-6 all have mass and mass moment of inertia. A spring is stretched between parts 5 and 6, with sufficient strength to raise the hood. Also a torsional spring and a viscous damper are placed between parts 1 and 2.

Figure 3.7 is a series of pictures taken at regular time intervals during the motion of this system. As shown, the oscillation decays rapidly due to the damping. Only one vector, the velocity vector  $V_{17}$ , is shown, representing the velocity of the center of mass situated at the center of the hood (only the rear half of the hood is shown). A system of nine second-order differential equations is solved

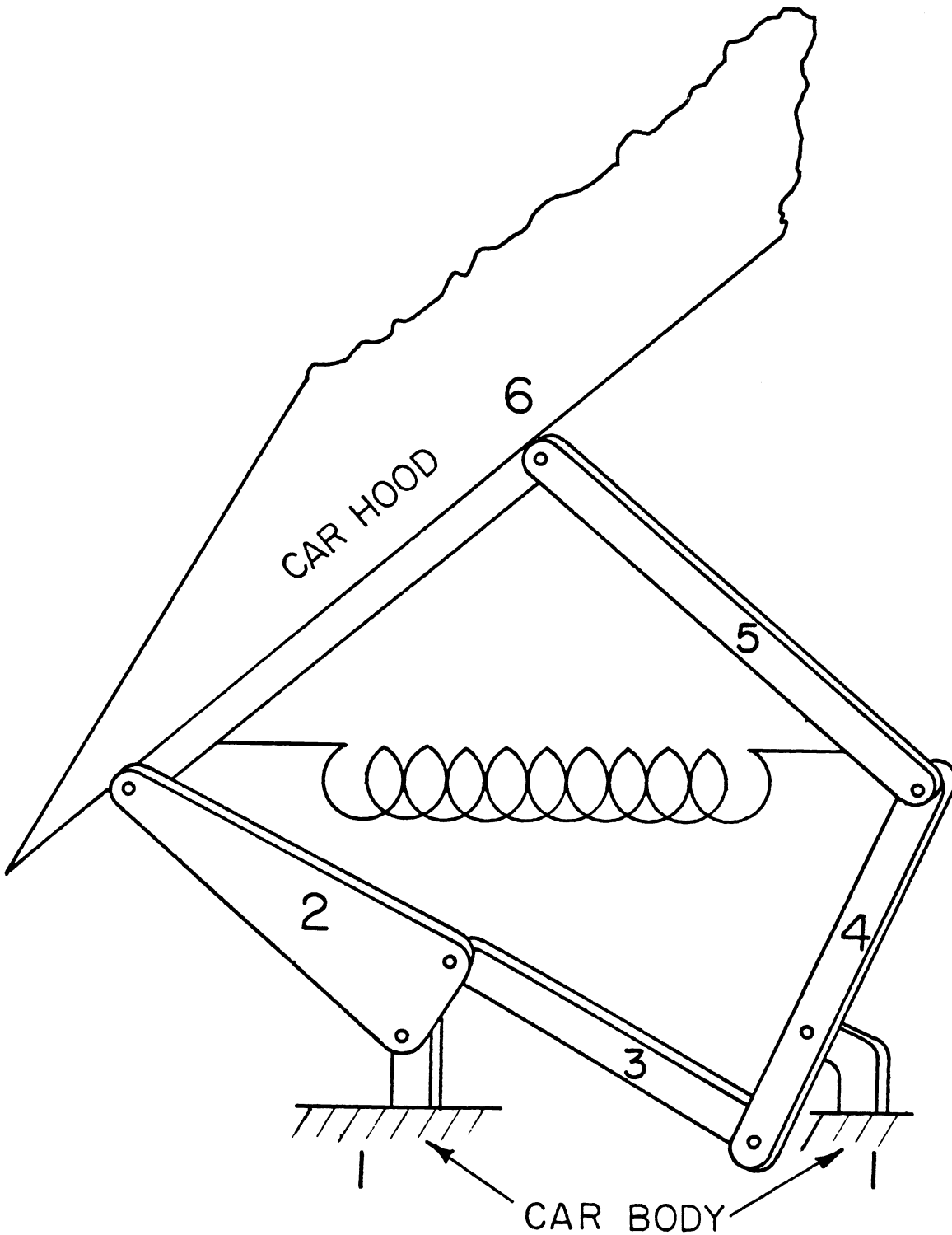


Figure 3.6a. Automobile hood mechanism used in connecting the hood to the body. The mechanism consists of six parts. Part 1 is the automobile body and part 6 the hood. A spring is stretched across parts 5 and 6 and is sufficiently strong to raise the hood from a near-horizontal position. The motion of the hood is impeded by a rotational viscous damper (not shown) between part 1 and part 2. This is a single-freedom, constrained dynamic system.

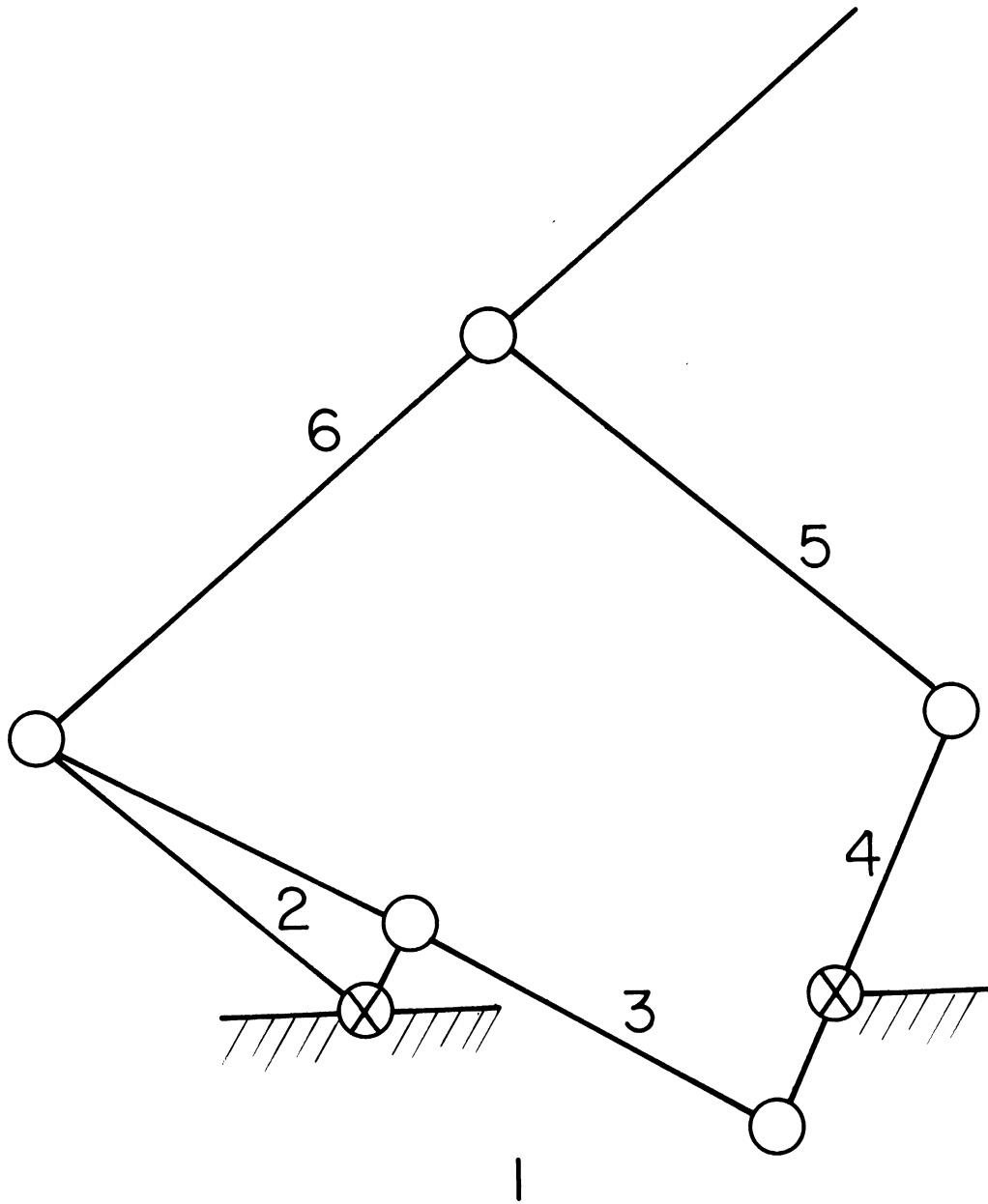


Figure 3.6b. Two-dimensional schematic representation of the hood mechanism of Figure 3.6a. Rotations (pin joints) are represented by small circles; parts are represented by straight lines and polygons.

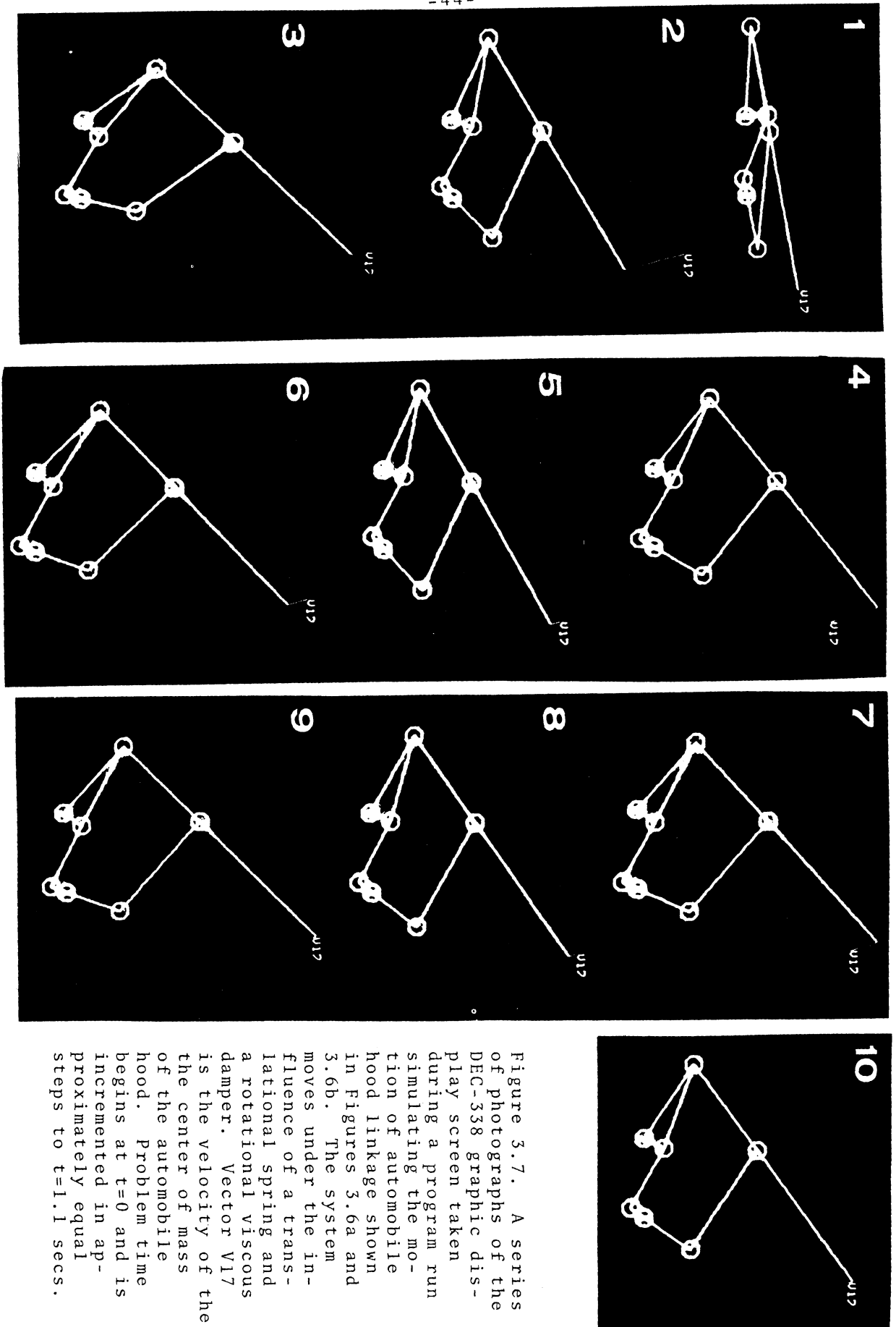


Figure 3.7. A series of photographs of the DEC-338 graphic display screen taken during a program run simulating the motion of automobile hood linkage shown in Figures 3.6a and 3.6b. The system moves under the influence of a translational spring and a rotational viscous damper. Vector  $V_{17}$  is the velocity of the center of mass of the automobile hood. Problem time begins at  $t=0$  and is incremented in approximately equal steps to  $t=1.1$  secs.

as described in Section 2 and the output program displays 7 lines, 7 symbols, and the velocity vector.

#### Other Examples

Figures 3.8a through 3.8d show other systems which have been simulated by DAMN. These examples show that DAMN can accept a wide variety of situations such as rotational and translational contact, kinematic constraint, multifreedom dynamic systems, constrained dynamic problems, inertially driven problems, etc. These and other examples also showed that the number of frames/min. was more often dependent on data transmission time than on problem integration time. Experience has also shown that graphics is particularly useful in analyzing systems which undergo large deflections and is less interesting for small-scale vibration.

#### 3.7 Man-Machine Interaction

An additional feature of the DAMN graphics output program is man-machine interaction through the use of an interpretive command language. This is not yet fully developed, but at present a user can interrupt the display at any point and, by issuing an appropriate command, terminate the program, access a new data set, alter the display, or change system parameters. When the user wishes to modify the display he first issues additional subcommands. Through these subcommands he adds or deletes

Figure 3.8. Photographs of the DEC 338 graphic display screen showing other systems simulated by DAMN.

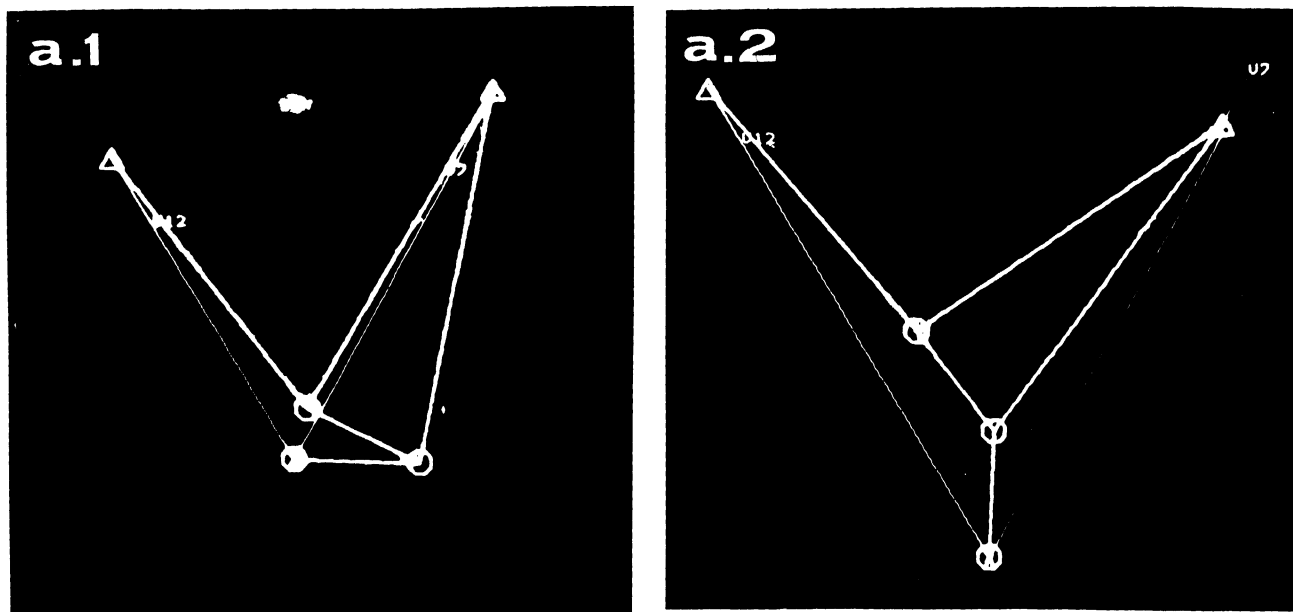


Figure 3.8a. A kinematic two-slider mechanism driven by a constant velocity input motion. Vectors  $V_7$  and  $V_{12}$  are the velocities of the sliders.

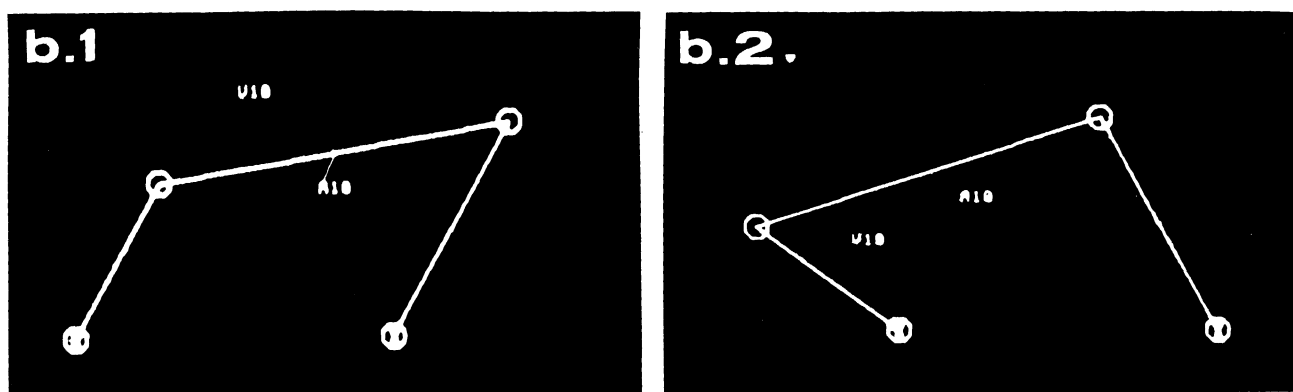


Figure 3.8b. A one-freedom four-bar linkage (no input motion). Each link has mass, moment of inertia, and initial velocity. Vectors  $V_{10}$  and  $A_{10}$  are the velocity and acceleration of the mass center of the coupler.

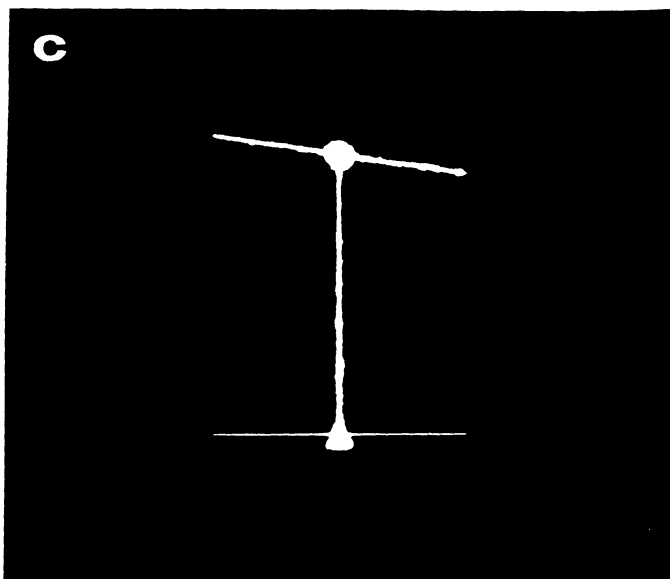


Figure 3.8c. A simplified two-dimensional automobile suspension system. The canted line is the frame, supported vertically by two springs (not shown), one at each end. The vertical line shows vertical displacement of the center of mass, and provides a reference for a rotational damper (not shown).

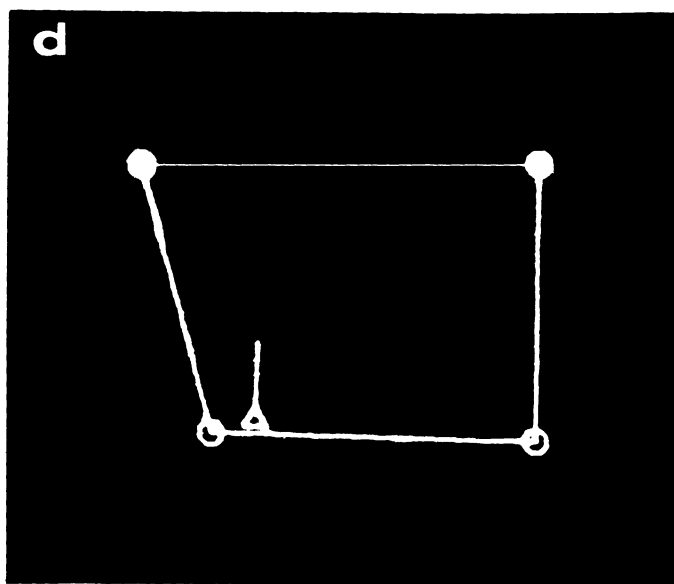


Figure 3.8d. A two-dimensional simulation of a washing machine. The rotating tub is represented as a sinusoidal oscillator mounted on a base-plate suspended from the frame by links. The effect of an eccentric mass in the tub is represented by specifying a mass for the oscillator.



lines, symbols, or vectors, alters scale parameters, and changes the frame increment. This feature allows the user to add a temporarily significant vector for a particular interval of time and then delete it to simplify the display. As planned, the command language will allow the user to add or delete fields or generators, and to alter part geometry and field or generator parameters, then restart the system, and examine the effects of the change.

## ACKNOWLEDGMENTS

The authors gratefully acknowledge the interest and financial support of particularly the Advanced Analytical Technology Department, Ford Motor Company, and the Concomp Project--itself sponsored by the Advanced Research Projects Agency of the U.S. Department of Defense. The support of the structural Dynamics Research Corporation and the Whirlpool Corporation is also appreciated. The project would have been impractical without the hardware and systems software facilities of the University of Michigan Computing Center and the graphic systems software support completed by Dr. Alfred Cocanower and the staff of the Concomp Project.

## REFERENCES

1. Survey performed in 1967 by the Engineers Joint Council under contract to the National Science Foundation; published in Engineer, March-April, 1969.
2. IBM Application Description Manual, H20-0493, Program Information Department, IBM Corporation, Hawthorne, N.Y.
3. Uicker, J.J., Jr., Denavit, J., and Hartenber, R.S., "An Iterative Method for the Displacement Analysis of Spatial Mechanisms," Journal of Applied Mechanics, Series E, Vol. 87, 1965, pp. 309-314.
4. Stern, Thomas E., Theory of Nonlinear Networks and Systems, Addison-Wesley, Reading, Mass., 1965.
5. Karnopp, Dean, and Rosenberg, Ronald C., Analysis and Simulation of Multipart Systems, M.I.T. Press, Cambridge, Mass., 1968.
6. Carnahan, Brice, and Wilkes, J.O., Introduction to Digital Computing and FORTRAN IV with MTS Applications, 1968 (available from Ulrichs Book Store, Ann Arbor, Michigan).
7. IBM Manual H20-0205-2, System/360 Scientific Subroutine Package (360A-CM-03X), Version II.
8. MTS Manual, Computing Center, University of Michigan, Ann Arbor, 1967.
9. Mills, D.L., The Data Concentrator, Technical Report 8, Concomp Project, University of Michigan, May 1968, 113 pp.
10. Cocanower, A.B., The DF Routines User's Guide, Memorandum 23, Concomp Project, University of Michigan, May 1969, 5 pp.

Security Classification

DOCUMENT CONTROL DATA - R & D		
<i>(Security classification of title, body of abstract and indexing information must be identical with the overall report is classified)</i>		
1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION	
CONCOMP PROJECT UNIVERSITY OF MICHIGAN	2b. GROUP	
3. REPORT TITLE		
COMPUTER GRAPHICS IN THE DYNAMIC ANALYSIS OF MECHANICAL NETWORKS		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)		
Technical Report 26		
5. AUTHOR(S) (First name, middle initial, last name)		
MILTON A. CHACE and MICHAEL E. KORYBALSKI		
6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1970		
8a. CONTRACT OR GRANT NO.	8a. ORIGINATOR'S REPORT NUMBER(S)	
DA-49-083 OSA-3050	Technical Report 26	
b. PROJECT NO.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.		
d.		
10. DISTRIBUTION STATEMENT		
Qualified requesters may obtain copies of this report from DDC.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY	
	Advanced Research Projects Agency	
13. ABSTRACT		
<p>Computer graphics is particularly appropriate for studying the time response of two-dimensional mechanical dynamic systems. This paper discusses the interactive use of a DEC-338 graphics terminal for input and display related to a generalized FORTRAN IV applications program named DAMN--Dynamic Analysis of Mechanical Networks. The DEC 338 is remotely timeshared to the University of Michigan IBM System 360/67 central computer, operating under MTS (Michigan Terminal System).</p>		

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

UNIVERSITY OF MICHIGAN



3 9015 02653 5768