DYNAMIC STUDY

OF A

FOUR BAR LINKAGE

WALKING MACHINE LEG

A Thesis

Presented in Partial Fulfillment of the Requirements
for the Degree Master of Science

by

T. Frank Brown Jr., B.S.M.E.

The Ohio State University

1982

Approved by:

Copyright © 1982
by T. Frank Brown Jr.
All rights reserved.
To my parents

T. Frank and Wylodine
Abstract

Walking is a well patterned behavior where the kinetic energy of the leg varies dramatically during a cycle. Most walking machines built in the past have converted large quantities of this kinetic energy to heat by braking the leg at the ends of the stride and return phases. An alternative approach for a four-bar linkage, planar, robot leg is to let the leg act as a pendulum. The mass distribution of the leg is such that the leg stores the kinetic energy at the end of the stride as gravitational potential energy which is reused to power the leg through the return phase. As part of the research reported in this thesis, an interactive computer simulation program was developed to study the effects of mass distribution on the pendulation characteristics of a four-bar linkage, planar leg. A six legged, "unpowered", walking machine was also designed and built for validation of the simulation program results. The maximum error between predicted and measured position measurements of the leg during the return phase is less than 10% of the full scale leg motion. Measurements of power input to the machine indicate a decrease in energy consumption over both the OSU Monopod and OSU Hexapod walking vehicles.
Acknowledgements

I would like to thank my advisors, Prof. G. Kinzel and Prof. K. Srinivasan for their guidance throughout the writing of this thesis. I acknowledge the support of Prof. K. Waldron who manages the walking machine project at The Ohio State University and the Defense Advanced Research Products Agency who sponsored this research. Also, I would like to thank the Department of Mechanical Engineering Machine Shop technicians for their contributions to building the DUWE. I gratefully acknowledge the aid of my friends and fellow graduate students who gave of their time during the building and testing of the DUWE. Most of all I would like to thank my family and especially my wife, Betsy, for their support, patience, and understanding during my college studies.
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Page</td>
<td>i</td>
</tr>
<tr>
<td>Dedication</td>
<td>ii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iv</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>Chapter 1: Overview of Walking Vehicles</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 DUWE Problem Statement</td>
<td>11</td>
</tr>
<tr>
<td>1.3 Review of Past Work</td>
<td>12</td>
</tr>
<tr>
<td>1.4 Contents of Thesis</td>
<td>12</td>
</tr>
<tr>
<td>Chapter 2: Equation Development for</td>
<td>14</td>
</tr>
<tr>
<td>Computer Simulation</td>
<td>14</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Kinematic Equations</td>
<td>15</td>
</tr>
<tr>
<td>2.3 Dynamic Equations</td>
<td>22</td>
</tr>
<tr>
<td>2.4 Miscellaneous Equations</td>
<td>30</td>
</tr>
<tr>
<td>2.4-1 Force Equation Transformation</td>
<td>30</td>
</tr>
<tr>
<td>2.4-2 Torque Input Equations</td>
<td>31</td>
</tr>
<tr>
<td>2.4-3 Reverse Kinematic Equations</td>
<td>34</td>
</tr>
<tr>
<td>2.4-4 Body Center of Mass</td>
<td>37</td>
</tr>
<tr>
<td>Chapter 3: Computer Simulation</td>
<td>40</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>40</td>
</tr>
</tbody>
</table>


3.2 Capabilities and Limitations 41
3.3 Program Description 44
   3.3-1 Run Stored Version 44
   3.3-2 Change Leg Geometry 47
   3.3-3 Change leg Parameters 47
   3.3-4 Change Bearing Parameters 49
   3.3-5 Change Stride Parameters 52
   3.3-6 Change Frame Parameters 55
   3.3-7 Run New Parameters 55
   3.3-8 Create Data File 59
   3.3-9 Exit 61
   3.3-10 Miscellaneous Options

Chapter 4: Mechanical Design of DUWE ............... 63
   4.1 Introduction 63
   4.2 Preliminary Design 66
   4.3 Adjustable Link Design 67
   4.4 Fixed Link Design 74
   4.5 Coupler Link Design 76
   4.6 Bearing Selection 79
   4.7 Frame Design 79
   4.8 Coupling and Latching Mechanism 82
   4.9 Towing Cart 83
   4.10 Force Sensor 85
   4.11 System Improvements 85

Chapter 5: Electrical Design of the DUWE .............. 92
   5.1 Introduction 92
   5.2 Microprocessor Controller 92
   5.3 Position Sensors 93
   5.4 Leg and Brake Actuators 95
List of Tables

Table | Description | Page
--- | --- | ---
2.1 | List of Variables for Equation Development | 24
4.1 | Design Values of Leg Geometry | 66
4.2 | Design Changes for Improved Foot Trace | 89
6.1 | States of the Solenoid Actuators | 100
7.1 | Measured Values of Mass Parameters | 109
7.2 | Measured Values of Bearing Parameters | 110
7.3 | Validation Results | 113
8.1 | Power Audit of DUWE | 128
8.2 | Power Audit of DUWE with Improved Actuators | 128
8.3 | Specific Resistance of DUWE | 132
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Schematic of Monopod Leg Geometry</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>Picture of Monopod Vehicle System</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>Picture of Hexapod Vehicle System</td>
<td>7</td>
</tr>
<tr>
<td>1.4</td>
<td>Pendulation of a Four-Bar Linkage Leg</td>
<td>9</td>
</tr>
<tr>
<td>1.5</td>
<td>Actuator Power vs. Time for Monopod Operating at 5.6 in./sec.</td>
<td>10</td>
</tr>
<tr>
<td>2.1</td>
<td>Schematic of DUWE Leg Geometry</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Vector Arrangement for Loop Equations</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>Free Body Diagram of Linkage Links</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Force Directions for Transformation</td>
<td>32</td>
</tr>
<tr>
<td>2.5</td>
<td>Element Centers of Mass Locations for DUWE Vehicle</td>
<td>38</td>
</tr>
<tr>
<td>3.1</td>
<td>Computer Generated Schematic of DUWE with Support Triangle</td>
<td>42</td>
</tr>
<tr>
<td>3.2</td>
<td>Initial Screen Page of Computer Simulation</td>
<td>45</td>
</tr>
<tr>
<td>3.3</td>
<td>Screen Page for Run Stored Version Option</td>
<td>46</td>
</tr>
<tr>
<td>3.4</td>
<td>Screen Page for Change Leg Geometry Option</td>
<td>48</td>
</tr>
<tr>
<td>3.5</td>
<td>Screen Page for Change Leg Parameter Option</td>
<td>50</td>
</tr>
<tr>
<td>3.6</td>
<td>Screen Page for Cursor Option in Locating the Center of Mass</td>
<td>51</td>
</tr>
<tr>
<td>3.7</td>
<td>Screen Page for Change Bearing Parameters Option</td>
<td>53</td>
</tr>
<tr>
<td>3.8</td>
<td>Screen Page for Change Stride Parameters Option</td>
<td>54</td>
</tr>
<tr>
<td>3.9</td>
<td>Screen Page for Change Frame Parameters Option</td>
<td>56</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.10</td>
<td>Screen Page for Run New Parameters Option</td>
<td>57</td>
</tr>
<tr>
<td>3.11</td>
<td>Screen Page for Create Data File Option</td>
<td>60</td>
</tr>
<tr>
<td>4.1</td>
<td>Picture of DUWE with Towing Cart</td>
<td>64</td>
</tr>
<tr>
<td>4.2</td>
<td>Cardboard Model of DUWE Leg</td>
<td>68</td>
</tr>
<tr>
<td>4.3</td>
<td>Assembly Drawing of DUWE</td>
<td>69</td>
</tr>
<tr>
<td>4.4</td>
<td>Schematic of Adjustable Link Toggle Mechanism</td>
<td>70</td>
</tr>
<tr>
<td>4.5</td>
<td>Picture of Adjustable Link Components</td>
<td>72</td>
</tr>
<tr>
<td>4.6</td>
<td>Drawing of Toggle Mechanism's Main Slide Plate</td>
<td>73</td>
</tr>
<tr>
<td>4.7</td>
<td>Drawing of Toggle Mechanism's Solenoid Slide</td>
<td>73</td>
</tr>
<tr>
<td>4.8</td>
<td>Picture of Adjustable Link Assembly</td>
<td>75</td>
</tr>
<tr>
<td>4.9</td>
<td>Picture of Fixed Link Assembly</td>
<td>77</td>
</tr>
<tr>
<td>4.10</td>
<td>Picture of Coupler Link Assembly</td>
<td>78</td>
</tr>
<tr>
<td>4.11</td>
<td>Bearing Duplex Arrangements</td>
<td>80</td>
</tr>
<tr>
<td>4.12</td>
<td>Picture of DUWE Leg Assembly</td>
<td>81</td>
</tr>
<tr>
<td>4.13</td>
<td>Picture of Tripod Latching Mechanism</td>
<td>83</td>
</tr>
<tr>
<td>4.14</td>
<td>Picture of Load Ring Force Transducer</td>
<td>86</td>
</tr>
<tr>
<td>4.15</td>
<td>Foot-trace of Original Monopod Leg Design</td>
<td>88</td>
</tr>
<tr>
<td>4.16</td>
<td>Foot-trace of Improved DUWE Leg Design</td>
<td>90</td>
</tr>
<tr>
<td>5.1</td>
<td>Picture of Librascope Gray Code Encoder</td>
<td>94</td>
</tr>
<tr>
<td>5.2</td>
<td>Picture of Guardian T12x19 Solenoid</td>
<td>94</td>
</tr>
<tr>
<td>5.3</td>
<td>Picture of P8 Optical Isolater</td>
<td>97</td>
</tr>
<tr>
<td>5.4</td>
<td>Picture of Helipot Rotary Potentiometer</td>
<td>98</td>
</tr>
<tr>
<td>6.1</td>
<td>Control Program Flow Chart</td>
<td>102</td>
</tr>
<tr>
<td>7.1</td>
<td>Method of Locating Link Center of Mass</td>
<td>106</td>
</tr>
</tbody>
</table>
7.2 Method of Determining Link Mass Moment of Inertia .................... 108

7.3 Sample Position/Time Record from Pen, Strip-Chart Recorder .............. 111

7.4 Comparison of Predicted Leg Motion with Measured Leg Motion at 5.80 in./sec. ... 114

7.5 Comparison of Predicted Leg Motion with Measured Leg Motion at 7.87 in./sec. ... 115

7.6 Comparison of Predicted Leg Motion with Measured Leg Motion at 9.00 in./sec. ... 116

7.7 Comparison of Predicted Leg Motion with Measured Leg Motion at 9.66 in./sec. ... 117

7.8 Improved Prediction of Leg Motion after Bearing Friction Adjustment ............ 119

8.1 Force/Time Record of DUWE Towing Force ........... 122

8.2 Towing Power Input at 6.20 in./sec. .............. 124

8.3 Towing Power Input at 7.87 in./sec. .............. 125

8.4 Towing Power Input at 9.66 in./sec. .............. 126

8.5 Sample Solenoid Power Measurement from Pen, Strip-Chart Recorder ............ 127

8.6 Specific Resistance of DUWE, Monopod, and Hexapod ..................... 133

8.7 Specific Resistance of DUWE, Monopod, and Hexapod Compared with other Vehicles Studied by Gabrielli and Von Karman ............ 135
Chapter 1

Overview of Walking Vehicles

1.1 Introduction

Studies have shown that approximately 50% of the earth's terrain is inaccessible by wheeled or tracked vehicles [1]*. However man and animals have little difficulty in traversing most of these areas. Land locomotion in these mountainous, swampy, or overgrown terrains requires a high degree of terrain adaptability which is apparent in walking systems. There is increasing interest in the development of machines with articulated limbs to simulate walking. Recent developments in computer control and new insights into the kinematics of walking have made such machines feasible.

There are certain theoretical advantages of walking over rolling. In soft terrain a wheeled or tracked vehicle leaves a continuous track of soil compaction. In contrast, walking is a discrete mode using less energy in soil compaction.** One might consider the above argument invalid

*Brackets indicate references listed in the Bibliography.
when applied to rail or highway transportation where no compaction is apparent. However the preparation cost of the surface must be included. With these considerations a machine which ambulates instead of rolling is advantageous in certain rough, unprepared terrains.

A number of machines have been constructed using one, two, four, six, or eight legs [3,4,5,6,7,8,9]. These machines demonstrated varying degrees of terrain adaptability; however, most were energy inefficient.*** This is a severe disadvantage in an autonomous vehicle, limiting its operating range between fuel stops. There are several contributing factors to this low efficiency, most stemming from poor understanding and application of the complicated kinematics of legged locomotion [10]. The previous machines were designed as test beds for control algorithms with little emphasis placed on mechanical efficiency. However, before the inherent energy savings of walking can be realized, the mechanical efficiency of these machines must be greatly improved.

Legged locomotion is highly cyclic in nature. This oscillatory motion causes radical fluctuation in the kinetic

**Such studies were completed by Becker in 1969 [2].

*** Energy efficiency is used as a relative term. Work output is required to overcome environmental resistance, but this is a small percentage of the losses within these machines.
energy of the system. This is not apparent in conventional wheeled vehicles where the kinetic energy is relatively constant. Leg motion requires energy input to accelerate the leg at the beginning of the cycle and energy removal to decelerate the leg at the end of the cycle. Converting the kinetic energy to heat by braking the leg at the end of the cycle is costly, accounting for approximately 22% of the OSU Hexapod's power consumption [11]. For efficient operation, this energy must be reused in other parts of the system or stored for later use. This operation is displayed in most mammalian legs which store the kinetic energy at the end of the support cycle as gravitational potential energy [12]. The leg then acts as a pendulum through the return cycle. The design of an efficient robot leg must be compatible with these principles.

The mechanical design of an energy efficient robot leg was completed by Vohnout [3,13]. The relationships of leg geometry and actuator placement to energy consumption were studied. The leg, shown schematically in Figure 1.1 is a simple four-bar linkage mechanism. Rotation of the drive crank causes the foot point to travel on an approximate straight line with respect to the frame. A second degree of freedom allows variation in working height. This is required for the return phase and also for obstacle avoidance. This variation is provided by the adjustable
Figure 1.1 Schematic of Monopod Vehicle
driven link of the linkage. The third degree of freedom, required for turning, is configured by mounting the linkage supports on an axis in the plane of the linkage and parallel to the horizon. This allows the leg to abduct-adduct much like the human leg. Only one leg of this type was constructed. It is mounted in a test cart as shown in Figure 1.2, and is referred to as the Monopod. The machine has operated effectively under computer control.

Normal straight level ground walking requires a straight foot path during the support phase. The leg geometry of the Monopod takes advantage of this pattern. Only actuation of the drive crank is required to approximate this desired foot motion. The adjustable crank is used only at the beginning and ends of the stride for clearance during the return phase. This is in contrast to the joint powered, dyad leg geometry of the OSU Hexapod [6], shown in Figure 1.3, which actuates all three joint actuators for straight level walking. A comparison of the two machines indicates an improvement in energy consumption for the Monopod by a factor of 25 over the Hexapod due to this fact.

Another important characteristic of the Monopod leg geometry is that it adapts readily to the energy storage and exchange principles discussed earlier. The four bar linkage arrangement is actually a double pendulum with the drive crank as one pendulum and the driven crank as the other
Figure 1.2 Picture of Monopod Vehicle System
Figure 1.3 Picture of Hexapod Vehicle System
pendulum. This is illustrated in Figure 1.4. Proper mass distribution in the links allows the linkage to store kinetic energy as potential energy at the end of the stride and then pendulate with no power input back to the beginning of the next stride. At the beginning of the stride the leg free falls to the toedown position with no energy spent in accelerating the leg and powering it through the return cycle. This mode of control is not exploited on the Monopod. The result is a significant power drain when accelerating the leg. This is easily seen in the time history trace of the power consumption data shown in Figure 1.5 where Mode 1 of the figure indicates the return phase, Mode 2 indicates the toe-down phase, Mode 3 indicates the support phase, and Mode 4 indicates the toe-off phase.

Effective use of the energy exchange control mode requires approximate balance between the kinetic energy developed by the leg and the potential energy storage capacity of the leg. This idea has not been applied to past machines. Actuator designs have used electric motors with large gear reductions. On the Hexapod this reduction is 4263:1. This creates large reflected inertias in the drive train with excessive kinetic energy in the motor armature. The Monopod has a smaller but similar problem with its 101.63:1 gear reduction. A more practical
Figure 1.4 Pendulation of a Four-Bar Linkage Leg
actuator must have little or no gear reduction [10]. This keeps the kinetic energy values at manageable levels and allows for more efficient conversion to potential energy.

1.2 DUWE Problem Statement

The goal of this research is to demonstrate the kinematic and dynamic mechanism principles required for efficient kinetic/potential energy transfer in a vehicle with legs. The improved energy efficiency of a pendulating leg based on the above principles is also demonstrated. Leg geometry is restricted to the planar, four bar linkage type leg of the Monopod. This geometry has already displayed significant improvements in energy consumption. For purposes of energy comparison the Hexapod and Monopod systems will be used.

This project can be divided into four subtasks.

1. Development of an interactive computer simulation program to model the motion of a four bar linkage leg during a pendulation return.

2. Design and construction of a six legged machine to demonstrate the above principles.

3. Testing of the machine under various conditions to validate the computer simulation.

4. Comparison of energy consumption with the Monopod and Hexapod vehicles.
For reasons discussed in Chapter 4 the entire project was entitled "Dynamic Unpowered Walking Experiment". The actual machine is named DUWE.

1.3 Review of Past Work

The mechanical design of walking machines has been of secondary concern to those interested with the control aspects. The Monopod vehicle is the main exception as discussed above.

Computer simulations of robotic mechanisms have been of great interest to control experts [14,15]. Cheng [16] developed a control simulation for the linkage type leg of the Monopod. He predicted actuator torque and motion for a designated foot trajectory, but his work did not readily adapt to the principles covered in this thesis. For this reason, the author's development is presented. However, the contributions of others are acknowledged in the main body of the thesis.

1.4 Contents of Thesis

Chapter 2 covers the equation development for the simulation program. Chapter 3 gives a description of the computer simulation. The mechanical design and electrical design of the DUWE are discussed in Chapters 4 and 5,
respectively. Chapter 6 briefly describes the control algorithms used to operate the DUWE. The computer simulation validation is explained in Chapter 7. Finally, energy consumption comparisons are covered in Chapter 8. The remaining chapter describes conclusions and suggested extensions of the research effort.
Chapter 2
Equation Development for Computer Simulation

2.1 Introduction

In this chapter the fundamental kinematic and dynamic equations are derived for the DUWE leg geometry. A simplified two degree of freedom planar linkage is modeled. This is sufficient for straight line walking. The added complexity of modeling the third degree of freedom did not warrant consideration in the initial scope of the project. The geometry and operation of the leg are identical to the Monopod leg.

Two major assumptions are made. The first is that shortening the driven link to raise the foot from the ground has negligible effect on the dynamics of the return phase. This assumption is accurate when the shortening is small and the time is short. Such operation is expected when cruise mode walking is employed. The second assumption is that the control of the leg during the support phase is pre-specified. The support phase is the phase where the leg supports some fraction of the vehicle weight. Control of the support phase was the major goal of Cheng's simulation [16]. The development presented here concerns
the dynamic prediction of the leg motion during the return phase.

Figure 2.1 shows a schematic of the four bar linkage with the parameters used to describe the geometry. Joints are labeled A, B, C, D while links are labeled 1, 2, 3, 4. Link 1 is the fixed link, link 2 the adjustable link, link 3 the coupler link, and link 4 the supports. With the dynamic effects negligible, the adjustable link can be viewed as having two states, lengthened for support phase or shortened for the return phase. The general linkage can then be represented as a single degree of freedom mechanism with the state of the adjustable link fixed. The state variable used to describe the linkage state is the fixed link angular position, theta, $\theta$.

2.2 Kinematic Equations

The four bar linkage position is described as a function of the fixed link angular position. Velocity equations are determined by differentiating the position equations with respect to time. Acceleration equations are obtained by differentiating the velocity equations with respect to time. The equations for the foot position, velocity, and acceleration are then derived.
Figure 2.1 Schematic of DUWE Leg Geometry

1 FIXED LINK
2 ADJUSTABLE LINK
3 COUPLER LINK
4 BASE "SUPPORT" LINK
A JOINT "A"
B JOINT "B"
C JOINT "C"
D JOINT "D"
A loop equation technique [17] is used to define the linkage position. Figure 2.2 shows the vector arrangement and describes the variables used. Equation (2.2-1) and (2.2-2) are derived by summing the x and y components of these vectors.

\[ L_3 \cos(\phi) = L_1 \cos(\theta) - L_2 \cos(\gamma) + L_x \]  \hspace{1cm} (2.2-1)

\[ L_3 \sin(\phi) = L_1 \sin(\theta) - L_2 \sin(\gamma) + L_y \]  \hspace{1cm} (2.2-2)

Squaring and adding these equations to eliminate \( \phi \) yields:

\[
L_3^2 = L_x^2 + L_y^2 + L_1^2 + L_2^2 + 2L_x \{L_x \cos(\theta) + L_y \sin(\theta)\} + 2L_2L_1 \{\cos(\theta) \cos(\gamma) + \sin(\theta) \sin(\gamma)\} \]  \hspace{1cm} (2.2-3)

Substituting the trigometric identities:

\[
\cos(\gamma) = \frac{1 - \tan^2(\gamma/2)}{1 + \tan^2(\gamma/2)} \]  \hspace{1cm} (2.2-4)

\[
\sin(\gamma) = \frac{2 \tan(\gamma/2)}{1 + \tan^2(\gamma/2)} \]  \hspace{1cm} (2.2-5)
Figure 2.2 Vector Arrangement for Loop Equations
into Equation (2.2-3) and multiplying by \( \{ 1 + \tan(\gamma/2)\} \) yields after rearranging:

\[
A \tan^2(\gamma/2) + B \tan(\gamma/2) + C = 0 \tag{2.2-6}
\]

where:

\[
A = L_3^2 - L_x^2 - L_y^2 - L_1^2 - L_2^2 - 2 L_1 \{L_x \cos(\theta) + L_y \sin(\theta)\} - 2 L_2 L_x - 2 L_1 L_2 \cos(\theta) \tag{2.2-7}
\]

\[
B = 4 L_2 L_y + 4 L_1 L_2 \sin(\theta) \tag{2.2-8}
\]

\[
C = A + 4 L_2 L_x + 4 L_1 L_2 \cos(\theta) \tag{2.2-9}
\]

Gamma, \( \gamma \) is determined to be:

\[
\gamma = 2 \tan^{-1} \left( \frac{-B + \sqrt{B^2 - 4 AC}}{2A} \right) \tag{2.2-10}
\]

for:

\( \theta \) deg. < \( \gamma \) < 180 deg.

Equations (2.2-1) and (2.2-2) can be used to directly solve
for \( \phi \) once \( \gamma \) is known. Using the trigonometric identity:

\[
\tan(\frac{\phi}{2}) = \frac{\sin(\phi)}{1 - \cos(\phi)}
\]  

(2.2-11)

gives:

\[
\phi = 2 \tan \left( \frac{L_y + L_1 \sin(\theta) - L_2 \sin(\gamma)}{L_3 + L_x + L_1 \cos(\theta) - L_2 \cos(\gamma)} \right)
\]  

(2.2-12)

This completes the position analysis of the linkage.

The velocity equations are derived by differentiating the position Equations (2.2-1) and (2.2-2). This yields:

\[
-L_1 \dot{\theta} \sin(\theta) = -L_2 \dot{\gamma} \sin(\gamma) - L_3 \dot{\phi} \sin(\phi)
\]  

(2.2-13)

\[
L_1 \dot{\theta} \cos(\theta) = L_2 \dot{\gamma} \cos(\gamma) + L_3 \dot{\phi} \cos(\phi)
\]  

(2.2-14)

Solving for \( \gamma \) and \( \phi \) in terms of \( \theta \) gives:

\[
\dot{\gamma} = \frac{\dot{\theta} L_1 \sin(\theta) L_3 \cos(\phi) - \dot{\theta} L_1 \cos(\theta) L_3 \sin(\phi)}{D}
\]  

(2.2-15)

\[
\dot{\phi} = \frac{\dot{\theta} L_1 \cos(\theta) L_2 \sin(\gamma) - \dot{\theta} L_1 \sin(\theta) L_2 \cos(\gamma)}{D}
\]  

(2.2-16)
where:

\[ D = L_2 \sin(\gamma) L_3 \cos(\phi) - L_2 \cos(\gamma) L_3 \sin(\phi) \]  \hspace{1cm} (2.2-17)

The acceleration equations are derived in a similar fashion by differentiating Equations (2.2-13) and (2.2-14). The results are:

\[ \ddot{\gamma} = \frac{E L_3 \cos(\phi) - F L_3 \sin(\phi)}{G} \]  \hspace{1cm} (2.2-18)

\[ \ddot{\phi} = \frac{E L_2 \cos(\gamma) - F L_2 \sin(\gamma)}{G} \]  \hspace{1cm} (2.2-19)

where:

\[ E = \dot{\theta} L_1 \sin(\theta) + \dot{\theta}^2 L_1 \cos(\theta) - \dot{\gamma}^2 L_2 \cos(\gamma) - \dot{\phi}^2 L_3 \cos(\phi) \]  \hspace{1cm} (2.2-20)

\[ F = \ddot{\theta} L_1 \cos(\theta) - \dot{\theta}^2 L_1 \sin(\theta) - \dot{\gamma}^2 L_2 \sin(\gamma) + \dot{\phi}^2 L_3 \sin(\phi) \]  \hspace{1cm} (2.2-21)

\[ G = L_2 \sin(\gamma) L_3 \cos(\phi) - L_2 \cos(\gamma) L_3 \sin(\phi) \]  \hspace{1cm} (2.2-22)
The foot position, velocity, and acceleration are described in terms of the angular position, velocity and acceleration of the links. This gives:

\[
E_x = L_2 \cos(y) + L_5 \cos(\phi + \phi_L) - L_x \quad (2.2-23)
\]

\[
\dot{E}_x = -\dot{y} L_2 \sin(y) - \dot{\phi} L_5 \sin(\phi + \phi_L) \quad (2.2-24)
\]

\[
\ddot{E}_x = -\ddot{y} L_2 \sin(y) - L_2 \ddot{y}^2 \cos(y) - \phi \ddot{L}_5 \sin(\phi + \phi_L) - L_5 \phi^2 \cos(\phi + \phi_L) \quad (2.2-25)
\]

\(\phi, \gamma, \dot{\phi}, \dot{y}, \ddot{\phi}, \ddot{y}\) can be represented in terms of \(\theta, \dot{\theta}, \ddot{\theta}\), as prescribed in the above equations.

2.3 Dynamic Equations

Two methods are generally used to describe the dynamic behavior of a rigid body. The first approach is based on Newton-Euler equations with constraints [14]. This method relies on representing the linkage as a number of "free-bodies" and formulating the equations of motion using Newton's second law:

\[
\text{force} = \text{mass} \times \text{acceleration}.
\]

The second approach is a Lagrangian method which involves computing the energy of the system [18]. Since the total energy is constant, the motion characteristics of a body can
be computed by comparing the potential and kinetic energy of the system as a function of time. Cheng [16] discussed the relative merits of the two approaches as applied to the four-bar linkage geometry. He chose the Newton-Euler approach because the equation development and solution are simpler. This approach is also adopted here.

The first consideration is to represent the linkage as a set of "free-bodies" and identify each. This is done in the expanded view of Figure 2.3. Variables used to describe the linkage and the forces acting upon it are listed in Table 2.1.

The dynamic equations are derived by equating forces in the x-direction, equating forces in the y-direction, and summing moments about the center of mass of each "free-body". For link 1, summing forces in the x-direction gives:

\[
F_{ax} - F_{bx} + m_1 g \sin(\theta) - m_1 A_x + m_1 L_{ml} \dot{\theta}^2 \cos(\theta + \theta_m) + m_1 L_{ml} \ddot{\theta} \sin(\theta + \theta_m) = 0
\]

(2.3-1)

where:

\[
m_1 L_{ml} \dot{\theta}^2 \cos(\theta + \theta_m) \\
m_1 L_{ml} \ddot{\theta} \sin(\theta + \theta_m) \\
m_1 A_x
\]
TABLE 2.1 List of Variables for Equation Development

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>Length of fixed link</td>
<td>(in.)</td>
</tr>
<tr>
<td>$L_2$</td>
<td>Length of adjustable link</td>
<td>(in.)</td>
</tr>
<tr>
<td>$L_3$</td>
<td>Length of coupler link</td>
<td>(in.)</td>
</tr>
<tr>
<td>$L_5$</td>
<td>Length of shank</td>
<td>(in.)</td>
</tr>
<tr>
<td>$L_x$</td>
<td>$x$-distance between link supports</td>
<td>(in.)</td>
</tr>
<tr>
<td>$L_y$</td>
<td>$y$-distance between link supports</td>
<td>(in.)</td>
</tr>
</tbody>
</table>
| $\phi_L$ | Angle measurement between  
            | coupler link and shank                                                      | (deg.)   |
| $\theta$ | Angular rotation of the fixed link with  
                       | respect to the frame                                                       | (deg.)   |
| $\gamma$ | Angular rotation of the adjustable link with  
                     | respect to the frame                                                       | (deg.)   |
| $\phi$ | Angular rotation of the coupler link with  
<pre><code>                 | respect to the frame                                                       | (deg.)   |
</code></pre>
<p>| $m_1$  | Mass of the fixed link                                                      | (lbf. sec.@/in.) |
| $m_2$  | Mass of the adjustable link                                                 | (lbf. sec.@/in.) |
| $m_3$  | Mass of the coupler link                                                    | (lbf. sec.@/in.) |
| $J_{cg1}$ | Mass moment of inertia about the center of mass of the fixed link         | (lbf. in. sec.@) |
| $J_{cg2}$ | Mass moment of inertia about the center of mass of the adjustable link    | (lbf. in. sec.@) |
| $J_{cg3}$ | Mass moment of inertia about the center of mass of the coupler link        | (lbf. in. sec.@) |
| $L_{m1}$ | Distance from fixed link support to the center of mass of the fixed link   | (in.)    |
| $L_{m2}$ | Distance from adjustable link support to the center of mass of the adjustable link | (in.) |
| $L_{m3}$ | Distance from coupler joint on adjustable link to the center of mass of the coupler link | (in.) |</p>
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_m$</td>
<td>Angle measurement locating the center of mass of the fixed link</td>
<td>(deg.)</td>
</tr>
<tr>
<td>$\gamma_m$</td>
<td>Angle measurement locating the center of mass of the adjustable link</td>
<td>(deg.)</td>
</tr>
<tr>
<td>$\phi_m$</td>
<td>Angle measurement locating the center of mass of the coupler link</td>
<td>(deg.)</td>
</tr>
<tr>
<td>$F_{ax}$</td>
<td>Force in the x-direction on Joint A</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$F_{ay}$</td>
<td>Force in the y-direction on Joint A</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$F_{bx}$</td>
<td>Force in the x-direction on Joint B</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$F_{by}$</td>
<td>Force in the y-direction on Joint B</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$F_{cx}$</td>
<td>Force in the x-direction on Joint C</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$F_{cy}$</td>
<td>Force in the y-direction on Joint C</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$F_{dx}$</td>
<td>Force in the x-direction on Joint D</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$F_{dy}$</td>
<td>Force in the y-direction on Joint D</td>
<td>(lbf.)</td>
</tr>
<tr>
<td>$T_a$</td>
<td>Friction torque applied at Joint A</td>
<td>(in. lbf.)</td>
</tr>
<tr>
<td>$T_b$</td>
<td>Friction torque applied at Joint B</td>
<td>(in. lbf.)</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Friction torque applied at Joint C</td>
<td>(in. lbf.)</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Friction torque applied at Joint D</td>
<td>(in. lbf.)</td>
</tr>
<tr>
<td>$T_{aa}$</td>
<td>Input torque applied at Joint A</td>
<td>(in. lbf.)</td>
</tr>
<tr>
<td>$T_{dd}$</td>
<td>Input torque applied at Joint D</td>
<td>(in. lbf.)</td>
</tr>
<tr>
<td>$g$</td>
<td>Acceleration of gravity</td>
<td>(in./sec.$^2$)</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Angle frame of vehicle makes with respect to the gravity vector</td>
<td>(deg.)</td>
</tr>
<tr>
<td>$A_x$</td>
<td>Acceleration of frame in x-direction</td>
<td>(in./sec.$^2$)</td>
</tr>
<tr>
<td>$A_y$</td>
<td>Acceleration of frame in y-direction</td>
<td>(in./sec.$^2$)</td>
</tr>
</tbody>
</table>
Figure 2.3 Free Body Diagram of Linkage Links
represent inertial loading (mass x acceleration).

Summing forces in the y-direction yields:

\[
F_{ay} - F_{by} - m_1 g \cos(\theta) - m_1 A_y - m_1 L_{ml} \dot{\theta}^2 \sin(\theta + \theta_m) \\
+ m_1 L_{ml} \ddot{\theta} \cos(\theta + \theta_m) = 0
\]  
(2.3-2)

And finally summing moments about the center of mass of the link gives:

\[
- T_{aa} - T_a + F_{ax} L_{ml} \sin(\theta + \theta_m) + F_{ay} L_{ml} \cos(\theta + \theta_m) \\
+ F_{bx} \{L_1 \sin(\theta) - L_{ml} \sin(\theta + \theta_m)\} \\
+ F_{by} \{L_1 \cos(\theta) - L_{ml} \cos(\theta + \theta_m) - J_{c_{gl}} \ddot{\theta} = 0
\]  
(2.3-3)

where:  \( J_{c_{gl}} \ddot{\theta} \) represents the inertial moment.

Similar equations can be derived for links 2 and 3.

For link 2:

\[
F_{dx} - F_{cx} + m_2 g \sin(\gamma) - m_2 A_x - m_2 L_{m2} \dot{\gamma}^2 \cos(\gamma + \gamma_m) \\
+ m_2 L_{m2} \ddot{\gamma} \sin(\gamma + \gamma_m) = 0
\]  
(2.3-4)

\[
F_{dy} - F_{cy} - m_2 g \cos(\gamma) - m_2 A_y - m_2 L_{m2} \dot{\gamma}^2 \sin(\gamma + \gamma_m) \\
+ m_2 L_{m2} \ddot{\gamma} \cos(\gamma + \gamma_m) = 0
\]  
(2.3-5)
\(- T_c - T_d + F_{dx} L_{m2} \sin(\gamma + \gamma_m) + F_{dy} L_{m2} \cos(\gamma + \gamma_m) \\
+ F_{cx} \{L_2 \sin(\gamma) - L_{m2} \sin(\gamma + \gamma_m)\} \\
+ F_{cy} \{L_2 \cos(\gamma) - L_{m2} \cos(\gamma + \gamma_m)\} = J_{cg2} \ddot{\gamma} \)  
\(2.3-6\)

And for link 3:

\(F_{cx} + F_{bx} + m_3 g \sin(\phi) - m_3 A_x + m_3 L_{m3} \dot{\phi}^2 \cos(\phi + \phi_m) \\
+ m_3 L_2 \ddot{\gamma} \cos(\gamma) + m_3 L_{m3} \ddot{\phi} \sin(\phi + \phi_m) \\
+ m_3 L_2 \dot{\gamma} \sin(\gamma) + F_{gx} = 0 \)  
\(2.3-7\)

\(F_{cy} + F_{by} - m_3 g \cos(\phi) - m_3 A_y - m_3 L_{m3} \dot{\phi}^2 \cos(\phi + \phi_m) \\
- m_3 L_2 \ddot{\gamma} \sin(\gamma) + m_3 L_{m3} \ddot{\phi} \cos(\phi + \phi_m) \\
+ m_3 L_2 \dot{\gamma} \cos(\gamma) + F_{gy} = 0 \)  
\(2.3-8\)

\(T_c + T_b + F_{cy} L_{m3} \cos(\phi + \phi_m) + F_{cx} L_{m3} \sin(\phi + \phi_m) \\
+ F_{gy} \{L_{m3} \cos(\phi + \phi_m) - L_5 \cos(\phi + \phi_L)\} \\
- F_{gx} \{-L_{m3} \sin(\phi + \phi_m) + L_5 \sin(\phi + \phi_L)\} \\
- F_{by} \{L_3 \cos(\phi) - L_{m3} \cos(\phi + \phi_m)\} \\
+ F_{bx} \{-L_3 \sin(\phi) + L_{m3} \sin(\phi + \phi_m)\} = J_{cg3} \ddot{\phi} = 0 \)  
\(2.3-9\)

This gives a total of 9 equations with \(F_{ax}, F_{ay}, F_{bx}, F_{by}, F_{cx}, F_{cy}, F_{dx}, F_{dy}, A_x, A_y, T_{aa}, T_{dd}, T_a, T_b, T_c, T_d, F_{gx}, F_{gy}, \theta, \gamma, \phi, \dot{\theta}, \dot{\gamma}, \dot{\phi}, \ddot{\phi}, \ddot{\gamma}, \dot{\theta}, \dot{\gamma}, \dot{\phi}, \theta, \gamma, \phi\) as unknowns.
Certain variables can be eliminated when limiting the study to the return phase of motion. \( F_{gx} \) and \( F_{gy} \) are zero corresponding to no foot force, and \( A_x \) and \( A_y \) can be expressed as functions of the gross machine motion. \( \gamma, \phi, \dot{\gamma}, \dot{\phi} \) can be represented in terms of \( \theta, \dot{\theta}, \ddot{\theta} \). Finally the six torque terms; \( T_{aa}, T_{dd}, T_{a'}, T_{b'}, T_{c'}, T_{d'} \); are either inputs or functions of the remaining variables. This is covered in section 2.4-2. The resulting 11 unknowns are the joint forces, \( F_{ax}, F_{ay}, F_{bx}, F_{by}, F_{cx}, F_{cy}, F_{dx}, F_{dy} \); and the fixed crank angular position, velocity, and acceleration, \( \theta, \dot{\theta}, \ddot{\theta} \), respectively.

The differential equations relating \( \theta, \dot{\theta}, \ddot{\theta} \) must be solved. These equations can be written as follows:

\[
\frac{d\theta}{dt} = \dot{\theta} \quad \quad (2.3-10)
\]

\[
\frac{d\dot{\theta}}{dt} = \ddot{\theta} \quad \quad (2.3-11)
\]

where \( \ddot{\theta} \) is a function of the remaining 10 unknowns.

A numerical solution technique employing 4th order Runge-Kutta methods [19] is implemented. This technique is widely used because it is one of the most accurate methods per computational effort. Initial conditions on \( \theta \) and \( \dot{\theta} \) are picked at the toepoff position. At this instant the leg
motion is defined by the support leg motion.

The solution can now proceed by substituting $\theta$ and $\dot{\theta}$ into Equations (2.3-1) thru (2.3-9). Since there are now only 9 unknowns, it is a simple algebraic manipulation to solve the equations. This is done numerically using Gaussian Elimination with back-substitution [19]. Once $\ddot{\theta}$ and the joint forces are solved, the solution to the differential equation can be advanced in time. This yields the next $\theta$ and $\dot{\theta}$. The algorithm is then repeated to compute the complete leg motion during the return phase.

2.4 Miscellaneous Equations

It is necessary to derive certain other equations in order to completely describe the computer simulation.

2.4-1 Force Equation Transformation

Normal and tangential force measurements on the adjustable and fixed links are helpful to a designer. Also approximate bearing friction can be modeled as a function of the normal force on the joints (see Section 2.4-2). It is necessary to convert the $x$ and $y$ force components to normal and tangential components. The simple transformations are listed below.

$$F_{ax} = F_{an} \cos(\theta) + F_{at} \sin(\theta)$$  \hspace{1cm} (2.4-1)
\[ F_{ay} = -F_{an} \sin(\theta) + F_{at} \cos(\theta) \quad (2.4-2) \]
\[ F_{bx} = F_{bn} \cos(\theta) + F_{bt} \sin(\theta) \quad (2.4-3) \]
\[ F_{by} = -F_{bn} \sin(\theta) + F_{bt} \cos(\theta) \quad (2.4-4) \]
\[ F_{cx} = F_{cn} \cos(y) + F_{ct} \sin(y) \quad (2.4-5) \]
\[ F_{cy} = -F_{cn} \sin(y) + F_{ct} \cos(y) \quad (2.4-6) \]
\[ F_{dx} = F_{dn} \cos(y) + F_{dt} \sin(y) \quad (2.4-7) \]
\[ F_{dy} = -F_{dn} \sin(y) + F_{dt} \cos(y) \quad (2.4-8) \]

The corresponding positive directions are illustrated in Figure 2.4.

2.4-2 Torque Input Equations

There are two types of torque inputs considered in this development. Friction torques are modeled at all joints and applied torques are input at the support ends of the fixed and adjustable links.

The initial method of modeling joint friction used only the normal component of the bearing reaction force on the fixed and adjustable links. This model assumes that the normal force is much larger than the tangential force acting on the joint. This assumption allows friction torques to be
Figure 2.4 Force Directions for Transformation
modeled as linear functions of the joint forces.
Equation (2.4-9) indicates the mathematical model.

\[ T = \mu R F_n \]  \hspace{1cm} (2.4-9)

where:

- \( T \) ..........Friction torque at joint
- \( \mu \) ..........Coefficient of friction of bearing
- \( R \) ..........Effective joint radius at which friction acts
- \( F_n \) ..........Normal force acting on the joint

Appropriate program logic is included to ensure that the friction torque always opposes motion.

An improved technique is derived to consider the magnitude of the normal and tangential force on the joint. The math model becomes:

\[ T = \mu R \sqrt{F_n^2 + F_t^2} \]  \hspace{1cm} (2.4-10)

where:

- \( F_t \) ..........Tangential force acting on the joint

This model is more accurate but places a non-linear term in the system of equations. An iteration scheme is used in the
program to solve the equations. To start the procedure, an initial estimate of the torque is necessary. The system of equations is then solved with "T" as a constant. A new iteration of "T" is then calculated using the resulting forces from the initial estimate. The system of equations is then resolved. By picking the initial "T" as the "T" of the last time step, sufficient accuracy is obtained in one iteration.

Torque inputs are modeled as functions of the adjustable link and fixed link angular rotation. The fixed link torque, $T_{aa}$ and the adjustable link torque, $T_{dd}$ are entered as constants in the original equation development. Setting these equal to zero is equivalent to a free swing, gravitational pendulum return. However it is possible to model a pendulum return which is aided by applied torques. This is accomplished by computing $T_{aa}$ and $T_{dd}$ at each time increment of the solution.

2.4-3 Reverse Kinematic Problem

It is necessary to solve the kinematic equations for the linkage leg in reverse if a model of the support leg motion for an input machine characteristic is desired. This entails representing $\theta$, $\dot{\theta}$, $\ddot{\theta}$ as functions of $E_x$, $\dot{E}_x$, $\ddot{E}_x$. 
Equations (2.2-1), (2.2-10), (2.2-12), and (2.2-13) relate $E_x$ and $\theta$. In Equations (2.2-1), (2.2-10), and (2.2-12), $\theta$ appears in non-linear trigonometric terms. To solve for $\theta$, Equation (2.2-23) is transformed into Equation (2.4-11). A Secant root finding method [19] is used

$$
\epsilon = -E_x + L_2 \cos(\gamma) + L_5 \cos(\phi + \phi_L) - L_x \tag{2.4-11}
$$

to find the solution, $\theta$, at which $\epsilon$ equals zero. Given the $x$-position, $E_x$, of the foot point there can be multiple solutions of the fixed crank angle; $\theta$. Constraints are imposed on theta, $\theta$, to avoid this.

Once the position analysis is solved, it is a simple task to solve the velocity analysis. The relationship between $\dot{E}_x$ and $\dot{\theta}$ is linear. Equation (2.4-12) can be derived through algebraic manipulation of Equations (2.2-15), (2.2-16), and (2.2-24).

$$
\dot{\theta} = \frac{A(-\dot{E}_x)}{B L_2 \sin(\gamma) + C L_5 \sin(\phi + \phi_L)} \tag{2.4-12}
$$

where:

$$
A = L_2 \sin(\gamma) L_3 \cos(\phi) - L_2 \cos(\gamma) L_3 \sin(\phi) \tag{2.4-13}
$$

$$
B = L_1 \sin(\theta) L_3 \cos(\phi) - L_1 \cos(\theta) L_3 \sin(\phi) \tag{2.4-13}
$$

35
\[ C = L_2 \sin(\gamma) L_1 \cos(\theta) - L_2 \cos(\gamma) L_1 \sin(\theta) \]  

(2.4-14)

The acceleration analysis can be completed once the position and velocity analysis are done. Equations (2.2-18), (2.2-19), (2.2-25) are manipulated to give:

\[
\ddot{\theta} = \frac{-E_x - \frac{D + E}{A}}{B L_2 \sin(\gamma)} - \frac{L_2 \dot{y}^2 \cos(\gamma) - L_5 \dot{\phi}^2 \cos(\phi + \phi_L)}{C L_5 \sin(\phi + \phi_L)} 
\]

(2.4-16)

where:

\[
D = L_2 \sin(\gamma) \{L_3 \cos(\phi) \left[ \dot{\theta}^2 L_1 \cos(\theta) - \dot{y}^2 L_2 \cos(\gamma) \right] \\
- \dot{\phi}^2 L_3 \cos(\phi) \} - L_3 \sin(\phi) \left[ -\dot{\theta}^2 L_1 \sin(\theta) \\
+ \dot{y}^2 L_2 \sin(\gamma) + \dot{\phi}^2 L_3 \sin(\phi) \right] 
\]

(2.4-17)

\[
E = L_5 \sin(\phi + \phi_L) \{L_2 \sin(\gamma) \left[ -\dot{\theta}^2 L_1 \sin(\theta) \\
+ \dot{y}^2 L_2 \sin(\gamma) + \dot{\phi}^2 L_3 \sin(\phi) \right] - L_2 \cos(\gamma) \left[ \dot{\theta}^2 L_1 \cos(\theta) \\
- \dot{y}^2 L_2 \cos(\gamma) - \dot{\phi}^2 L_3 \cos(\phi) \right] \}
\]

(2.4-18)
2.4-4 Body Center of Mass

Equations are developed here to compute the body center of mass. This feature is added for completeness. It is used to graphically determine the static stability of the machine in a tripod gait. A simplified approach is presented.

Figure 2.5 shows a schematic of a six legged machine with the various element centers of mass located. The machine is assumed symmetric about the y-axis. This locates the y-position of the center of mass along this axis. The x-location can be calculated as:

\[
X_t = \frac{1}{m_t} \left[ X_f m_f + X_1 m_1 + X_2 m_2 + X_3 m_3 + X_4 m_4 + X_5 m_5 + X_6 m_6 \right] 
\]  

(2.4-19)

where:

\[ X_t \] ........location of vehicle center of mass
\[ X_f \] ........location of frame center of mass
\[ X_1 \rightarrow X_6 \] ..........location of legs center of mass
\[ m_t \] ........mass of entire vehicle
\[ m_f \] ........mass of frame
\[ m_1 \rightarrow m_6 \] ..........mass of legs

The z-location is not considered although calculation would
Figure 2.5  Element Centers of Mass Locations for DUWE Vehicle
proceed as above.
Chapter 3

Computer Simulation

3.1 Introduction

This chapter describes the general outline and use of the DUWE simulation program. Emphasis is placed on how to use the program rather than the technical aspects of developing the code. A flow chart and statement listing are located in Appendix A.

The computer program is written in standard FORTRAN and is compatible with the VAX 750 mini-computer and Tektronix 4014 Graphics Terminal. The terminal has been slightly modified to permit a simplified animation capability. Animation is not done in real time but is accomplished by loading the terminal's internal memory with draw vectors for successive positions. The vectors are then rapidly flashed to the screen simulating animation. All graphics commands are implemented using Plot10 Graphics Subroutines [20].
3.2 Capabilities and Limitations

The computer program is written as an interactive design tool. Data entry is accomplished in three ways: keyboard entry, cursor entry, or file entry. The user is prompted for input. Inconsistent data entry is checked and may be corrected. Results are displayed in three ways: tabular form to the screen, tabular form to a data file, and graphical representation of the machine motion cycle. The graphical representation is a two-dimensional side view of the machine as shown in Figure 3.1. A top view of the support pattern of the legs is also given along with the machine center of mass location. The motion cycle can be displayed frame by frame or in continuous animation.

The equations for the simulation program are developed in Chapter 2. The two major assumptions are repeated below.

1. Shortening the adjustable link has negligible effects on the dynamics of the return phase.

2. The support leg motion and control are pre-specified.

In addition to these assumptions, the computer model is designed for six legged vehicles constrained to walk in a tripod gait. The tripod gait is when three legs, the front and rear legs of one side and the middle leg of the other side, are supporting the machine. This is the designated cruise mode gait for six legged vehicles.
Figure 3.1 Computer Generated Schematic of DUWE with Support Triangle
The solution technique predicts the position of one leg from toe-off, through the return phase, to toe-down. The support phase leg motion is specified by the input machine motion. This constitutes a complete analysis. Walking is artificially simulated by transforming the coordinates of the return and support legs and repeating the motion cycle. This simulation technique is fundamentally correct when constant speed walking is considered. However one must be aware of the possible errors when simulating an accelerating machine.

The simulation is very versatile allowing several different straight line walking conditions to be modeled. These conditions include the following:

1. Constant forward velocity of machine.
2. Uphill/downhill constant forward velocity at any slope.
3. Accelerating/decelerating forward motion.

Gravitational pendulum return phases or torque aided return phases can be modeled for each of the above. These conditions encompass a large majority of the conditions experienced during walking.
3.3 Program Description

The program is entitled DUWE and is accessed by RUN DUWE. The first page, shown in Figure 3.2, displays the main menu selection, a brief description of the program, and a side view schematic of the DUWE vehicle. Presently nine options are available from the menu. Menu selection is performed by aligning the terminal's cursor cross-hairs in the desired brackets and pressing the terminal space bar.

3.3-1 Run Stored Version

Selecting the Run Stored Version option allows the user to review the results of a previously analyzed case. As shown in Figure 3.3, the user is prompted for the name of the data file he wishes to review. If the file is not available, an input error is detected and an available file must be re-entered. The file format is made compatible with the program during the Create Data File mode. However, one can edit the file independent of the simulation program if the file format remains unchanged. After a successful input, a graphical display of the motion cycle can be reviewed. If not, a return to the menu is executed.
### D.U.W.E. Simulation

| 1 | Run Stored Version? |
| 2 | Change Leg Geometry? |
| 3 | Change Leg Parameters? |
| 4 | Change Bearing Parameters? |
| 5 | Change Stride Parameters? |
| 6 | Change Frame Parameters? |
| 7 | Run New Parameters? |
| 8 | Create Data File? |
| 9 | Exit? |

Align cursor in brackets for option. Hit "Space" to register.

---

**D.U.W.E. Analysis Program**

This is an interactive design program for determining the time history of a 4-bar linkage robot leg as it pendulates during the return cycle. The influence of mass distribution, leg geometry, and input torques on the return phase can be studied. The simulation models a six leg machine constrained to walk in a tripod gate. Data entry is via the keyboard, cursor, or data file. A sample case, stored in the program, can be accessed by running new parameters. Changes to this case are accomplished by the menu selection shown at left.

---

Figure 3.2 Initial Screen Page of Computer Simulation
3.3-2 Change Leg Geometry

This option is used to change the linkage geometry. It is useful when comparing machine behavior at different operating heights of the leg. It also allows different four-bar geometries to be readily modeled. Figure 3.4 shows the input format. A schematic of the latest version of the leg is drawn and scaled to the proper proportion. The numerical values for the current linkage geometry are listed and the user is prompted to input, via the keyboard, any desired changes. To avoid re-entering the current value, a "0" input instructs the program to default to the current value. After all values are entered, the page is redrawn with the updated version. The user may now selectively correct any input errors or return to the main menu to analyze the new linkage.

3.3-3 Change Leg Parameters

This selection is for changing the mass distribution of each link of the four-bar linkage. It is used to design the optimum mass distribution of the leg links for desirable dynamic operation of the leg. The location of the center of mass, the value of the mass, and the mass moment of inertia about the center of mass can be altered for each link.
DUWE

SIMULATION

DEFINE NEW LEG GEOMETRY
INPUT OF "0" DEFAULTS TO OLD VALUE.

(1) LINK LENGTH AB = 6.774
(2) LINK LENGTH CD = 8.200
(3) LINK LENGTH BC = 4.050
(4) LEG LENGTH CE = 9.900
(5) LEG ANGLE (deg) BCE = 94.000
(6) LENGTH DF = 3.170
(7) LENGTH AF = 3.500

INPUT NEW LINK LENGTH AB 6.73
INPUT NEW LINK LENGTH CD 8.25
INPUT NEW LINK LENGTH BC 4.31
INPUT NEW LEG LENGTH CE 0
INPUT NEW ANGLE (deg) BCE 9
INPUT NEW LENGTH -->(+) DF 0
INPUT NEW LENGTH <(+) AF 0

LATEST VERSION

Figure 3.4 Screen Page for Change Leg Geometry Option
Figure 3.5 illustrates the screen page for this routine.

Two modes of changing the location of the center of mass of a link are available. The first is to align the cross-hairs of the graphic cursor on the desired position of the scaled linkage shown. This is demonstrated in Figure 3.6 for the fixed and adjustable links of the four-bar leg. Although not a precise method, it is quick and adequate for the initial iterations of the design. After an approximate acceptable location is determined one can refine the position by using the keyboard mode of input.

The format for changing the value of the mass and mass moment of inertia is similar to that of changing the leg geometry. In the early preliminary design stages a legitimate estimate of these values is required. The mass moment of inertia about the center of mass of the link is dependent on the structural design as well as the location of the center of mass. This entails continued use of the program throughout the design process.

3.3-4 Change Bearing Parameters

Bearing characteristics of the joints are changed in this section. The input includes the dynamic coefficient of friction for the bearing surface and the effective radius at which the friction force acts. The input format (see
## D.U.U.E. SIMULATION

### Changing Leg Mass Parameters from Current Values

**Location**

<table>
<thead>
<tr>
<th>No. Link</th>
<th>Length (L)</th>
<th>Angle (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1582E+01</td>
<td>0.7100E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.1600E+01</td>
<td>0.5500E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.2060E+01</td>
<td>0.1150E+03</td>
</tr>
</tbody>
</table>

**Value**

<table>
<thead>
<tr>
<th>No. Link</th>
<th>Mass (JCG)</th>
<th>Value (JCG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4410E-02</td>
<td>0.5950E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.1950E-01</td>
<td>0.3770E-01</td>
</tr>
<tr>
<td>3</td>
<td>0.3650E-02</td>
<td>0.6190E-01</td>
</tr>
</tbody>
</table>

### Change Location of Mass EY, MJ in Current Version

### Change Value of Mass EY, MJ in Current Version

**Input New Value of Mass and Moment of Inertia 'JCG'**

**Input of 'b' Defaults to Old Value**

**Input Mass JCG**

<table>
<thead>
<tr>
<th>Link 1</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link 2</td>
<td>0.004</td>
<td>0.06</td>
</tr>
<tr>
<td>Link 3</td>
<td>0.004</td>
<td>0.07</td>
</tr>
</tbody>
</table>

---

Figure 3.5 Screen Page for Change Leg Parameter Option
D.U.W.E.
SIMULATION

CHANGING LEG MASS PARAMETERS FROM CURRENT VALUES

LOCATION

<table>
<thead>
<tr>
<th>NO.</th>
<th>LINK</th>
<th>LENGTH</th>
<th>ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>1</td>
<td>0.1500E+01</td>
<td>0.7100E+02</td>
</tr>
<tr>
<td>(2)</td>
<td>2</td>
<td>0.1500E+01</td>
<td>0.5500E+02</td>
</tr>
<tr>
<td>(3)</td>
<td>3</td>
<td>0.2000E+01</td>
<td>0.1150E+03</td>
</tr>
</tbody>
</table>

VALUE

<table>
<thead>
<tr>
<th>NO.</th>
<th>LINK</th>
<th>MASS</th>
<th>MMCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4)</td>
<td>1</td>
<td>0.4100E-02</td>
<td>0.5050E+01</td>
</tr>
<tr>
<td>(5)</td>
<td>2</td>
<td>0.1800E+01</td>
<td>0.9370E+01</td>
</tr>
<tr>
<td>(6)</td>
<td>3</td>
<td>0.3800E+02</td>
<td>0.6190E+01</td>
</tr>
</tbody>
</table>

CHANGE LOCATION OF MASS EV, N3P V

BY CURSOR OR KEYBOARD EV, K3P C

LOCATE MASS C.G. ON CURRENT LEG VERSION AND PRESS "SPACE"

LENGTH ANGLE

<table>
<thead>
<tr>
<th>LINK</th>
<th>LENGTH</th>
<th>ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2688E+01</td>
<td>0.1067E+03</td>
</tr>
<tr>
<td>2</td>
<td>0.3082E+01</td>
<td>0.2687E+02</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.6 Screen Page for Cursor Option in Locating the Center of Mass
Figure 3.7) is identical to that for changing the leg geometry.

3.3-5 Change Stride Parameters

Figure 3.8 shows the screen page for changing stride parameters. The input format is identical to that shown in previous sections. The five inputs are described below.

1. Stride Length - the actual length of the foot stride under study.

2. Toe-Down Position - the distance from the fixed link support to the toe-down position measured in the + x, horizontal direction.

3. Link 2 Short - the adjustment allowed in the adjustable link for returning the leg.

4. Machine Speed - the average machine speed through one step of support leg motion. The machine acceleration must be accounted for in this average speed.

5. Number Of Time Increments - Controls the time increment used in the Runge Kutta Analysis. The actual increment is also a function of average speed and length of stride of the support leg.

This routine is used to study the effects that speed and stride characteristics of the machine have on the dynamics of the returning leg. The graphical display is used to perturb the toe-down and toe-off machine heights.

52
Figure 3.8 Screen Page for Change Stride Parameters Option
3.3-6 Change Frame Parameters

This routine changes the frame characteristics of the DUWE. The parameters input include the following:

1. Leg Placement - The initial leg placement on the frame.
2. Leg Spacing - The leg spacing on the frame.
3. Frame Width - The frame width; width between foot support points.
4. Frame Length - The frame length.
5. Frame Weight - The frame weight minus the legs.

Figure 3.9 gives the screen page for inputting these parameters. This section is used to detect interference between leg sets. It is also used to study the impact of frame dimensions and leg placement on the static stability of the machines.

3.3-7 Run New Parameters

This option directs the computer to analyze the current walking condition. The support leg motion is defined to negotiate the desired machine characteristics. The swing legs are allowed to pendulate under the force of gravity and any applied input torques. The solution is graphically displayed (see Figure 3.10) as it progresses from the
**D.U.W.E. SIMULATION**

Define new frame parameters. Input of '0' defaults to old value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Leg Placement</td>
<td>12.415</td>
</tr>
<tr>
<td>(2) Leg Spacing</td>
<td>22.000</td>
</tr>
<tr>
<td>(3) Frame Width</td>
<td>36.000</td>
</tr>
<tr>
<td>(4) Frame BBLENT</td>
<td>72.000</td>
</tr>
<tr>
<td>(5) Frame Weight</td>
<td>15.000</td>
</tr>
</tbody>
</table>

Input new leg placement: 12.25
Input new leg spacing: 24
Input new frame width: 0
Input new frame length: 0
Input new frame weight: 100.0

---

Figure 3.9 Screen Page for Change Frame Parameters Option
<table>
<thead>
<tr>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN STORED VERSION?</td>
</tr>
<tr>
<td>CHANGE LEG GEOMETRY?</td>
</tr>
<tr>
<td>CHANGE LEG PARAMETERS?</td>
</tr>
<tr>
<td>CHANGE BEARING PARAMETERS?</td>
</tr>
<tr>
<td>CHANGE STRIDE PARAMETERS?</td>
</tr>
<tr>
<td>CHANGE FRAME PARAMETERS?</td>
</tr>
<tr>
<td>RUN NEW PARAMETERS?</td>
</tr>
<tr>
<td>CREATE DATA FILE?</td>
</tr>
<tr>
<td>EXIT?</td>
</tr>
</tbody>
</table>

Align cursor in brackets for option. Hit "space" to register.

**Leg Geometry Acceptable**

- Draw D.U.W.E. \[\text{EV, M}\] Y
- Animate Leg \[\text{EV, M}\] Y
initial toe-off conditions.

The program detects invalid solutions or conditions for which the machine would not operate properly. These include the following:

1. The swing phase leg fails to return to the toe-down position or returns too low.
2. The swing phase leg returns past the limits of the linkage causing the mechanism to lock-up.
3. At toe-down, the swing phase leg's velocity is too slow.

Graphical displays of these solutions are available to determine necessary corrective actions such as changing the location of the center of mass of a link to increase the speed of return.

Acceptable solutions are those for which the swinging leg returns past the toe-down position and then pendulates down to toe down with the foot velocity equal to or greater than vehicle velocity. If the leg returns too quickly, it is simply latched until the support leg has completed its stride. This latching height is pre-determined to give velocity matching at toe-down. For the speed corresponding to optimum energy efficiency, the latching is not required. The leg returns to precisely the location for correct velocity matching at toe-down. At constant speed, this is the resonant gait of the machine. It also is the upper
limit on speed for the leg conditions modeled. Since this solution corresponds to only one set of leg conditions, slower speeds warrant the use of the latching mechanism. It is possible, however, to vary the characteristics of the leg and get a loci of resonant speeds for the machine. Three apparent methods of accomplishing this are given below.

1. Move the center of mass location of the linkage
2. Vary the compliance of a spring used in a torque aided return
3. Use a varied impulse torque from the drive actuation system to aid the return.

3.3-8 Create Data File

This section of the program creates a data file compatible with the Run Stored Version option. Program efficiency is improved by saving the results of an analysis for future reference. This feature is desirable in a design situation. Figure 3.11 shows the general input format which includes the name of the data file and a case title. All parameters of the last modeled condition and a time history description of the motion cycle are saved. This file can be printed, reviewed, and edited externally to the program or it can be modified by the program using the Run Stored Version option. Sample files are located in Appendix B.
D.U.W.E.
SIMULATION

CREATING DATA FILE
ENTER OUTPUT FILE NAME: DUNE1.DAT
INPUT THE TITLE FOR THIS CASE
LESS THAN 60 CHAR.
CASE TITLE:
SAMPLE CASE FOR EXPERIMENT NO. 1
DATA IN FILE: DUNE1.DAT
HIT 'CR' TO CONTINUE

Figure 3.11 Screen Page for Create Data File Option
3.3-9 Exit

This is the last selection available on the menu. It can be selected at any time during program operation. It causes the simulation to halt execution and designates the end of a program run.

3.3-10 Miscellaneous Options

There are three remaining options, machine motion input, fixed link torque input, and adjustable link torque input. These options are accessed by linking function subroutines to the program.

The first option allows one to input desired machine motion. The average speed of the machine must be entered during the "Change Stride Parameters" option, however the actual instantaneous position, velocity, and acceleration of the machine are calculated during this option. The machine characteristics must be modeled as a function of time starting at time = 0.0 for support leg toe-down. This information is stored in "Subroutine GRLEG1". Any desired machine motion can be modeled in this fashion. Two samples of GRLEG1 are given in Appendix C. The first models constant velocity locomotion. The second routine uses a spline curve fit [19] of the fixed crank angular position
data from actual machine motion to calculate the machine motion.

Inputs of the desired torques at the adjustable and fixed links of the return phase legs are the second and third options. These are entered as functions of the adjustable and fixed links' angular rotation, respectively. These options are used to study aided returns of the swing leg. These are returns that are a combination of applied power and gravity force. A designer can model springs or other mechanisms as a means of energy storage for leg return by correctly using these options.
Chapter 4

Mechanical Design of DUWE

4.1 Introduction

This chapter describes the detailed mechanical design of the DUWE vehicle. This machine was designed and built to test the ideas presented in this thesis and to validate the computer simulation program of Chapter 3.

The DUWE is a six legged vehicle. It is constrained, via mechanical coupling of legs into a tripod set, to walk in a tripod gait. A picture of the machine and towing cart is provided in Figure 4.1. A four bar linkage leg geometry similar to the Monopod is used. The adjustable link has two effective lengths, one for the support phase and one for the return phase. This adjustment is controlled by a solenoid toggle mechanism. A solenoid latching mechanism on each tripod set controls the phase between the two sets. Position and velocity sensors are mounted on one leg of each tripod set. This information is fed to a microprocessor which controls the actuation of the eight solenoids. The remainder of the machine is unpowered, hence the project title: Dynamic "Unpowered" Walking Experiment.

63
Figure 4.1 Picture of DUWE with Towing Cart
The DUWE is operated by towing it across the floor. The driving force for the support legs is provided by the friction interface between the feet and the floor. At the end of the stride, the opposite tripod assumes support of the machine. The adjustable links of the initial tripod set are shortened allowing these legs to clear the ground as they pendulate through the return cycle. These return legs are latched ahead of toe down position until the support legs complete a stride. The adjustable link of the latched legs is then lengthened and the latch deactivated. This allows the latched set to again support the machine and to complete one cycle of operation. In essence, the DUWE can be considered a wagon with legs instead of wheels.

An unpowered walking machine enables the study of the kinetic/potential energy exchange principles at a minimal cost. The initial investigation did not warrant the additional complexity and time effort required for the design of a fully powered machine.

There are three advantages to the DUWE.

1. Kinematic and Dynamic mechanism principles can be studied without the constraints imposed by actuators and their control schemes.

2. Measurements of the towing force and speed allow power consumption measurements for legged locomotion without the losses inherent in actuation systems.
3. Passive control of machine height by a four bar linkage, planar leg is validated in a completely legged vehicle.

4.2 Preliminary Design

The leg geometry of the DUWE is similar to the Monopod leg. Table 4.1 gives the design values used for the linkage leg. These values are consistent with the nomenclature defined in Chapter 2, although the structure differs due to the different loading and operating conditions experienced. The frame dimensions were chosen to be similar to the Hexapod.

<table>
<thead>
<tr>
<th>L_1</th>
<th>6.774 in.</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_2</td>
<td>8.224 in.</td>
</tr>
<tr>
<td>L_3</td>
<td>3.772 in.</td>
</tr>
<tr>
<td>L_5</td>
<td>10.540 in.</td>
</tr>
<tr>
<td>L_x</td>
<td>3.170 in.</td>
</tr>
<tr>
<td>L_y</td>
<td>3.620 in.</td>
</tr>
<tr>
<td>\phi_L</td>
<td>90.00 deg.</td>
</tr>
</tbody>
</table>

The machine weight was estimated at 75 pounds. With this estimate, the leg components were sized.
To aid in the development, a cardboard model of the leg was constructed as shown in Figure 4.2. The mass distribution of each link of the model was modified for proper swing characteristics. This distribution provided the initial approximation of the required mass distribution of the real leg. The model also served as initial validation for the computer simulation.

An assembly drawing, shown in Figure 4.3, of the preliminary design concept was completed. Then the detail design of components commenced.

4.3 Adjustable Link Design

The most complicated component of the leg is the adjustable link of the linkage. Design constraints included the following.

1. Minimum adjustment of .25" for clearance of foot during the return phase.
2. Load carrying capacity equal to the maximum load during the support phase.
3. Operation by a solenoid on/off actuator.

A toggle mechanism was selected to shorten the link. This allows the solenoid to lift the leg but permits the link structure to carry the load during the support phase. A schematic of the toggle mechanism is shown in Figure 4.4. The solenoid force decreases as the stroke increases.
Figure 4.2  Cardboard Model of DUWE Leg
Figure 4.3 Assembly Drawing of DUWE
Figure 4.4 Schematic of Adjustable Link Toggle Mechanism
toggle mechanism is advantageous in this situation because the mechanical advantage of the arrangement increases as the solenoid force decreases. A picture of the actual hardware components is provided in Figure 4.5.

The main body of the link is constructed from 1 1/2" square aluminum tubing with 1/8" wall thickness. This has more than adequate strength and provides sufficient structure for bearing mounts and other attachments. It was also readily available from the supplier.

The main slide plate has three slots milled in it as shown in Figure 4.6. The two 1/4" vertical slots provide guides for the shoulder screws of the aluminum tube to slide (see Figure 4.5). This produces the desired relative motion between the slide plate and aluminum tube for shortening the link. The horizontal slot allows motion of the solenoid slide. The plate material is 1040 steel. After limited use, there has been no signs of wear.

The solenoid slide shown in Figure 4.7 was made first from bronze. However, there were stiction problems in the operation of the adjustable link. To alleviate this problem the solenoid slide material was changed to teflon. With this, stiction was reduced to a fraction of the original value. The slide is spring loaded in the locked position with the link extended. When the solenoid is activated, the
Figure 4.5 Picture of Adjustable Link Components
Figure 4.6 Drawing of Toggle Mechanism's Main Slide Plate

Figure 4.7 Drawing of Toggle Mechanism's Solenoid Slide
spring is elongated. Upon release of the solenoid, the spring returns the slide to the locked position.

Two steel links couple the solenoid slide to the lower link. These links are placed on either side of the main slide plate. Teflon and aluminum spacers are added to ensure proper operation.

The final parts of the adjustable link are the end supports consisting of the shaft plate and the counterweight plate. The shaft plate is constructed from 1/8" 1040 steel plate which is permanently brazed to the bearing shaft. This is necessary for coupling legs in a tripod set. The counterweight plate is constructed of 1/8" aluminum plate. It serves as a mounting plate for lead weights and the solenoid actuator. The weights are used for changing the mass distribution of the link.

The adjustable link is not designed to support side loading. This moment is carried by the remaining linkage. The completed assembly of the link is shown in Figure 4.8.
Figure 4.8 Picture of Adjustable Link Assembly
4.4 Fixed Link Design

The second major component of the linkage leg is the fixed link. It is constructed of 1 1/2" square, aluminum tubing with 1/8" wall thickness. This structure provides the required stiffness to support the side loads on the leg. The link is drilled for location of two rotational encoders used for linkage position measurement. Mounting the encoders inside the tube protects them from the environment and makes the link compact. The encoders are attached to the joint by Nordex timing belts and pulleys. Bearings are pressed into each end of the link in a duplex arrangement. The final parts of the link are the counter weights and counter weight mounting plate. These are used in changing link mass distribution. The resulting structure is shown in Figure 4.9.

4.5 Coupler Link Design

The final component of the leg is the coupler link of the four-bar linkage. The completed assembly is shown in Figure 4.10. This represents the knee, shank, and foot of the leg. There are two major components of the coupler link. The first is the knee plate which is made from 1/4" aluminum plate. It bolts firmly to the fixed link and is straddled by the adjustable link duplex bearing arrangement.
Figure 4.9 Picture of Fixed Link Assembly
Figure 4.10 Picture of Coupler Link Assembly
After initial testing this arrangement was stiffened by adding a 1/8" thick aluminum plate, visible in Figure 4.12, to the outside of the link. The second part of the square aluminum tubing with a 1/16" wall thickness. However, it was difficult to obtain this material and 1/8" thick wall tubing was used instead. An attempt to lighten the structure by drilling holes through the tube wall was implemented. The shank bolted to the knee plate using 5/16" cap screws. The foot is constructed of a 1/2" wooden dowel, sheathed with a rubber tubing, and screwed to the bottom of the shank. This cylindrical foot is light-weight and sturdy, and prevents scuffing of the laboratory floor.

4.6 Bearing Selection

There are 52 rotational bearings required. The design process was standardized using 3/8" bore flanged ball bearings. All bearings are mounted using the duplex arrangements shown in Figure 4.11. This gives a stiff joint with adjustable preload.

This completed the leg design. A picture of the resulting structure is shown in Figure 4.12
Figure 4.12 Picture of DUWE Leg Assembly
4.7 Frame Design

The next major design problem was the frame design. For simplicity, it was constructed from 1 1/2" square by 1/8" thick aluminum tubing welded into a rectangle, 36" by 72". Design calculations [21] indicated the frame was stiffness rather than strength limited. However a kingpost/guide wire arrangement could be easily installed for additional stiffness if required. Initial testing proved this to be unnecessary.

Mounting the legs to the frame is accomplished by 1/4" aluminum plate supports which are shown in Figure 4.12. These are bolted directly to the frame. A 1/8" aluminum plate located at the fixed link joint was added for additional stiffness, which greatly reduced the bending deflection experienced at this joint.

4.8 Coupling and Latching Mechanism

A cable pulley system was designed to couple the legs in a tripod set. Close examination of Figure 4.1 reveals the arrangement used. A 6 in., die cast, V-belt pulley is mounted to the shaft of the fore and aft legs', adjustable link. A third pulley is mounted to the latching mechanism (see Figure 4.13). Stainless steel cable (200 lb. test) is pinned to each pulley and wrapped to fix the pulley
rotation. The system uses three turn-buckles to adjust the tension in the cable and to adjust the relative rotation of the pulleys with respect to each other. A 1/2" dia. stainless steel drive shaft, with universal joints at each end, connects the third pulley with the middle leg's adjustable link on the opposite side of the machine. This couples the legs in a tripod stance. The drive shaft has significant torsional compliance; however, this does not affect the operation of the machine. This compliance could be reduced by replacing the stainless shaft with a larger diameter hollow shaft.

The latching mechanism is a modified block brake. A 3" aluminum disc is keyed with a standard 1/8" key to the 3/8" shaft of the middle pulley of each tripod set. Latching the disc effectively brakes the entire tripod set.

Original design calculations were for a block brake to be self locking against the disc. This brake was actuated by a solenoid. In initial tests however, the solenoid could not disengage the block. The brake therefore was changed to an adjustable latching mechanism. A hole was drilled into the brake disc to insert a latch pin. The brake block was modified to have a steel shim screwed to the top. The block is then inserted under the pin to latch the leg. Adjustments in braking height are accomplished by placing additional shims under the steel shim of the block. This
arrangement provides the required latching torque to latch the tripod set using the solenoid actuator.

4.9 Towing Cart

A standard lightweight utility cart is used to tow the DUWE. The cart carries the microprocessor and power supplies for the solenoids. A strip-chart recorder is also loaded in the cart.

4.10 Force Sensor

A force sensor, shown in Figure 4.14, was designed to measure the towing force required to pull the DUWE. It consists of a load ring (see Appendix C) with four resistance strain gages bonded to it. The gages form the sides of a Wheatstone Bridge circuit. The load ring is mounted in a towing bar for the DUWE. Universal joints are attached to each end of the towing bar to ensure that only tension/compression forces are applied to the ring. The ring is designed for force measurements up to 25 pounds in either tension or compression.
Figure 4.14 Picture of Load Ring Force Transducer
4.11 System Improvements

After complete assembly of the machine, some problems were encountered. Most of these have been explained in previous text regarding the stiffness and brake problems. However, one rather serious problem not foreseen in the design stages occurred and is described below.

Figure 4.15* shows the original Monopod linkage kinematic design with the foot path traced for support and return phase. The support path is not perfectly straight and is rotated with respect to the horizon. When the support leg is at the point corresponding to the minimum leg height, the return leg has minimum ground clearance. This clearance is further reduced by any compliance in the support leg. This proved to be insufficient clearance except for negotiation of very flat surfaces. Since the laboratory floor is too uneven, the leg design had to be changed to get better clearance. This problem is not apparent on the Monopod for the following two reasons.

1. The test cart frame remains at a constant height during the return phase. In contrast, the DUWE frame height is a function of the support leg height.

*This output is from a computer-aided-design program for four-bar linkages[].

87
Figure 4.15 Foot-trace of Original Monopod Leg Design
2. The adjustable link of the Monopod has a travel of 2.51" which is sufficient to compensate for most irregularities in the foot trajectory and walking surface.

To alleviate this problem in the DUWE vehicle, alterations to the original four-bar linkage were required. These modifications were kept to a minimum since considerable machine work had already been completed. Modification of the adjustable link or the fixed link proved difficult and costly. Attention was focused on the frame mounts and the coupler link. Small perturbations from the design values were studied using the four-bar analysis program mentioned above. A suitable linkage design is shown in Figure 4.16. The linkage changes are noted in Table 4.2. All of these changes were easily implemented on the actual structure by slotting mounting holes. This avoided machining new parts.

<table>
<thead>
<tr>
<th>Description</th>
<th>Old Value</th>
<th>New Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_3 )</td>
<td>3.772 in.</td>
<td>4.050 in.</td>
</tr>
<tr>
<td>( L_5 )</td>
<td>10.540 in.</td>
<td>9.900 in.</td>
</tr>
<tr>
<td>( L_y )</td>
<td>3.620 in.</td>
<td>3.500 in.</td>
</tr>
<tr>
<td>( \phi_L )</td>
<td>90.00 deg.</td>
<td>94.00 deg.</td>
</tr>
</tbody>
</table>
Figure 4.16 Foot-trace of Improved DUWE Leg Design
There are three significant changes in the linkage operation.

1. The minimum clearance for the return phase is more than twice the previous amount.

2. The foot trajectory is staighter and more horizontal.

3. The loading on the adjustable link is smaller, thus reducing the deflection in the leg.

These changes were sufficient for correct operation of the DUWE.

This completed the DUWE design. Additional component specifications are included in Appendix C.
Chapter 5

Electrical Design of the DUWE vehicle

5.1 Introduction

Selection and assembly of the electronic hardware for the DUWE vehicle is discussed in this chapter. The wiring diagram is given in Appendix D.

5.2 Microprocessor controller

An Apple II Plus microcomputer with 48K bytes memory was chosen for the controller. This processor has sufficient computation speed for the control algorithms implemented. It also allows the DUWE to be portable and independent of a stationary computer used by most laboratory walking machines. The computer and all required interfaces were readily available at a reasonable cost. There was additional equipment purchased. A Pascal language card converted the Apple from Basic to Pascal computer language.*

*Pascal is the standard computer language of the walking machine project.
This added an additional 16K bytes of memory to the computer. The DI09 Digital Interface card from Interactive Systems Inc. gives 32 digital I/O lines for interfacing with external circuitry. This card plugs directly into the Apple. A 5" floppy disc drive for program storage and a standard TV screen monitor were the final major components. Various cables and connectors were required to complete the system.

5.3 Position measurement

Position measurement of the DUWE legs is simplified due to the mechanical coupling of the legs. Only one leg of each tripod set is instrumented. Each leg has two degrees of freedom, one is the general linkage, the other is the adjustable link. To determine the position of the linkage, two joint angles must be known. If both angles are at opposite ends of the same link the computational time to calculate the joint angles of the remaining linkage is reduced. The DUWE leg has a 256 bit, Librascope, gray code encoder (see Figure 5.1), mounted at each end of the fixed link. The encoder motion is amplified by a Nordex timing belt arrangement (see Figure 4.9), to give approximately a full revolution in the angle limits of the linkage. A 2.5:1 amplification is used at the mount end corresponding to a
Figure 5.1 Picture of Librascope Gray Code Encoder

Figure 5.2 Picture of Guardian T12x19 Solenoid
resolution of 0.563 deg/bit. The coupler end encoder has a 4:1 amplification for a 0.352 deg./bit resolution. The four encoders, 2 for each set of legs, are wired to the 32 I/O lines of the DI09 interface.

5.4 Leg and Brake Actuators

There are eight actuators required, one for each adjustable link and one for each of the two latches. Electric solenoids provided the most effective actuation system. Selection of these solenoids is based on the following criteria.

1. Maximum force required - approximately 5 lbs. at end of stroke.

2. Lightweight to reduce the effect on the mass distribution of the leg.

3. Minimum of 1" total stroke.

The Guardian Txl9, 24 VDC continuous duty solenoid of Figure 5.2 is the best selection. The maximum force is 180 oz. at 1/8" stroke. The leg actuators are pull type solenoids while the brake actuators are push type. Under actual operation, 60 VDC is supplied to give the solenoid larger force capacity. This is possible because the solenoid operation is intermittent rather than continuous.
Also, mounting the solenoids on aluminum plates provides a sufficient heat sink to prevent over-heating.

The solenoids are wired to the Interactive Systems Inc. P8 Optical Isolator shown in Figure 5.3. This performs the function of a relay but prevents the solenoid supply power from damaging the electronics of the computer. The isolator is connected to the four Game I/O annunciator ports of the Apple computer. These ports are toggle switches which can turn the optical relays on and off. Only four such switches are required, one for each tripod leg set and one for each leg latch.

5.5 Recording Devices

The remaining electronics include two Helipot, model SJ660 rotary potentiometers (see Figure 5.4). They are directly coupled to the fixed link of the front leg of each tripod set. These potentiometers are connected to a strip chart recorder to obtain time history plots of the leg position during walking.
Figure 5.4 Picture of Helipot rotary Potentiometer
Chapter 6
Control Program

6.1 Introduction

The development of efficient control algorithms has been emphasized for past walking machines. At the Ohio state University's Digital Systems Laboratory, studies have progressed using the Monopod and Hexapod [3,6,22,23]. Unfortunately, the algorithms do not readily apply to an unpowered machine such as the DUWE. For this reason, the specialized control scheme, presented in the following sections, was developed.

6.2 Control Algorithm

A simple control approach is implemented to take full advantage of the mechanical logic of the DUWE. There are eight electric solenoids, one on each of the six legs and one on each of the two latches, which must be controlled. The solenoids are two state devices and are either on or off. Since the legs are coupled in a tripod, leg solenoids are grouped in sets of three. This reduces the controlled variables to four on/off switches.
A finite state control algorithm based only on leg position is used. Walking is basically a two state process for each tripod leg set. The legs are either supporting some fraction of the machine weight or returning under no load in preparation for the support phase. The DUWE utilizes two additional states to ensure proper phasing between tripod leg sets. These states are latched states where the legs of a tripod set are latched in position prior to toe-down. These two states are necessary when non-resonant walking (low speed walking when the return leg period is shorter than the stride period) is encountered. The four solenoid states are given below in Table 6.1.

Table 6.1 States of the Solenoid Actuators

1. Latch solenoid off....Leg extended in support
   Leg solenoid off     phase.

2. Latch solenoid off....Leg shortened in return
   Leg solenoid on      phase.

3. Latch solenoid on.....Leg latched in shortened
   Leg solenoid on      position.

4. Latch solenoid on.....Leg latched in extended
   Leg solenoid off      position.

It is interesting to note that only states 1 and 2 are required for resonant walking. Also, during non-resonant
walking, a smooth toe down can be accomplished with the leg solenoid off because of the path traced by the foot at toe-down (see Figure 4.16). This allows instantaneous use of state 3 during the latching phase.

The DUWE walks by properly controlling the above four solenoid states for each tripod set while the machine is towed along the floor. The flow chart of Figure 6.1 gives the control algorithm based on position measurement of each tripod set. Three position control variables, which are listed below, are required.

1. Theta Release: Fixed link angle of support leg at which the latch of the return leg is released.

2. Theta Toe Down: Fixed link angle of non-support leg at which toe down is reached. This signals the support leg to shorten and begin return.

3. Theta Latch: Fixed link angle of return leg at which latching mechanism is activated and leg is lengthened.

The corresponding four solenoid states of each tripod set are controlled by these variables. When "theta release" is reached, the state of the non support leg changes from 4 to 1. At "theta toe down" of the non-support leg, the state of the support leg solenoid is switched from state 1 to 2. The non-support leg then takes over machine support. The return leg solenoid state is switched to state 3 as "theta
Figure 6.1 Control Program Flow Chart
latch" is reached. Almost immediately the leg is lengthened, corresponding to state 4, in preparation for the support phase. This completes a cycle of operation.

There are disadvantages of this control algorithm. With no velocity feedback, changes in machine speed affect the control quality. Strip chart recordings of leg position presented in Chapter 7 indicate poor transitions of toe-down/toe-off for certain speeds. Also, time lags of the solenoids can be compensated only by perturbing the position control variables from their optimum positions. Perhaps the worst disadvantage is that no corrective action is taken when the machine reaches an unstable speed. The result is that the machine falls. This situation is mechanically overcome by three castered leg stops on the frame which catch the machine and prevents any hardware damage. However, for the initial testing, this control algorithm had the advantage of being easily implemented. Most of the control errors could be minimized by changing the position control variables for different operating speeds. The DUWE performed more than adequately under this control.
6.3 Computer Program

The main control program is given in Appendix E. The program is written in Pascal and Assembler language compatible with the Apple II Plus micro computer [24,25,26,27]. Assembler language routines were required for the high speed I/O to and from the encoders and solenoids.

6.3-1 Program Operation

The program first prompts the user for input of the position control variables; "theta release", "theta toe down", and "theta latch". These values are entered directly as encoder readings rather than degrees. After input, the program begins executing. At this time the legs of the DUWE must be manually initialized by placing one of the tripod sets in the latched position. From this point the machine towing may commence. Encoder values are continuously sampled as the machine is pulled across the floor. Proper solenoid actuation is carried out by the microprocessor.
Chapter 7

Validation of Computer Simulation Program

7.1 Introduction

This chapter describes the methods and results of the testing used to validate the accuracy of the computer simulation program previously developed. Measurements of leg motion from actual test runs of the DUWE vehicle are compared with predicted leg motion from the computer simulation.

7.2 Parameter Measurement

The computer simulation program requires the mass, location of mass, and mass moment of inertia about the center of mass for each link of the linkage leg. Also, the bearing friction parameters at each joint are needed.

The mass of each link was determined using a standard scale balance. Figure 7.1 illustrates the method used for locating the center of mass of the link. The link is hung in two different configurations from a fulcrum. An imaginary vertical line is drawn on the link for each position. The intersection of these lines locates the
Figure 7.1 Method of Locating Link Center of Mass
center of mass of the link.

The mass moment of inertia about the center of mass of each link was determined using swing tests. Each link was suspended from the fulcrum (see Figure 7.2) and swung with a small amplitude of vibration. The frequency of the oscillation was measured. From this the mass moment of inertia about the center of mass was calculated using equations derived in Reference 28. The equations are listed below.

\[ J = \frac{m l g}{\omega} \]

\[ J_{cg} = J - m l^2 \]

where:

\( J \ldots \) Mass moment of inertia about the point locating the fulcrum.

\( J_{cg} \ldots \) Mass moment of inertia about the center of mass of the link.

\( m \ldots \) Mass of the link.

\( l \ldots \) Distance between center of mass of the link and the point locating the fulcrum.

\( g \ldots \) Acceleration of gravity.

\( \omega \ldots \) Frequency of oscillation.
Figure 7.2 Method of Determining Link Mass Moment of Inertia
The results of the mass parameter measurements are listed in Table 7.1. These values represent average values.

<table>
<thead>
<tr>
<th>Link</th>
<th>( m ) (lbf. sec(^2)/in.)</th>
<th>( J_{cg} ) (lbf. in. sec(^2))</th>
<th>( L_m ) (in.)</th>
<th>( \theta_m ) (deg.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0044</td>
<td>0.050</td>
<td>1.50</td>
<td>71.0</td>
</tr>
<tr>
<td>2</td>
<td>0.0105</td>
<td>0.094</td>
<td>1.60</td>
<td>55.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0039</td>
<td>0.062</td>
<td>2.08</td>
<td>115.0</td>
</tr>
</tbody>
</table>

for each of the six legs of the DUWE. The values are consistent with the coordinate system defined in Figure 2.3 of Chapter 2.

The bearing friction parameters include the coefficient of friction, \( \mu \) and effective radius, \( R \) as described in Chapter 2. Standard rolling element ball bearings are used at all joints. The value for \( R \) is calculated from the mean radius of the bearings. \( \mu \) was predicted from standard handbook values. The high estimates of \( \mu \) resulted from the quality of the bearings which under small preloads operated very inefficiently. Inexpensive flanged ball bearings were used to keep the cost down. For joint D, the coefficient is even higher to compensate for the coupling of the tripod.
set. The resulting values are given in Table 7.2.

TABLE 7.2 Measured Values of Bearing Parameters

<table>
<thead>
<tr>
<th>Joint</th>
<th>$\mu$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint A</td>
<td>0.5</td>
<td>0.32 in.</td>
</tr>
<tr>
<td>Joint B</td>
<td>0.5</td>
<td>0.32 in.</td>
</tr>
<tr>
<td>Joint C</td>
<td>0.5</td>
<td>0.32 in.</td>
</tr>
<tr>
<td>Joint D</td>
<td>0.8</td>
<td>0.32 in.</td>
</tr>
</tbody>
</table>

7.3 Testing Procedure

The testing procedure consisted of towing the DUWE along the floor at several operating speeds. Care was taken to prevent rapid accelerations in the machine motion; however, smooth accelerations were acceptable. Angular position data of the fixed link of each tripod set was recorded using a pen, strip-chart recorder. A sample record, with the appropriate markings, is shown in Figure 7.3. This position/time record was then analyzed.
VOLTAGE OUTPUT FROM FIXED LINK POTENTIOMETERS
5V EXCITATION; 2.5V FULL SCALE; 25 mm./sec.

Figure 7.3 Sample Position/Time Record from Pen, Strip-Chart Recorder
The position/time data for the support leg's fixed crank rotation were digitized for computer analysis. Velocity and acceleration information were obtained by fitting a spline curve [19] through the data and differentiating the result with respect to time. In this manner, the complete motion history of the fixed link of the four-bar linkage was described. The machine motion was then calculated using the kinematic equations developed in Chapter 2.

At this point the computer simulation was used to predict the motion of the return phase legs. The angular position of the fixed link was collected as a function of time. The results were then graphically compared with actual position/time measurements of the return leg's fixed crank rotation under identical stride characteristics. The maximum error of the simulation prediction is computed as a percentage of the full scale motion experienced by the fixed crank of the measured return. This procedure is completed for several average speeds of the DUWE.

Some discretion is necessary when picking the position/time record to be analyzed from the DUWE measurements. Close examination of the sample record of Figure 7.3 shows that the second return phase of Tripod Leg Set 2 is obstructed at the beginning of the swing. This is caused by irregularities in the floor surface and small
errors in the toe-off and toe-down timing. However, the initial return stroke of Tripod Leg Set 2 is much smoother, indicating an unobstructed return. Since the computer simulation models a perfect return with no ground interference, it is appropriate to choose the initial return stroke for validation comparisons.

7.4 Validation Results

The results of the testing are presented graphically in Figures 7.4 thru 7.7. Table 7.3 summarizes these results.

<table>
<thead>
<tr>
<th>AVERAGE SPEED</th>
<th>MAXIMUM ERROR</th>
<th>PERCENT FULL SCALE</th>
</tr>
</thead>
<tbody>
<tr>
<td>in./sec</td>
<td>deg.</td>
<td>%</td>
</tr>
<tr>
<td>5.80</td>
<td>10.4</td>
<td>10.1</td>
</tr>
<tr>
<td>7.87</td>
<td>5.0</td>
<td>4.6</td>
</tr>
<tr>
<td>9.00</td>
<td>5.0</td>
<td>4.2</td>
</tr>
<tr>
<td>9.66</td>
<td>10.0</td>
<td>9.0</td>
</tr>
</tbody>
</table>

Upon inspection of these graphs, it is evident that the
Figure 7.4 Comparison of Predicted Leg Motion with Measured Leg Motion at 5.80 in./sec.
Figure 7.5  Comparison of Predicted Leg Motion with Measured Leg Motion at 7.87 in./sec.
Figure 7.6 Comparison of Predicted Leg Motion with Measured Leg Motion at 9.00 in./sec.
Figure 7.7  Comparison of Predicted Leg Motion with Measured Leg Motion at 9.66 in./sec.
simulation prediction always leads the actual measured motion. The reason for this could be errors in the prediction of the bearing friction. Changes to the friction coefficient at the joints can dramatically change the characteristics of the leg motion. Figure 7.8 shows the improved motion prediction after increasing the coefficient of friction of Joint A. In this figure, the period of the actual and predicted return phases are different but the amplitude of the motion is essentially the same.

Another probable source of error is that the simulation models only one leg with perfect toe-off/toe-down characteristics. The DUWE measurements are for a three legged, tripod set. Average mass characteristics of the tripod set are compared to ideal characteristics of one leg. This induces errors into the results which degrade the accuracy.

The final point to be made about the graphical results is the difference in predicted and actual latching heights. The DUWE was operated with a fixed latching height. Upon release, the toe-down velocity did not match the vehicle velocity. This was acceptable since the rubber feet absorbed the shock experienced in bringing the leg up to speed. However, the computer simulation predicts the necessary braking height for velocity matching at toe-down based on the instantaneous speed at toe-down of the ground.
Figure 7.8  Improved Prediction of Leg Motion
after Bearing Friction Adjustment
support leg. This explains the differences in braking height predictions for the average speeds of 7.87 in./sec. and 9.00 in./sec. shown in Figures 7.5 and 7.6.
8.1 Introduction

The final task; discussed by this thesis, is to compare the energy consumption of the DUWE with that of the Hexapod and Monopod systems. During the validation testing discussed in Chapter 7, towing force measurements were recorded using the towing bar and load cell described in Chapter 4. Also, solenoid power requirements were measured using a calibrated shunt across the 60 V DC power supply. These measurements were recorded on the pen, strip-chart recorder. The only power not considered was the power required to operate the Apple Computer.

8.2 Reduction of Energy Consumption Data

Data reduction of the strip chart recording is accomplished by digitizing the data and numerically calculating the desired quantities.

A sample force/time record from the DUWE testing is shown in Figure 8.1. The load cell calibration curve of Appendix C is only for tension loading. Compression loading
Figure 8.1 Force/Time Record of DUWE Towing Force
of the load cell, resulting from the DUWE pushing the cart, was estimated using linear interpolation. This does not create any problems since only positive forces corresponding to power input needed to be measured. Power output from the machine could not be recovered for use.

Instantaneous power curves were computed by multiplying the instantaneous force by the instantaneous velocity. Since only power inputs are of importance, negative power measurements were set equal to zero. Figures 8.2 thru 8.4 show typical instantaneous power curves for different operational speeds of the DUWE. Average power was calculated by integrating these curves and dividing by the total leg cycle time.

Solenoid power measurements were computed using the calibrated shunt output recorded by the strip chart recorder (see Figure 8.5). The shunt produces a linear voltage signal of 5.0 mV/amp. Calculating the instantaneous amperage and multiplying by the 60 V DC power source gave the instantaneous solenoid power. These results were time averaged to give the average solenoid power.

Table 8.1 gives the results of the DUWE energy audit for three different vehicle speeds. The latching and leg solenoids used approximately 15 to 25 times more energy than required to tow the DUWE. The reason for this is that the solenoids expend much of their energy as resistance
Figure 8.2 Towing Power Input at 6.20 in./sec.
Figure 8.3  Towing Power Input at 7.87 in./sec.
Figure 8.4  Towing Power Input at 9.66 in./sec.
Figure 8.5 Sample Solenoid Power Measurement from Pen, Strip-Chart Recorder
### TABLE 8.1 Power Audit of DUWE

<table>
<thead>
<tr>
<th>AVERAGE SPEED (in./sec.)</th>
<th>TOWING POWER (ft. lbf. sec.)</th>
<th>SOLENOID POWER (ft. lbf. sec.)</th>
<th>TOTAL POWER (ft. lbf. sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.20</td>
<td>1.35</td>
<td>20.8</td>
<td>22.2</td>
</tr>
<tr>
<td>7.87</td>
<td>1.52</td>
<td>30.7</td>
<td>32.2</td>
</tr>
<tr>
<td>9.66</td>
<td>1.35</td>
<td>30.9</td>
<td>31.3</td>
</tr>
</tbody>
</table>

### TABLE 8.2 Power Audit of DUWE with Improved Actuators

<table>
<thead>
<tr>
<th>AVERAGE SPEED (in./sec.)</th>
<th>TOWING POWER (ft. lbf. sec.)</th>
<th>LINEAR ACTUATOR POWER (ft. lbf. sec.)</th>
<th>TOTAL POWER (ft. lbf. sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.20</td>
<td>1.35</td>
<td>5.0*</td>
<td>6.4</td>
</tr>
<tr>
<td>7.87</td>
<td>1.52</td>
<td>6.0*</td>
<td>7.5</td>
</tr>
<tr>
<td>9.66</td>
<td>1.35</td>
<td>6.5*</td>
<td>6.8</td>
</tr>
</tbody>
</table>

*Estimated from Monopod data from Vohnout[3] (see Appendix E)
heating. Their operation is extremely inefficient. However, by replacing the solenoids with ball-screw, servo-motor operated linear actuators similar to that used on the Monopod, energy consumption would decrease to a fraction of the original value. This was not considered in the original design because of the additional cost and complexity required to implement the system.

For energy consumption comparisons, the actual DUWE and an imaginary DUWE using the ball-screw, servo motor actuators were analyzed. The results of the power and energy measurements are presented in Table 8.2. Estimates of the power required to run the ball-screw, servo motor actuators were taken from the Monopod measurements performed by Vohnout [3] (see Appendix C). The actual power requirements would probably be less since the components of the Monopod are much larger than the DUWE requires. However, to be conservative, the Monopod measurements were used.
8.3 Methods of Energy Comparison

Two methods were used by Vohnout [3] to compare the energy consumptions of the Monopod and Hexapod. The first method directly compared the energy consumption based on equivalent scaled leg loading. The second method used the dimensionless motion resistance coefficient developed by Gabrielli and Von Karman [30].

The DUWE vehicle weighs 80 pounds while the Hexapod weighs approximately 220 pounds. The design of the DUWE prohibited adding the additional 140 pounds to make the vehicles equal in weight. Thus direct energy consumption comparisons could not be validly applied here.

The second method of using the dimensionless motion resistance coefficient to compare the DUWE, Hexapod, and Monopod is applicable. The coefficient is termed Specific Resistance or Specific Tractive Force and is defined as:

\[ \epsilon = \frac{P}{WV} \]

where:

P.....Average power required (ft.-lb.\(_f\)./sec.)
W.....System gross weight (lb.\(_f\).)
V.....Velocity produced by P (ft./sec.)
Vohnout discusses the precautions one must take when comparing the DUWE, Hexapod, or Monopod to other vehicles plotted by Gabrielli and Von Karman. The problem is that the DUWE, Hexapod, and Monopod do not carry their own power supplies or sufficient electronics to control leg coordination. However the emphasis here is on comparing the DUWE with the Hexapod and Monopod and not the other vehicles.

Finally, the DUWE vehicle is inherently different from the Hexapod and Monopod because the DUWE's main driving force, the towing force, is external to the system. Thus the mechanical losses of a main drive actuation system are avoided. The only energy expended in towing the DUWE is that required to overcome environmental resistance and the mechanical losses of the leg linkage. This energy value is of importance when considering the minimal power requirements of a six legged vehicle.

8.4 Energy Consumption Comparisons

As stated earlier, the Specific Resistance, $\epsilon$, is used to compare the DUWE with the Hexapod and Monopod vehicles. Table 8.3 shows the results of the computation of $\epsilon$ for the DUWE and a DUWE with ball-screw, servo motor actuators. These results are graphically displayed in Figure 8.6 on a plot which was used by Vohnout to compare the Monopod and
TABLE 8.3 Specific Resistance of DUWE

<table>
<thead>
<tr>
<th>AVERAGE SPEED (in./sec.)</th>
<th>DUWE POWER (ft. lbf. sec.)</th>
<th>*DUWE POWER (ft. lbf. sec.)</th>
<th>DUWE SPECIFIC RESISTANCE</th>
<th>*DUWE SPECIFIC RESISTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.20</td>
<td>22.2</td>
<td>6.4</td>
<td>0.64</td>
<td>0.15</td>
</tr>
<tr>
<td>7.87</td>
<td>32.2</td>
<td>7.5</td>
<td>0.61</td>
<td>0.14</td>
</tr>
<tr>
<td>9.66</td>
<td>31.3</td>
<td>6.8</td>
<td>0.49</td>
<td>0.10</td>
</tr>
</tbody>
</table>

*Estimated from Monopod data from Vohnout[3] (see Appendix E)
Figure 8.6 Specific Resistance of DUWE, Monopod, and Hexapod
Hexapod vehicles. It is evident that the Specific Resistance of the DUWE is approximately equivalent to the Specific Resistance of the Monopod system even though the inefficient solenoid actuators were used. Approximation, of the Specific Resistance of a DUWE having ball-screw, servo motor actuators, indicates an improvement of energy consumption of about 5 times over the energy consumption of the DUWE and Monopod systems. It was mentioned in Chapter 1 that the Monopod already displayed a 25 times improvement in energy consumption over the Hexapod vehicle.

A plot of the Specific Resistance of the DUWE, Hexapod, and Monopod on the same graph as other vehicles studied by Gabrielli and Von Karman is given in Figure 8.7. It is shown that the DUWE, operating with efficient actuators, is comparable to other useful vehicular systems. However, recall that the DUWE does not carry its own power supply. Improvements in the control and operation of the DUWE should further decrease the energy consumption requirements and make the DUWE even more appealing as a useful vehicle.
Figure 8.7 Specific Resistance of DUWE, Monopod, and Hexapod Compared with other Vehicles Studied by Gabrielli and Von Karman

135
Chapter 9

Discussion, Conclusions, and Extensions

9.1 Introduction

In summary, the DUWE experiment was successful. The DUWE demonstrates that proper kinetic/potential energy interchange during the walking cycle of a leg can decrease the energy consumption of walking machines. Specifically, the energy consumption of the DUWE is approximately 5 times less than the Monopod and over 30 times less than the Hexapod. These results are based on a DUWE vehicle using linear actuators similar to those used on the Monopod. Since the DUWE actuators require less force than those of the Monopod, the above results tend to be conservative.

The computer simulation program was designed to predict the dynamic behavior of the leg during the kinetic/potential energy transfer mentioned above. Direct comparisons of the simulation results with actual experimental measurements of the DUWE motion were completed. The resulting errors, between predicted and actual leg motion, were in the range of 5% to 10%. These errors are considered acceptable for using the program as a tool for designing the optimum mass
distribution of future four-bar linkage legs since small perturbations from predicted leg motion can be corrected using additional counterweights.

Two conclusions which were not explicitly covered by the testing procedures are explained in the following paragraphs.

The DUWE demonstrates that active control of the machine frame height is unnecessary during cruise mode walking. The four-bar linkage leg geometry passively controls the height within acceptable levels. This point is important during the design of efficient walking machines since only the main drive actuator is operated during the support stroke. It is admitted that additional actuators are required for turning.

The second conclusion is that the DUWE's maximum speed, resonant speed, is approximately 12.0 in./sec. This speed is calculated from measurements of the foot stride and the return leg period. The DUWE is at a point of instability when operating at this speed because any increase in speed results in the return phase legs returning too slow. However, speeds of over 13.0 in./sec. were recorded by allowing the stride length to exceed the design value. This proved undesirable since the vertical variation in the machine operating height increased (see figure 4.16).
Before presenting possible extensions of the DUWE experiments, it is necessary to discuss some possible improvements to the DUWE system. The first improvement would be the re-design of the tripod coupling and latching mechanisms. The major flaw in the present design is that the coupling system is attached to the adjustable link instead of the fixed link. Although the DUWE works quite well in its present state, the new design would improve the operation. The reasons for this are listed below.

1. The torque requirements for latching the legs are smaller at the fixed link joint. This decreases the torque demand on the latching mechanism allowing it to operate more efficiently.

2. In the latched position, torque is transmitted through the adjustable link. This causes the slide of the toggle mechanism to experience higher loading which increases the force requirements to operate it.

3. Finally, the relative joint rotation of the fixed link with respect to the frame is greater than the relative rotation of the adjustable link. With the fixed links coupled together, misalignment errors between the legs of a tripod set would be less severe.

Implementation of this design change would require little additional machining or cost. Another suggested improvement is to replace the solenoid actuators with more efficient actuators such as the ball-screw, servo motor actuator discussed in Chapter 8. Minimally the solenoids should be replaced with stronger solenoids. This would eliminate the need to "over power" the ones currently installed and
increase the efficiency of the machine.

The final suggested improvement is to tow the DUWE in reverse. From examination of the Assembly Drawing of the DUWE shown in Figure 4.3, it can be inferred that operating the leg in reverse improves the dynamic characteristics of the return phase. Due to the mass distribution of the linkage and the friction losses in the bearings, counterweights are required to force the leg to return past the toe-down position when operated in the forward direction. Without the counterweights, the leg has more potential energy stored at the original toe-down position than at toe-off. This difference in potential energy is sufficient, without the addition of counterweights, to overcome the friction losses in the bearings during a reversed return swing. This is advantageous because the additional weight of the counterweights are avoided. It is admitted that operating the leg in reverse was initially overlooked due to viewing the forward direction initially used on the Monopod as correct.

A final comment on the DUWE operation is that the kinetic energy of the leg at toe-off during low speed walking is insufficient to power the leg through the return phase. Because of this, a certain fraction of the energy required for the return phase is stored in the leg by raising the center of mass of the linkage during the support
phase. Using this mode of operation seems equivalent to powering the legs through the return cycle because the energy requirements are similar. However in an autonomous walking machine, it is more efficient to store the energy for the return phase at such a time when the actuators are operating efficiently under full design load (the support phase) rather than powering the return phase with the actuators operating inefficiently under partial load. As the speed of the machine increases, sufficient kinetic energy is stored in the leg at toe-off and this mode of operation is no longer needed.

One obvious extension to the DUWE experiment is to incorporate velocity feedback into the control algorithm. This should eliminate most of the interference problems caused by improper timing of the toe-down/toe-off transition. Additional experiments should study the effects of leg mass distribution on the dynamics of the return phase. Walking up and down gentle inclines should be investigated. This would determine the sensitivity of leg dynamics to the angle at which gravity acts. Finally, other forms of potential energy storage such as springs should be implemented to completely study the kinetic/potential energy exchange ideas presented. This method of interchanging kinetic and potential energy during the leg cycle shows the greatest promise for a vehicle which must operate under a
wide range of speed and terrain conditions.

The final objective of these studies would be to incorporate the results into a powered leg design such as the Monopod. This would take walking machines one step closer to being competitive with other forms of land locomotion.
Bibliography


18. Cappozza, A., Leo, T., Pedetti, A., A General Computing Method for the Analysis of Human Locomotion


APPENDIX A

SIMULATION PROGRAM
FLOW CHART OF SIMULATION PROGRAM'S MAIN MENU SELECTION
FLOW CHART OF SIMULATION PROGRAM'S ANALYSIS ROUTINE
ANALYSIS ROUTINE FLOW CHART CONT.
C*** SUB
C
PROGRAM MAIN
PENDULATION OF WALKING MACHINE LEO
C
3-MAY-82 AUTHOR: FRANK BROWN
C
DESCRIPTION
C
THIS PROGRAM ANALYZES A FOUR-BAR LINKAGE WALKING MACHINE LEO.
It determines the time-history of the leg position as it pendulates through the return cycle. Only planar motion is considered. LEO adjustment for the return phase is provided by a step change in the adjustable link. The program is developed as part of the "DYNAMIC UNPOWERED WALKING EXPERIMENT".
C
PARAMETERS INPUT
NONE
C
PARAMETERS RETURNED
NONE
C
PARAMETERS INTERNAL
IFK : CONTROL FLAG USED IN DUMODL
: =0 DRAWS SAMPLE DUNE VEHICLE WITH WELCOM HEADING
: =1 DRAWS CURRENT VERSION WITHOUT HEADING
: =2 CALL WELCOME AND PARAM TO INITIATE PROGRAM
IFL : CONTROL FLAG FOR CREATING DATA FILE
: =0 READY TO CREATE FILE
: =1 CHANGES HAVE OCCURRED. ANALYZE BEFORE CREATE FILE
I : CONTROL FLAG OF MENU SELECTION
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES
C
EXTERNAL REFERENCES
DRFRM, DUMODL, OPTION, RDDAT, CHHEOM, CHFARA, CHBEAR,
CHFRM, ANALYS, LINFED, DRWTPG CREDAT, PLOTO ROUTINES
C
CALLING ROUTINES
NONE
C
CALL INITT(960)
CALL TERM(2.4096)
C
IFK=0

151
IFL=1
I=1

* INITIALIZE CONTROL FLAGS

10 CALL DRWFMRM
   CALL DUMODL(IFK)

* DRAW SAMPLE OF DUME

IF(IFK.EQ.2) CALL PARAM
   IF(IFK.EQ.2) CALL WELCOM

* PROGRAM DESCRIPTION

12 CALL OPTION(I,LINe)
   GO TO (1,2,3,4,5,6,7,8,9) I

1 CALL RDDAT(LINE)
   IFL=0
   GO TO 20

2 CALL CHGEOm
   IFL=1
   GO TO 10

3 CALL CHPARA
   IFL=1
   GO TO 10

4 CALL CHBEAR
   IFL=1
   GO TO 10

5 CALL CHSTRD
   IFL=1
   GO TO 10

6 CALL CHFRAM
   IFL=1
   GO TO 10

7 CALL ANALYS(LINE)
   IFL=0
   GO TO 20

8 IF(IFL.NE.1) GO TO 11
   I=0
   LINE=LINE-2
   CALL LINFED(LINE,-1)
   LINE=LINE+2
   CALL BELL
   WRITE(5,102)
102 FORMAT(12X,'RUN DATA OR NEW PARAMETERS FIRST.')
   GO TO 12

* RUN PARAMETERS AFTER A CHANGE BEFORE
* CREATING A DATA FILE

11 CALL CREDAT
GO TO 10

20 CALL LINFED(LINE, 2)
IFK=1
WRITE(5, 101)
101 FORMAT(12X, 'DRAW D.U.W.E.
READ(5, 200) ANS
200 FORMAT(A1)
IF(ANS.EQ. 'Y') CALL DRWTP(0, 0, LINE)
GO TO 10

9 CALL NEWPAG
LINE=18
CALL LINFED(LINE, 1)
CALL CHRSIZ(1)
CALL ANMODE
WRITE(5, 100)
100 FORMAT(19X, 'END OF D.U.W.E. ANALYSIS PROGRAM. ')

CALL CHRSIZ(4)
CALL FINIT(4800, 4800)
STOP
END
SUBROUTINE WELCOME

3-MAY-62 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE WRITES THE INTRODUCTION TO THE PROGRAM ONTO THE
SCREEN

PARAMETERS INPUT

NONE

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING ROUTINES

NONE

**************************************************************************

LINE=5
CALL LINFED(LINE,1)
WRITE(5,100)

100 FORMAT(59X,'"Duwe" ANALYSIS PROGRAM',//
'59X,'THIS IS AN INTERACTIVE DESIGN PROGRAM FOR DETERMINING',
'59X,'THE TIME HISTORY OF A 4-BAR LINKAGE ROBOT LEG AS IT',
'59X,'PENDULATES DURING THE RETURN CYCLE. THE INFLUENCE',
'59X,'OF MASS DISTRIBUTION, LEG GEOMETRY, AND INPUT TORQUES',
'59X,'ON THE RETURN PHASE CAN BE STUDIED. THE SIMULATION',
'59X,'MODELS A SIX LEG MACHINE CONSTRAINED TO WALK IN A TRIPOD',
'59X,'DATA ENTRY IS VIA THE KEYBOARD. CURSOR, OR DATA',
'59X,'FILE. A SAMPLE CASE, STORED IN THE PROGRAM, CAN BE',
'59X,'ACCESSED BY RUNNING NEW PARAMETERS. CHANGES TO THIS',
'59X,'CASE ARE ACCOMPLISHED BY THE MENU SELECTION SHOWN',
'59X,'AT LEFT."

* PARAGRAPH EXPLAINING THE PROGRAM

RETURN
END

154
SUBROUTINE VERACC(AX,AY,TIME)

C*******************************************************************************
C
C 3-MAY-82 AUTHOR: FRANK BROWN

C DESCRIPTION

C THIS ROUTINE DETERMINES THE VERTICAL ACCELERATION OF
C THE BODY.  IT IS BASED ON THE ACCELERATION OF THE FEET
C IN CONTACT WITH THE GROUND.

C PARAMETERS INPUT

L1 : LENGTH OF DRIVE CRANK
L2 : LENGTH OF DRIVEN CRANK
L3 : LENGTH OF COUPLER
L5 : LENGTH OF COUPLER SHANK
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT --LEFT POS.--
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT --UPWARD POS.--
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK
SHORTL : AMOUNT DRIVEN CRANK IS SHORTENED FOR
     RETURN PHASE OF LEG.
THETA : DRIVE CRANK ANGLE MEASURED FROM POS. X
     HORIZONTAL --CW--
GAMMA : DRIVEN CRANK ANGLE MEASURED FROM POS. X
     HORIZONTAL --CW--
PHI : COUPLER ANGLE MEASURED FROM POS. X
     HORIZONTAL --CW--
THDOT1 : ANGULAR VELOCITY OF DRIVE CRANK
     CW POS.
GADOT1 : " " " DRIVEN CRANK
     CW POS.
PHTDOT1 : " " " COUPLER
     CW POS.
THDOT2 : ANGULAR ACCELERATION OF DRIVE CRANK
     CW POS.
GADOT2 : ANGULAR ACCELERATION OF DRIVEN CRANK
     CW POS.
PHTDOT2 : ANGULAR ACCELERATION OF COUPLER
     CW POS.
COSTH : COSINE OF DRIVE CRANK ANGLE
COSGA : " " DRIVEN CRANK ANGLE
COSPH : " " COUPLER
SINH : SINE OF DRIVE CRANK ANGLE
SINGA : " " DRIVEN CRANK ANGLE
SINPH : " " COUPLER
PI : PI=4.0*ATAN(1.)
FOOTST : LENGTH OF LEG STRIDE
EXPLAC : DISTANCE FROM DRIVE CRANK SUPPORT TO
     TOE-DOWN POSITION --POSITIVE RIGHT--
EXPICK : DISTANCE FROM DRIVE CRANK SUPPORT TO
     TOE-OFF POSITION --POSITIVE RIGHT--
SPEED : SPEED OF MACHINE
PERIOD : TIME FOOT IS IN CONTACT WITH GROUND.
     =.5*TOTAL LEG PERIOD
INCRE : NUMBER OF INCREMENTS USED IN TIME STEPPING
INCI : INCREMENT OF "INCRE" AT WHICH POSITION IS
     SAVED FOR DRAWING
DELTIM : INCREMENT OF TIME STEP
AX : ACCELERATION OF BODY IN X-DIRECTION

155
PROGRAM PAGE = 134

C AY : ACCELERATION OF BODY IN Y-DIRECTION (IN./SEC2)
C BI : ANGLE FOR ACCELERATION OF GRAVITY (RAD.)
C Q : POSITIVE FOR DOWNSHILL SLOPE
C O : ACCELERATION OF GRAVITY (386 IN./SEC2)

PARAMETERS RETURNED
A1, AY

PARAMETERS INTERNAL
SSTHETA : TEMP. STORAGE OF THETA
STHETD : " " " THDOT1
STHEDD : " " " THDOT2
SSL2 : " " " L2
EXDOT2 : X-VELOCITY OF FOOT WITH RESPECT TO FRAME
EYDOT2 : Y-VELOCITY " " " " " "

EXTERNAL REFERENCES
ORLEG1, ANGLE, VELOC, ACCEL

CALLING Routines
COMACC

******************************************************************************

COMMON /A/ L1, L2, L3, L5, LX, LY, PHIL
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /D/ THDOT1, QADOT1, PHDOT1
COMMON /E/ THDOT2, QADOT2, PHDOT2
COMMON /F/ CSTDH, SINH, COSH, SINGA, SINGA, CSDPH, SINPH
COMMON /Q/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
# INCRE, INC1, DELTIM, SHORTL, SL2

C REAL L1, L2, L3, L5, LX, LY
C
SSTHETA=THETA
STHETD=THDOT1
STHEDD=THDOT2
SSL2=L2

* STORE VALUES FOR LATER USE
L2=SSL2

* SET L2 TO EXTENDED LENGTH

CALL ORLEG1(TIME, ORTHET, ORTHED, ORTHDD)
THETA=ORTHET
THDOT1=ORTHED
THDOT2=ORTHDD
CALL ANGLE
CALL VELOC
CALL ACCEL
COSPHA=CD(S(PHI+PHI))
SINPHA=SIN(PHI+PHI)

C * COMPUTE ANGLES, VELOCITIES, AND ACCELERATIONS
C * FOR COMPUTATION OF "AX" AND "AY"

C EXDOT2=-L2*GADOT2*SINGA-L2*GADOT1**2*COSGQA-L5*PHDOT2
   *SINPHA-L5*PHDOT1**2*COSPHA
C

C AX=-EXDOT2
C
C * COMPUTE X-ACCELERATION

C EVDOT2=-L2*GADOT2*COSGQA+L2*GADOT1**2*SINGA-L5*PHDOT2
   *COSPHA+L5*PHDOT1**2*SINPHA
C

C AY=-EVDOT2
C
C * COMPUTE Y-ACCELERATION

C THETA=STHETA
THDOT1=STHETD
THDOT2=STHETDD
L2=SSL2
CALL ANGLE
CALL VELOC
CALL ACCEL

C * RETURN VALUES TO PREVIOUS STATE BEFORE ROUTINE CALL

C RETURN
END
SUBROUTINE ACCEL

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE CALCULATES THE ANGULAR ACCELERATIONS OF THE
DRIVEN CRANK AND COUPLER GIVEN THE ANGULAR ACCELERATION OF
THE DRIVE CRANK.

PARAMETERS INPUT

L1 : LENGTH OF DRIVE CRANK (IN.)
L2 : LENGTH OF DRIVEN CRANK (IN.)
L3 : LENGTH OF COUPLER (IN.)
THD1 : ANGULAR VELOCITY OF DRIVE CRANK (CW POS.)
QADOT1 : " " " DRIVEN CRANK (CW POS.)
PNDOT1 : " " " COUPLER (CW POS.)
THD2 : ANGULAR ACCELERATION OF DRIVE CRANK (CW POS.)
COSTH : COSINE OF DRIVE CRANK ANGLE
COSOA : " " DRIVEN CRANK ANGLE
COSPH : " " COUPLER
SINTH : SINE OF DRIVE CRANK ANGLE
SINGA : " " DRIVEN CRANK ANGLE
SINPH : " " COUPLER

PARAMETERS RETURNED

QADOT2 : ANGULAR ACCELERATION OF DRIVEN CRANK (CW POS.)
PNDOT2 : ANGULAR ACCELERATION OF COUPLER (CW POS.)

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING Routines

ANALYS

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /D/ THD1, QAD1, PND1
COMMON /E/ THD2, QAD2, PND2
COMMON /F/ COSTH, SINTH, COSOA, SINGA, COSPH, SINGPH

REAL L1, L2, L3, L5, LX, LY

QADOT2=(L3*COSPH)*(THD2*L1*SINTH+THD1*L1*COSTH-
* QADOT1**2*L2*COSQA-PHDOT1**2*L3*COSPH-(L3*SINPH)*
  (THDOT2*L1*COSTH-THDOT1**2*L1*SINTH+QADOT1**2*L2*SINGA+  
  PHDOT1**2*L3*SINPH))/(L2*SINGA*L3*COSPH-L2*COSQA*L3*SINPH)

C C C
* CALCULATE QADOT2

PHDOT2=-(L2*SINGA)*(THDOT2*L1*COSTH-THDOT1**2*L1*SINTH+  
QADOT1**2*L2*SINGA+PHDOT1**2*L3*SINPH)-(L2*COSQA)*
  (THDOT2*L1*SINTH+THDOT1**2*L1*COSTH-QADOT1**2*L2*COSQA-  
  PHDOT1**2*L3*COSPH))/(L2*SINGA*L3*COSPH-L2*COSQA*L3*SINPH)

C C C
* CALCULATE PHDOT2

RETURN
END
SUBROUTINE ANALYSIS(LINE)

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE PERFORMS THE DYNAMIC ANALYSIS OF THE LEG USING
4TH-ORDER RUNGE KUTTA TECHNIQUES. THE PROBLEM IS DEFINED AS
AN INITIAL BOUNDARY VALUE PROBLEM. THE FOOT VELOCITY AND
POSITION AT PICP ARE THE INITIAL CONDITIONS. THE SOLUTION
STEPS FORWARD IN TIME UNTIL TOEDOWN IS REACHED.

PARAMETERS INPUT

LINE :LINE SPACING PARAMETER FOR SCREEN WRITES
L1 :LENGTH OF DRIVE CRANK (IN.)
L2 :LENGTH OF DRIVEN CRANK (IN.)
L3 :LENGTH OF COUPLER (IN.)
L5 :LENGTH OF COUPLER SHANK (IN.)
DX :X DISTANCE FROM DRIVE CRANK SUPPORT TO
:DRIVE CRANK SUPPORT --LEFT POS.-- (IN.)
DY :Y DISTANCE FROM DRIVE CRANK SUPPORT TO
:DRIVE CRANK SUPPORT --UPWARD POS.-- (IN.)
PHIJ :ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
PI :PI=4.0*ATAN(1.)
FOOTST :LENGTH OF LEG STRIDE (IN.)
EXPLAC :DISTANCE FROM DRIVE CRANK SUPPORT TO
:TOE-DOWN POSITION --POSITIVE RIGHT-- (IN.)
EXPICK :DISTANCE FROM DRIVE CRANK SUPPORT TO
:TOE-OFF POSITION --POSITIVE RIGHT-- (IN.)
SPEED :AVERAGE SPEED OF MACHINE (IN./SEC.)
PERIOD :TIME FOOT IS IN CONTACT WITH GROUND. (SEC.)
=.5*TOTAL LEG PERIOD (SEC.)
INCRE :NUMBER OF INCREMENTS USED IN TIME STEPPING
INC1 :INCREMENT OF "INCRE" AT WHICH POSITION IS
:SAVED FOR DRAWING
DELTIM :INCREMENT OF TIME STEP (SEC.)
SHORTL :AMOUNT DRIVEN CRANK IS SHORTENED FOR
:RETURN PHASE OF LEG (IN.)
THETAM :ANGLE LOCATION OF CENTER OF MASS OF DRIVE
:CRANK. CW FROM DRIVE CRANK. (RAD.)
GAMHAM :ANGLE LOCATION OF CENTER OF MASS OF DRIVEN
:CRANK. CW FROM DRIVEN CRANK. (RAD.)
PHIK :ANGLE LOCATION OF CENTER OF MASS OF COUPLER.
:CW FROM COUPLER CRANK. (RAD.)
LM1 :DISTANCE FROM DRIVE CRANK SUPPORT TO CENT
:OF MASS OF DRIVE CRANK. (IN.)
LM2 :DISTANCE FROM DRIVEN CRANK SUPPORT TO CENTER
:OF MASS OF DRIVE CRANK. (IN.)
LM3 :DISTANCE FROM COUPLER JOINT ON DRIVEN CRANK
:TO CENTER OF MASS OF COUPLER (IN.)
M1 :MASS OF DRIVE CRANK (LBF. SEC2/IN.)
M2 :MASS OF DRIVEN CRANK (LBF. SEC2/IN.)
M3 :MASS OF COUPLER (LBF. SEC2/IN.)
JCQ1 :MASS MOMENT OF INERTIA OF DRIVE CRANK (LBF. IN. SEC2)
PROGRAM PAGE = 7

C JCG2 : MASS MOMENT OF INERTIA OF DRIVEN CRANK (LBF. IN. SEC2)
C JCG3 : MASS MOMENT OF INERTIA OF COUPLER (LBF. IN. SEC2)
C RA : BEARING RADIUS AT JOINT A (DRV CRK SUP) (IN.)
C RB : BEARING RADIUS AT JOINT B (DRV CRK END) (IN.)
C RC : BEARING RADIUS AT JOINT C (DRVN CRK SUP) (IN.)
C RD : BEARING RADIUS AT JOINT D (DRVN CRK END) (IN.)
C MUA : BEARING FRICTION COEFFICIENT AT JOINT A
C MUB : BEARING FRICTION COEFFICIENT AT JOINT B
C MUC : BEARING FRICTION COEFFICIENT AT JOINT C
C MUD : BEARING FRICTION COEFFICIENT AT JOINT D
C FGX : FORCE AT FOOT IN X-DIRECTION (RIGHT POS) (IN./SEC)
C FGY : FORCE AT FOOT IN Y-DIRECTION (UP POS) (IN./SEC)
C AX : ACCELERATION OF BODY IN X-DIRECTION (IN./SEC)
C AY : ACCELERATION OF BODY IN Y-DIRECTION (IN./SEC)
C B1 : ANGLE FOR ACCELERATION OF GRAVITY
C B2 : POSITIVE FOR DOWNHILL SLOPE (RAD.)
C C : ACCELERATION OF GRAVITY (396 IN./SEC)
C PLEN : DISTANCE FROM FRAME EDGE TO DRIVEN CRANK SUPPORT (IN.)
C SPACL : DISTANCE BETWEEN LEGS (IN.)
C BWND : WIDTH OF FRAME (IN.)
C BLN : LENGTH OF FRAME (IN.)
C BDW : WEIGHT OF FRAME (LBS.)

PARAMETERS RETURNED

NUM1 : ARRAY SUBSCRIPT FOR STORING DRAWING VERSION OF LEG
BTHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG (RAD.)
BGAMMA : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEG (RAD.)
BPHI : ARRAY OF COUPLER ANGLES FOR DRAWING LEG (RAD.)
BL2 : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEG (IN.)
XM : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE

PARAMETERS INTERNAL

TIME : TIME (SEC.)
IFLAG : FLAG FOR DETERMINING PORTION OF STRIDE
1.. LEG SWINGING SAME DIRECTION AS STRIDE
2.. LEG SWINGING BACK
3.. LEG SWINGING DOWN TO TOEDOWN
THE : ANGLE THETA CORRESPONDING TO TOEDOWN (RAD.)
THETA : " " BRAKING HEIGHT (RAD.)
THETIM : TIME REQUIRED FOR LEG TO SWING TO TOEDOWN (SEC.)
NUMBER : NUMBER OF TIME STEP
TAF : MAGNITUDE OF FORCE AT JOINT A (SEE COMACC)

EXTERNAL REFERENCES

ORDLEO, TOEDWN, PICKOF, ANGLE, LEGPOS, RKSTEP, VELOC, LINFED

CALLING ROUTINES

MAIN
PROGRAM PAGE = 8

C ************************************************************
C
C DIMENSION A(9,9), B(9,1)
C DIMENSION $THETA(B0), $GAMMA(B0), $PHI(B0), BL2(B0), XMASS(B0)

C COMMON /A/ L1, L3, L5, LX, LY, PHIL
C COMMON /B/ NUM1, $THETA, $GAMMA, $PHI, BL2, XMASS
C COMMON /C/ L2, $THETA, $GAMMA, PHI
C COMMON /D/ THD21, QADOT1, PHD21
C COMMON /E/ THD22, QADOT2, PHD22
C COMMON /F/ COSTH, SINH, COSQ, SINGA, COSPH, SINPH
C COMMON /O/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
$ INCRE, INC1, DELTIM, SHORTL, BL2
C COMMON /H/ THETAI, GAMMAI, PHII, LM1, LM2, LM3, JC01, JC02, JC03,
$ M1, M2, M3
C COMMON /AA/ RA, RB, RC, RD, MA, MUB, MUC, MUD, FOX, FOY, AX, AY, BI, O
C COMMON /BB/ PLACLO, SPACLO, BDWIDT, BDLENT, BDWEIT
C COMMON /DD/ TAF, TBF, TCF, TDF
C
C REAL L1, L2, L3, L5, LX, LY, LM1, LM2, LM3, M1, M2, M3, JC01, JC02, JC03,
$ MUA, MUB, MUC, MUD
C
C C

BL2= L2
C CALL GRDLEQ
C
C * LOADS THE POSITION OF THE LEG IN CONTACT WITH THE
C * GROUND INTO THE DRAWING BUFFER.
C CALL TOEDWN(THETOE, THEBRA, THETIM, LINE)
C
C * COMPUTES THE REQUIRED BRAKING HEIGHT OF THE LEG FOR
C * SMOOTH POSITION AND VELOCITY TRANSFER AT TOEDOWN
C TIME1= PERIOD
C CALL PICKOF(TIME1)
C
C * DETERMINES THETA AND THETA DOT AT PICKOFF POSITION
C
C TAF= 0.
C TBF= 0.
C TCF= 0.
C TDF= 0.
C TIME= 0.
C NUM1= 0
C IFLAG= 0
C
C * INITIALIZE VARIABLES TO 0.0 FOR START OF ANALYSIS
C
C NUMBER= 1.0
C 11 NUMBER= NUMBER+1
C IF(NUMBER .GE. INCRE) GO TO 15
C
C * IF NUMBER IS > OR = TO INCRE THE PROGRAM
C * HAS SUCCESSFULLY COMPLETED A LEG CYCLE.

162
C * NUMBER GOES FROM 0 TO INCRE-1 FOR A TOTAL OF
C * INCRE POINTS.
C * THEREFORE WE GO TO THE END OF CALCULATION PHASE.
C
C IF(NUMBER/INC1=INC1-NUMBER)40,41,40
C 41 NUM1=NUMBER/INC1+1
C IF(NUM1.EQ.90. AND. THETA.GT.THETDE) THETA=THETDE
C
C * CORRECT FOR TOEDOWN DUE TO SIZE OF TIME STEP
C
C CALL ANGLE
C CALL VELOC
C CALL LEOPOS(L2,THETA,GAMMA,PHI)
C
C * IF NUMBER IS DIVISIBLE BY INC1 STORE RESULT
C * FOR FUTURE DRAWING.
C
C 40 TIME=DELTIM*NUMBER
C HOLD=THDOT1
C
C * COMPUTE TIME AND SAVE OLD THDOT1 FOR COMPARISON LATER
C
C MUA=ABS(MUA)
C MUB=ABS(MUB)
C MUC=ABS(MUC)
C MUD=ABS(MUD)
C IF(THDOT1.LT.0.) MUA=-MUA
C IF(QADOT1.LT.0.) MUD=-MUD
C IF(THDOT1.LT.PHDOT1) MUB=-MUB
C IF(QADOT1.LT.PHDOT1) MUC=-MUC
C
C * CHECK THE DIRECTION OF FRICTION TORQUE AND
C * ADJUST ACCORDINGLY
C
C CALL RKSTEP(DELTIM,TIME,THETA,THDOT1,LINE)
C
C * ADVANCES THE SOLUTION OF THE EQUATION ONE TIME STEP
C * USING 4TH ORDER RUNGE KUTTA METHODS.
C
C IF(IFLAG.GE.2) GO TO 14
C
C * IF LEG IS SWINGING DOWN TO TOUCH GROUND THEN
C * ALREADY CHANGED DIRECTIONS OF FRICTION TORQUES.
C
C IF(THDOT1*HOLD) 10,11,11
C
C * IF LEG CHANGES DIRECTIONS THEN CHANGE DIRECTIONS
C * OF FRICTION TORQUES ELSE DO ANOTHER STEP.
C
C 10 IFLAG=IFLAG+1
C IF(IFLAG=1) 12,12,13
C
C * IF THIS IS THE FIRST CHANGE OF DIRECTION THEN
C * L2 MUST BE SHORTENED FOR SWINGBACK PHASE.
C * FOR SECOND CHANGE OF DIREC. INCREASE L2.
C
C 12 L2=L2-SHORTL

163
GO TO 11
   * SHORTEN LEG
13  L2=L2+SHORTL
   * LENGTHEN LEG
   IF(THETA-THEBRA) 14, 14, 62
      * CHECK AND SEE IF LEG SWING BACK TO BRAKE HEIGHT
   IF(THETA-THETOE) 18, 18, 16
      * CHECK AND SEE IF LEG SWING BACK TO TOE DOWN HEIGHT
   14 CONTINUE
   IF(IFLAG.EQ.3) GO TO 60
      * LET SWING TO BRAKE HEIGHT BEFORE LOCKING.
      * IF IFLAG=3 THEN WE HAVE REACHED BRAKE HEIGHT
   IF((THETA-THEBRA).OE.0) IFLAG=3
      * SET IFLAG WHEN BRAKE HEIGHT IS REACHED
   IF((TIME+THETIM).GT.PERIOD) GO TO 20
      * CHECK IF TOTAL PERIOD OF SWING BACK IS LESS THAN
      * PERIOD OF STRIDE.
   IF(IFLAG.NE.3) GO TO 11
      * STEP TO THE NEXT POSITION IN TIME
      * ELSE STOP AT BRAKE POSITION
   THEBRA=THETA
   THDOT1=0.0
60  IF((THETIM.LT.DEPTIM).AND.(TIME+THETIM).GE.(PERIOD-DELTIM))
   &.OR.(THETIM.GE.DEPTIM. AND. TIME+THETIM.GE.PERIOD)) GO TO 11
      * IF LEG IS SWINGING DOWN, CONTINUE ANALYSIS
      * IF NOT--BRAKE AT THIS HEIGHT UNTIL TIME TO
      * SWING.
   NUMBER=NUMBER+1
   TIME=NUMBER*DELTIM
   IF(NUMBER/INCI*INCI-NUMBER) 60, 64, 60
64  NUM1=NUMBER/INCI+41
    CALL ANGLE
    CALL LEOPDS(L2, THETA, GAMMA, PHI)
   GO TO 60
      * KEEP LEG BRAKED WHILE INCREASING THE TIME
      * STORE POSITION FOR DRAWING WHEN NUMBER IS
      * DIVISIBLE BY INCI
CALL LINFED(LINE, 1)
WRITE(5, 104)
FORMAT(12X, 'LEG GEOMETRY ACCEPTABLE')
GO TO 19

CALL LINFED(LINE, 1)
WRITE(5, 102)
FORMAT(12X, '!!! LEG DOES NOT RETURN !!!')
GO TO 6

CALL LINFED(LINE, 1)
WRITE(5, 103)
FORMAT(12X, '!!! LEG VELOCITY TO SLOW AT TOUCHDOWN !!!')
GO TO 6

CALL LINFED(LINE, 2)
WRITE(5,105)
FORMAT(12X, '!!! SWING PHASE PERIOD EXCEEDS STRIDE ',
   '12X, 'PERIOD !!!')
CONTINUE

NUMBER=NUMBER+1
IF(NUMBER.GE.INCRE) GO TO 19
IF(NUMBER/INC1+INC1-NUMBER) 72,74,72

TIME=DELTIM*NUMBER
NUM1=NUMBER/INC1+41
CALL ANGLE
CALL LEGPOS(L2, THETA, GAMMA, PHI)
GO TO 72

* IF LEG IS NOT ACCEPTABLE, FILL REMAINING DRAWING
* BUFFER WITH THE FINAL LEG POSITION.

MUA=ABS(MUA)
MUB=ABS(MUB)
MUC=ABS(MUC)
MUD=ABS(MUD)

* RETURN FRICTION COEFFICIENTS TO POSITIVE VALUES
RETURN
END
SUBROUTINE ANGLE

C 3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE CALCULATES THE ANGLES OF A 4-BAR LINKAGE GIVEN
THE ANGULAR POSITION OF THE DRIVE CRANK. A LOOP EQUATION
TECHNIQUE IS USED.

PARAMETERS INPUT

L1 :LENGTH OF DRIVE CRANK (IN.)
L2 :LENGTH OF Driven CRANK (IN.)
L3 :LENGTH OF COUPLER (IN.)
L5 :LENGTH OF COUPLER SHANK (IN.)
LX :X DISTANCE FROM DRIVE CRANK SUPPORT TO
:DRIVEN CRANK SUPPORT --LEFT POS.-- (IN.)
LY :Y DISTANCE FROM DRIVE CRANK SUPPORT TO
:DRIVEN CRANK SUPPORT --UPWARD POS.-- (IN.)
THETA :DRIVE CRANK ANGLE MEASURED FROM POS. X
:HORIZONTAL --CW-- (RAD.)

PARAMETERS RETURNED

GAMMA :DRIVE CRANK ANGLE MEASURED FROM POS. X
:HORIZONTAL --CW-- (RAD.)
PHI :COUPLER ANGLE MEASURED FROM POS. X
:HORIZONTAL --CW-- (RAD.)
COSTH :COSINE OF DRIVE CRANK ANGLE
COSGA :" " DRIVEN CRANK ANGLE
COSPH :" " COUPLER
SINTH :SINE OF DRIVE CRANK ANGLE
SINGA :" " DRIVEN CRANK ANGLE
SINPH :" " COUPLER

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING Routines

ANALYS, CHOEDI, COMACC, ORLEQ1, PICKOF

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /F/ COSTH, SINTH, COSGA, SINGA, COSPH, SINPH
REAL L1, L2, L3, L5, LX, LY

166
DEFINE ANGLES AS THE FOLLOWING

C
COStH = COS(THETA)
SINTh = SIN(THETA)

C
GET ANGLES AS A FUNCTION OF INPUT ANGLE "THETA"

C
G = L3**2 - lx**2 - ly**2 - 11**2 - 2*L2 + 2 - 2*L1*(lx*COStH + ly*SINTh)
B = 4*L2*ly + 4*L1*L2*SINTh
C = G + 2*L2*lx + 2*L1*L2*COStH

C
* COMPUTE PARAMETERS FOR FINDING QUADRATIC
* ROOTS.

C
CHECK = B**2 - 4*A*C
IF (CHECK .LE. 0.) GO TO 10

C
* CHECK TO SEE IF LINKAGE CAN BE ASSEMBLED

C
GAMMA = 2*ATAN((-B-SQRT(B**2 - 4*A*C))/2/A)
COsGAMMA = COS(GAMMA)
SINgamma = SIN(GAMMA)
PHI = 2*ATAN((LY + L1*SINTh - L2*SINgamma)/(L3 + LX + L1*COStH - L2*COsGAMMA))
COsPHI = COS(PHI)
SINPHI = SIN(PHI)
RETURN

C
* RETURN IF NO ERROR DETECTED

C
10 THtemp = THETA**180. / 4. / ATAN(1.)
GAMMA = 1
PHI = 1
WRITE (5, 100) THtemp
100 FORMAT (12X, 'LINKAGE LIMITS EXCEEDED', ',',
* 12X, 'THETA = ', F10.2, ',',
* 12X, 'GAMMA AND PHI SET TO 1', ',',
* 12X, 'PROGRAM EXECUTION CONT. ')
RETURN

C
* PRINT ERROR MESSAGE IF LINKAGE LIMITS EXCEEDED

C
END
SUBROUTINE ANGLE1(ANG, X1, Y1, X2, Y2)

DESCRIPTION
   This routine determines the value of an angle in radians
   relative to the horizontal and referenced to the line
   defined by the two parameter coordinates.

ACCESS
   call angle(ANG, X1, Y1, X2, Y2)

PARAMETERS INPUT
   X1   -X coord. where angle is measured
   Y1   -Y coord. where angle is measured
   X2   -X coord. of pt on reference line
   Y2   -Y coord. of pt on reference line

PARAMETERS RETURNED
   ANG   -measured angle(0. to 2*pi) radians.

ERROR CONDITIONS
   none

EXTERNAL REFERENCES
   SIN, COS, ATAN

CALLING ROUTINES
   CHFAR2, CHFAR5

PI=4.*ATAN(1.)
TDL=0.000001
length of the reference line.
XLEN=SQRT((X2-X1)**2+(Y2-Y1)**2)**.5
ANG=0.0
IF(ABS(XLEN).LE.TDL)RETURN

cosine of the angle.
COSA=(X2-X1)/XLEN

sine of the angle.
SINA=(Y2-Y1)/XLEN

1 tangent of the half angle determines ANG.
ANG=2.*ATAN2(SINA, (COSA+1))

IF(ANG.LT.0)ANG=ANG+2.*PI
RETURN
END
SUBROUTINE ARROW(ICOOR, X, Y, RLENAR, THETA)

AUTHOR: LAWRENCE OUTKOWSKI

DESCRIPTION

THIS ROUTINE DRAWS AN ARROW HEAD AT ANY ORIENTATION AND OF ANY LENGTH

ICOOR = 0 IF POINT OF ARROW IS DEFINED RELATIVE TO GLOBAL SYSTEM = 1 IF POINT OF ARROW IS DEFINED RELATIVE TO ANOTHER POINT

X = X POSITION OF ARROW POINT

Y = Y POSITION OF ARROW POINT

RLENAR = LENGTH OF ARROW IN X, Y UNITS

THETA = ORIENTATION OF ARROW HEAD (DEG); 0 WILL MAKE ARROW POINT TO THE LEFT, THETA INCREASES COUNTERCLOCKWISE

COMPUTE ANGLES BETWEEN BARBS AND X-AXIS (RAD)

BANG1 = 0.01745*(THETA-10.)
BANG2 = 0.01745*(THETA+10.)

COMPUTE COORDINATES OF FIRST BARB RELATIVE TO POINT

XBRB1 = RLENAR*COS(BANG1)
YBRB1 = RLENAR*SIN(BANG1)

COMPUTE COORDINATES OF SECOND BARB RELATIVE TO POINT

XBRB2 = RLENAR*COS(BANG2)
YBRB2 = RLENAR*SIN(BANG2)

DETERMINE WHETHER GLOBAL OR RELATIVE POINT COORDINATE

IF(ICOOR.EQ.0) GO TO 40
CALL MOVER(X, Y)
GO TO 50
40 CALL MOVEA(X, Y)

MOVE TO END POINT OF BARB 1

CALL MOVER(XBRB1, YBRB1)
CALL DRAWR(-XBRB1, -YBRB1)
CALL DRAWR(XBRB2, YBRB2)

RETURN CURSOR TO POINT OF ARROW (FOR FURTHER RELATIVE MOVES)

CALL MOVER(-XBRB2, -YBRB2)
SUBROUTINE BUSH (R, XC, YC, THETA)

C******************************************************************************************
C 3-MAY-82  AUTHOR: GARY L. KINZEL
C DESCRIPTION
C THIS ROUTINE DRAWS A BUSHING FOR A DRAW A BEARING BUSHING
C AT LOCATION XC, YC ON VIRTUAL COORDINATE GRID
C PARAMETERS INPUT
C R : RADIUS OF BUSHING IN VIRTUAL SCREEN UNITS
C XC : X LOCATION OF CENTER OF BUSHING IN SCREEN UNITS
C YC : Y LOCATION OF CENTER OF BUSHING IN SCREEN UNITS
C THETA : ANGLE BUSHING IS ORIENTED (DEG.)
C
C EXTERNAL REFERENCES
C CIRCLE
C CALLING ROUTINES
C CHGEO2
C******************************************************************************************
C CALL THE CIRCLE ROUTINE TO DRAW THE INNER RADIUS
C OF THE BUSHING.
C CALL CIRCLE (XC, YC, 0.0, 360., R, 0)
C DEFINE VARIOUS DIMENSIONS OF THE BUSHING.
C
T1=THETA
T2=T1+180.
RR=2.*R
RB=3.0*R
C
C CONVERSION FACTOR FROM DEGREES TO RADIANS.
C
RAD=ATAN(1.0)/45.
T=THETA*RAD
C
DRAW BUSHING HOUSING WITH STRAIGHT LINES AND
A SEMICIRCLE FROM ROUTINE CIRCLE.
C
S=SIN(T)
C=COS(T)
X=XC+RB*S+RR*C
Y=YC+RR*S-RB*C
CALL MOVEA (X, Y)
X=-RB*S
Y=RB*C
CALL DRAWR (X, Y)
CALL CIRCLE (XC, YC, T1, T2, RR, O)
X=RR*S
Y=RR*C
CALL DRAWR (X, Y)

DRAW BASE OF BUSHING.

X=RR*C
Y=RR*S
CALL MOVER (X, Y)
X=6.*RR*C
Y=6.*RR*S
CALL DRAWR (X, Y)

DRAW HASH MARKS

P=RAD+.45.
P=P+T
X=-.5*RR*C
Y=-.5*RR*S
CALL MOVER (X, Y)
CP=COS(P)
SP=GIN(P)
N=1
1 X=-.7*RR*CP
Y=-.7*RR*SP
CALL DRAWR (X, Y)
X=.7*RR*CP-RR*C
Y=.7*RR*SP-RR*S
CALL MOVER (X, Y)
N=N+1
IF(N.NE.7) GO TO 1
CALL TSND
CALL ANMODE
RETURN

END
SUBROUTINE CANIMA(XM, XR)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE CONTROLS THE ANIMATION OF THE LEGO. IT LOADS
THE ANIMATION DEVICE AND SETS THE CORRECT PARAMETERS

PARAMETERS INPUT

PLACLO : DISTANCE FROM FRAME EDGE TO DRIVEN CRANK SUPPORT (IN.)
SPACLO : DISTANCE BETWEEN LEGS (IN.)
XM : VIRTUAL COORDINATE (MIN.) INPUT TO DRWLEG
XR : VIRTUAL COORDINATE RANGE INPUT TO DRWLEG

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

DX : DISTANCE FROM FRAME EDGE TO DRIVEN CRANK SUP. (IN.)
DX1 : DISTANCE FROM FRAME EDGE TO 2ND DRIVEN CRANK SUP. (IN.)
DX2 : DISTANCE FROM FRAME EDGE TO 3RD DRIVEN CRANK SUP. (IN.)
IR : ASCII CODE FOR INPUT OF NUMBERS
I1 : DO-LOOP COUNTER (INPUT TO DRWLEG)
I2 : " "
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

EXTERNAL REFERENCES

DRWFRM, DRWLEG

CALLING ROUTINES

DRWTYP

COMMON /BB/ PLACLO, SPACLO, BDWIDT, BDLENT, BDWEIT

CALL ANIMARM
CALL KEYFUN(1)
CALL KEYSTP(0)
CALL DELAY(0)
CALL NUMREP(2)

* SET UP ANIMATION BUFFER

DX=PLACLO
DX1=SPACLO+PLACLO

173
PROGRAM PAGE = 20

DX2=2*SFACLO+PLACL0

* PARAMETERS REQUIRED BY DRAWEO ROUTINE

CALL DRWFRM

* DRAWS THE WINDOW FRAME AND GROUND LINES

LINE=1
CALL LINFED(LINE, 6)
WRITE(5, 100)

100 FORMAT(13X, 'PLEASE WAIT. ',//,
 $ 13X, 'BUFFER BEING FILLED. ',//,
 $ 13X, 'BELL WILL SOUND WHEN READY')

* WRITE MESSAGE TO SCREEN - BUFFER BEING FILLED.

CALL CZAXIS(2)
DO 11 II=1, 2
   DO 11 I2=1, 40
      CALL FRAME(1, 1)
      CALL DRWLEQ(I1, I2, DX, DX1, DX2, 0, XM, XR, XM, XR, 1373, 310, 2500)
11 CONTINUE

CALL STPFILE(1)

* FILLS THE ANIMATION BUFFER

CALL BELL
CALL BELL
CALL LINFED(LINE, 3)
WRITE(5, 101)

101 FORMAT(13X, 'ANIMATION IN PROGRESS. ',//,
 $ 13X, 'HIT "CR" TO STOP. ',//,
 $ 13X, 'ENTER NO. 0 THRU 9',//,
 $ 13X, 'TO CHANGE ANIMATION SPEED. ')

* STOP ANIMATION BY ENTERING "CR".

CALL CZAXIS(2)
CALL ANIMAT(1, 0)

21 CALL TINPUT(IR)
IF(IR .LE. 57 .AND. IR .GE. 48) CALL NUMREP(IR-47)
IF(IR .NE. 32) GO TO 21
CALL STPANM

CALL CZAXIS(0)
CALL HOME
RETURN
END
SUBROUTINE CHBEAR

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE ALLOWS INTERACTIVE INPUT OF BEARING PARAMETERS
"MU" AND "R" OF THE LINKAGE JOINTS

PARAMETERS INPUT

RA : BEARING RADIUS AT JOINT A (DRV CRK SUP) (IN.)
RB : BEARING RADIUS AT JOINT B (DRV CRK END) (IN.)
RC : BEARING RADIUS AT JOINT C (DRVN CRK SUP) (IN.)
RD : BEARING RADIUS AT JOINT D (DRVN CRK END) (IN.)
MUA : BEARING FRICTION COEFFICIENT AT JOINT A
MUB : BEARING FRICTION COEFFICIENT AT JOINT B
MUC : BEARING FRICTION COEFFICIENT AT JOINT C
MUD : BEARING FRICTION COEFFICIENT AT JOINT D

PARAMETERS RETURNED

RA, RB, RC, RD, MUA, MUB, MUC, MUD

PARAMETERS INTERNAL

LINE : LINE SPACING PARAMETER FOR SCREEN WRITES
DAT : TEMPORARY ARRAY STORAGE FOR INPUT PARAMETERS
KK : ARRAY SUBSCRIPT FOR DAT

EXTERNAL REFERENCES

LINRED, CHOEO1, CHOEO2, DRWFRM

CALLING ROUTINES

MAIN

DIMENSION DAT(B)

COMMON /AA/ RA, RB, RC, RD, MUA, MUB, MUC, MUD, FOX, FOY, AX, AY, SI, G

REAL MUA, MUB, MUC, MUD

PI=4.0*ATAN(1.0)
CALL DRWFRM(IPAGE)

CALL CHOEO1(1373, 310, 2500, 20.)

* DRAWS THE LATEST VERSION OF LEG ON SCREEN

CALL CHOEO2(1)

175
* LABELS THE INPUT PARAMETERS TO BE INPUT

LINE=1
CALL LINFED(LINE, 4)
WRITE(5, 100)

* DESCRIBES PROGRAM INPUTS

CALL LINFED(LINE, 9)
WRITE(5, 111) MUA
WRITE(5, 112) MUB
WRITE(5, 113) MUC
WRITE(5, 114) MUD
WRITE(5, 115) RA
WRITE(5, 116) RB
WRITE(5, 117) RC
WRITE(5, 118) RD

* PRINT CURRENT VALUES

301 CALL LINFED(LINE, 2)
WRITE(5, 101)
READ(5, *, ERR=301) DAT(1)
IF(DAT(1).EQ.0) DAT(1)=MUA

* READ IN VALUE OF MUA

302 CALL LINFED(LINE, 1)
WRITE(5, 102)
READ(5, *, ERR=302) DAT(2)
IF(DAT(2).EQ.0) DAT(2)=MUB

* READ IN VALUE OF MUB

303 CALL LINFED(LINE, 1)
WRITE(5, 103)
READ(5, *, ERR=303) DAT(3)
IF(DAT(3).EQ.0) DAT(3)=MUC

* READ IN VALUE OF MUC

304 CALL LINFED(LINE, 1)
WRITE(5, 104)
READ(5, *, ERR=304) DAT(4)
IF(DAT(4).EQ.0) DAT(4)=MUD

* READ IN VALUE OF MUD

305 CALL LINFED(LINE, 1)
WRITE(5, 105)
READ(5, *, ERR=305) DAT(5)
IF(DAT(5).EQ.0) DAT(5)=RA

* READ IN VALUE OF RA

306 CALL LINFED(LINE, 1)
WRITE(5, 106)
PROGRAM PAGE = 23

READ(5,*,ERR=306) DAT(6)
IF(DAT(6).EQ.0) DAT(6)=RB

* READ IN VALUE OF RB

307 CALL LINFED(LINE,1)
WRITE(5,107)
READ(5,*,ERR=307) DAT(7)
IF(DAT(7).EQ.0) DAT(7)=RC

* READ IN VALUE OF RC

308 CALL LINFED(LINE,1)
WRITE(5,108)
READ(5,*,ERR=308) DAT(8)
IF(DAT(8).EQ.0) DAT(8)=RD

* READ IN VALUE OF RD

100 FORMAT(12X,'DEFINE NEW BEARING PARAM. ',/,
* 12X,'INPUT OF "0"',/,
* 12X,'DEFAULTS TO OLD VALUE. ')
101 FORMAT(12X,'INPUT NEW COEFFICIENT MUA ',*)
102 FORMAT(12X,'INPUT NEW COEFFICIENT MUB ',*)
103 FORMAT(12X,'INPUT NEW COEFFICIENT MUC ',*)
104 FORMAT(12X,'INPUT NEW COEFFICIENT MUD ',*)
105 FORMAT(12X,'INPUT NEW PIN RADIUS RA ',*)
106 FORMAT(12X,'INPUT NEW PIN RADIUS RB ',*)
107 FORMAT(12X,'INPUT NEW PIN RADIUS RC ',*)
108 FORMAT(12X,'INPUT NEW PIN RADIUS RD ',*)

MUA=DAT(1)
MUB=DAT(2)
MUC=DAT(3)
MUD=DAT(4)
RA=DAT(5)
RB=DAT(6)
RC=DAT(7)
RD=DAT(8)

* CONVERTS DAT ARRAY TO ACTUAL PARAMETERS

CALL DRWFRM
CALL CHQED1(1373,310,2500,20.)
CALL CHQED2(1)

* REDRAW NEW VERSION ON SCREEN

LINE=1
CALL LINFED(LINE,3)
WRITE(5,110)
CALL LINFED(LINE,1)
WRITE(5,111) MUA
CALL LINFED(LINE,1)
WRITE(5,112) MUB
CALL LINFED(LINE,1)
WRITE(5,113) MUC
CALL LINFED(LINE,1)
WRITE(5,114) MUD
CALL LINFED(LINE,1)
WRITE(5,115) RA
CALL LINFED(LINE,1)
WRITE(5,116) RB
CALL LINFED(LINE,1)
WRITE(5,117) RC
CALL LINFED(LINE,3)
WRITE(5,118) RD

* ECHOS THE INPUT DATA

110 FORMAT(12X,'NEW BEARING PARAM. ')
111 FORMAT(12X,'(1) COEFFICIENT MUA = ',FB,3)
112 FORMAT(12X,'(2) COEFFICIENT MUB = ',FB,3)
113 FORMAT(12X,'(3) COEFFICIENT MUC = ',FB,3)
114 FORMAT(12X,'(4) COEFFICIENT MUD = ',FB,3)
115 FORMAT(12X,'(5) PIN RADIUS RA = ',FB,3)
116 FORMAT(12X,'(6) PIN RADIUS RB = ',FB,3)
117 FORMAT(12X,'(7) PIN RADIUS RC = ',FB,3)
118 FORMAT(12X,'(8) PIN RADIUS RD = ',FB,3)

309 CALL LINFED(LINE,2)
WRITE(5,120)
120 FORMAT(12X,'ARE VALUES CORRECT [Y,N]? ',*)
READ(5,200,ERR=309) ANS
200 FORMAT(1A1)
IF(ANS.EQ.'Y') RETURN

* IF INPUT DATA IS CORRECT RETURN—ELSE CHANGE

* THE INCORRECT VALUES.

310 CALL LINFED(LINE,1)
10 WRITE(5,121)
121 FORMAT(12X,'INPUT LINE NO., NEW VALUE ',*)
READ(5,*.ERR=310) KK,DAT(KK)

* CHANGE AN INCORRECT VALUE

311 CALL LINFED(LINE,1)
WRITE(5,122)
122 FORMAT(12X,'ANOTHER CHANGE [Y,N]? ',*)
READ(5,200,ERR=311) ANS
IF(ANS.EQ.'Y') 00 TO 10
00 TO 20

* RE-ECHOS INPUT AFTER CHANGES ARE COMPLETE

END
SUBROUTINE CHGEO2M

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE IS USED TO CHANGE THE LEG GEOMETRY DURING
INTERACTIVE INPUT

PARAMETERS INPUT

L1 : LENGTH OF DRIVE CRANK  (IN.)
L2 : LENGTH OF DRIVEN CRANK  (IN.)
L3 : LENGTH OF COUPLER  (IN.)
L5 : LENGTH OF COUPLER SHANK  (IN.)
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO
: DRIVEN CRANK SUPPORT --LEFT POS.--  (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO
: DRIVEN CRANK SUPPORT --UPWARD POS.--  (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK  (RAD.)

PARAMETERS RETURNED

L1, L2, L3, L5, LX, LY, PHIL

PARAMETERS INTERNAL

LINE : LINE SPACING PARAMETER FOR SCREEN WRITES
DAT : TEMPORARY STORAGE FOR INPUT PARAMETERS
PI : PI

EXTERNAL REFERENCES

LINFED, CHGEO1, CHGEO2, DRAWFRM

CALLING ROUTINES

MAIN

DIMENSION DAT(10)

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /C/ L2, THETA, GAMMA, PHI

REAL L1, L2, L3, L5, LX, LY

PI=4.*ATAN(1.0)
CALL DRAWFRM(IPAGE)
CALL CHGEO1(1373, 310, 2500, 20.)

* DRAWS THE LATEST VERSION OF LEG ON SCREEN

179
CALL CH1EQ002(0)

* LABELS THE INPUT PARAMETERS TO BE INPUT

LINE=1
CALL LIN1EQD(LINE, 4)
WRITE(5, 100)

* DESCRIBES PROGRAM INPUTS

CALL LIN1EQD(LINE, 8)
WRITE(5, 111) L1
WRITE(5, 112) L2
WRITE(5, 113) L3
WRITE(5, 114) L5
PHOUT=PHI*180./PI
WRITE(5, 115) PHOUT
WRITE(5, 116) LX
WRITE(5, 117) LY

* PRINT CURRENT VALUES

301 CALL LIN1EQD(LINE, 2)
WRITE(5, 101)
READ(5, *, ERR=301) DAT(1)
IF(DAT(1).EQ.0) DAT(1)=L1

* READ IN VALUE OF L1

302 CALL LIN1EQD(LINE, 1)
WRITE(5, 102)
READ(5, *, ERR=302) DAT(2)
IF(DAT(2).EQ.0) DAT(2)=L2

* READ IN VALUE OF L2

303 CALL LIN1EQD(LINE, 1)
WRITE(5, 103)
READ(5, *, ERR=303) DAT(3)
IF(DAT(3).EQ.0) DAT(3)=L3

* READ IN VALUE OF L3

304 CALL LIN1EQD(LINE, 1)
WRITE(5, 104)
READ(5, *, ERR=304) DAT(4)
IF(DAT(4).EQ.0) DAT(4)=L5

* READ IN VALUE OF L5

305 CALL LIN1EQD(LINE, 1)
WRITE(5, 105)
READ(5, *, ERR=305) DAT(5)
IF(DAT(5).EQ.0) DAT(5)=PHI*180./PI

* READ IN VALUE OF PHI

180
306 CALL LINFD(LINE, 1)
WRITE(5, 106)
READ(5, *, ERR=306) DAT(6)
IF(DAT(6).EQ.0) DAT(6)=LX
* READ IN VALUE OF LX
307 CALL LINFD(LINE, 1)
WRITE(5, 107)
READ(5, *, ERR=307) DAT(7)
IF(DAT(7).EQ.0) DAT(7)=LY
* READ IN VALUE OF LY
100 FORMAT(12X, 'DEFINE NEW LEG GEOMETRY', /,
* 12X, 'INPUT OF "O",', /
 12X, 'DEFAULTS TO OLD VALUE.', )
101 FORMAT(12X, 'INPUT NEW LINK LENGTH AB', *)
102 FORMAT(12X, 'INPUT NEW LINK LENGTH CD', *)
103 FORMAT(12X, 'INPUT NEW LINK LENGTH BC', *)
104 FORMAT(12X, 'INPUT NEW LEG LENGTH CE', *)
105 FORMAT(12X, 'INPUT NEW ANGLE (DEG) BCE', *)
106 FORMAT(12X, 'INPUT NEW LENGTH ---> (+) DF', *)
107 FORMAT(12X, 'INPUT NEW LENGTH ---> (+) AF', *)
20 L1=DAT(1)
L2=DAT(2)
L3=DAT(3)
L5=DAT(4)
PHIL=DAT(5)/180.*PI
LX=DAT(6)
LY=DAT(7)
* CONVERTS DAT ARRAY TO ACTUAL PARAMETERS
CALL DRWFRM
CALL CHQED1(1373, 310, 2500, 20, )
CALL CHQED2(0)
* REDRAW NEW VERSION ON SCREEN
LINE=1
CALL LINFD(LINE, 3)
WRITE(5, 110)
CALL LINFD(LINE, 1)
WRITE(5, 111) L1
CALL LINFD(LINE, 1)
WRITE(5, 112) L2
CALL LINFD(LINE, 1)
WRITE(5, 113) L3
CALL LINFD(LINE, 1)
WRITE(5, 114) L5
CALL LINFD(LINE, 1)
WRITE(5, 115) DAT(5)
CALL LINFED(LINE,1)
WRITE(5,116) LX
CALL LINFED(LINE,3)
WRITE(5,117) Ly

* ECHDS THE INPUT DATA

110 FORMAT(12X,'NEW LEG GEOMETRY')
111 FORMAT(12X,'(1) LINK LENGTH AB = ',F9.3)
112 FORMAT(12X,'(2) LINK LENGTH CD = ',F9.3)
113 FORMAT(12X,'(3) LINK LENGTH BC = ',F9.3)
114 FORMAT(12X,'(4) LEG LENGTH CE = ',F9.3)
115 FORMAT(12X,'(5) LEG ANG(DEG) BCE = ',F9.3)
116 FORMAT(12X,'(6) LENGTH DF = ',F9.3)
117 FORMAT(12X,'(7) LENGTH AF = ',F9.3)

308 CALL LINFED(LINE,2)
WRITE(5,120)
120 FORMAT(12X,'ARE VALUES CORRECT [Y,N]?',*)
READ(5,200,ERR=309) ANS
200 FORMAT(1A1)
IF(ANS.EQ.'Y') RETURN

* IF INPUT DATA IS CORRECT RETURN—ELSE CHANGE
* THE INCORRECT VALUES.

309 CALL LINFED(LINE,1)
10 WRITE(5,121)
121 FORMAT(12X,'INPUT LINE NO., NEW VALUE ',*)
READ(5,4,ERR=309) KK, DAT(KK)

* CHANGE AN INCORRECT VALUE

310 CALL LINFED(LINE,1)
WRITE(5,122)
122 FORMAT(12X,'ANOTHER CHANGE [Y,N]? ',*)
READ(5,200,ERR=310) ANS
IF(ANS.EQ.'Y') GO TO 10
GO TO 20

* RE-ECHO INPUT AFTER CHANGES ARE COMPLETE

END
SUBROUTINE CHGED1(IXM, IYM, IR, THETDR)

J-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DRAWS THE LATEST VERSION OF LEO

PARAMETERS INPUT

IXM : MIN. X-SCREEN COORDINATE
IYM : " Y " = " "
IR : RANGE OF SCREEN COORDINATE
THETDR : DRIVE CRANK ANGLE FOR DRAWING LEO (IN.)
L1 : LENGTH OF DRIVE CRANK (IN.)
L2 : LENGTH OF DRIVEN CRANK (IN.)
L3 : LENGTH OF COUPLER (IN.)
L5 : LENGTH OF COUPLER SHANK (IN.)
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT --LEFT POS.-- (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT --UPWARD POS.-- (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
THETA : DRIVE CRANK ANGLE MEASURED FROM POS. X HORIZONTAL --CW-- (RAD.)
GAMMA : DRIVEN CRANK ANGLE MEASURED FROM POS. X HORIZONTAL --CW-- (RAD.)
PHI : COUPLER ANGLE MEASURED FROM POS. X HORIZONTAL --CW-- (RAD.)
NUM1 : ARRAY SUBSCRIPT FOR STORING DRAWING VERSION OF LEO
BTHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEO (RAD.)
BQGAMMA : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEO (RAD.)
BPphi : ARRAY OF COUPLER ANGLES FOR DRAWN LEO (RAD.)
BL2 : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEO (IN.)

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

DY : Y LOCATION OF DRIVEN CRANK SUPPORT
XR : X RANGE OF VIRTUAL COORDINATES
YR : Y RANGE OF VIRTUAL COORDINATES
XM : MIN X VIRTUAL COORDINATE
YM : MIN Y VIRTUAL COORDINATE
X : X LOCATION OF DRIVEN CRANK SUPPORT
X1 : X LOCATION OF 2ND DRIVEN CRANK SUPPORT
X2 : X LOCATION OF 3RD DRIVEN CRANK SUPPORT
IFR : FLAG DESIGNATING ONLY ONE LEO TO BE DRAWN
R : RADIUS OF BEARING BUSHING

EXTERNAL REFERENCES

ANGLE, DRWLEG, BUSH

183
CALLING Routines

CHQEDM, CHBEAR, CHPAR2, CHSTRD, CHSTR1

******************************************************************************

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /B/ NUM1, BTHETA(B0), B GAMMA(B0), BPHI(B0), BL2(B0),
* XMASS(B0)
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /CC/ AX, AY, BX, BY, CX, CY, DX, DY, EX, EY

REAL L1, L2, L3, L5, LX, LY

PI=4.*ATAN(1.0)
THETA=THETDR/180.*PI

* DRAW THE LEG WITH INPUT ANGLE = THETDR DEG.

CALL ANGLE

* DETERMINE THE OTHER ANGLES OF THE LEG

BTHETA(1)=THETA
B GAMMA(1)=GAMMA
BPHI(1)=PHI
BL2(1)=L2

* SET UP ANGLES SO THAT ROUTINE DRWLEG CAN BE USED

DY=L2*SIN(GAMMA)+L5*SIN(PHI+PHIL)
XR=2.0*DY
XM=0.0
YM=-DY/2.0
VR=XR

* SCALES THE SCREEN TO FIT THE LEG

I1=1
I2=1
X=2./3.*DY
X1=X
X2=X
IFR=5

* DEFINE PARAMETERS USED IN ROUTINE DRWLEG SO IT
* WILL PRODUCE THE DESIRES RESULT.

CALL DRWLEG(I1, I2, X, X1, X2, IFR, XM, XR, YM, YR, IXM, IYM, IR)

* DRAW THE LEG

R=0.006*XR
CALL BUSH(R, DX, DY, 180.)
CALL BUSH(R, AX, AY, 180.)
PROGRAM PAGE = 31

C
C * DRAWS BUSHINGS AT THE FIXED MOUNTS
C
RETURN
END
SUBROUTINE CHQEO2(I)

PARAMETERS INPUT

I : CONTROL FLAG TO SUPPRESS SPECIFIC LABELS
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO
 : DRIVEN CRANK SUPPORT ---LEFT POS.--- (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO
 : DRIVEN CRANK SUPPORT ---UPWARD POS.--- (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
AX : X LOCATION OF DRIVE CRANK SUPPORT (VIRTUAL CORD)
AY : Y
BX : X LOCATION OF DRIVE CRANK END (VIRTUAL CORD)
BY : Y
CX : X LOCATION OF DRIVEN CRANK END (VIRTUAL CORD)
CY : Y
DX : X LOCATION OF DRIVEN CRANK SUPPORT (VIRTUAL CORD)
DY : Y
EX : X LOCATION OF FOOT POINT (VIRTUAL CORD)
EY : Y

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

FX : X LOCATION OF SUPPORT HORIZONTAL/
 : VERTICAL CROSSING (VIRTUAL CORD)
FY : Y

EXTERNAL REFERENCES

PLOTIO Routines

CALLING Routines

CHQEO1, CHBEAR, CHBTR1

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /CC/ AX, AY, BX, BY, CX, CY, DX, DY, EX, EY

REAL L1, L3, L5, LX, LY

CALL CHRSIZ(4)
CALL CSIZE(IH, IV)
CALL MOVEA(EX, EY)
CALL MOVREL(-IH/3,-IV/4)
CALL ANCHO(111)

* OUTPUT "o" AT POINT "E"

CALL MOVEA(BX, BY)
CALL MOVREL(-IH/3,-IV/4)
CALL ANCHO(111)

* OUTPUT "o" AT POINT "B"

CALL MOVEA(CX, CY)
CALL MOVREL(-IH/3,-IV/4)
CALL ANCHO(111)

* OUTPUT "o" AT POINT "C"

CALL CHRSIZ(2)
CALL CSIZE(IH, IV)
CALL MOVEA(DX, DY)
CALL MOVREL(-2*IH, O)
CALL ANCHO(68)

* LABEL POINT "D"

CALL MOVEA(AX, AY)
CALL MOVREL(2*IH, O)
CALL ANCHO(65)

* LABEL POINT "A"

CALL MOVEA(BX, BY)
CALL MOVREL(2*IH, O)
CALL ANCHO(66)

* LABEL POINT "B"

CALL MOVEA(CX, CY)
CALL MOVREL(-2*IH, -IV)
CALL ANCHO(67)

* LABEL POINT "C"

CALL MOVEA(EX, EY)
CALL MOVREL(2*IH, O)
CALL ANCHO(69)

* LABEL POINT "E"

AA=DX
BB=EY
CALL MOVEA(AA, BB)
CALL LINEF
CALL LINEF
CALL ANMODE
WRITE(5, 100)
100 FORMAT(1H+’LATEST VERSION’)
* LABEL THE DRAWING

IF(I.EQ.1) RETURN

* RETURN WITHOUT LABELLING POINT F

FX=DX+LX
FY=AY+LY

CALL MOVEA(DX, DY)
CALL DASHA(FX, FY, 56)
CALL DASHA(AX, AY, 56)
    CALL ARROW(0, FX, FY, 20, 180.)
    CALL ARROW(0, FX, FY, 20, 270.)

* DRAWS DASH LINE INDICATING LX, LY

CALL MOVEA(FX, FY)
CALL MOVREL(IH, IV)
CALL ANCHO(70)

* LABELS POINT "F"

CALL CHRSIZ(4)
CALL CSIZE(IH, IV)
CALL MOVEA(FX, FY)
CALL MOVREL(-IH/3, -IV/4)
CALL ANCHO(111)

* PUTS "e" AT POINT "F"

RETURN
END
SUBROUTINE CHFRAM
C
C******************************************************************************
C
3-MAY-82  AUTHOR: FRANK BROWN
C
DESCRIPTION
C
THIS ROUTINE IS USED TO INTERACTIVELY CHANGE FRAME PARAMETERS
C
PARAMETERS INPUT
C
   PLACLG  :DISTANCE FROM FRAME EDGE TO DRIVEN
   :CRANK SUPPORT           (IN. )
   SPACLG  :DISTANCE BETWEEN LEGS         (IN. )
   BDWIDT  :WIDTH OF FRAME          (IN. )
   BDLENT  :LENGTH OF FRAME        (IN. )
   BDWEIT  :WEIGHT OF FRAME         (LBS. )
C
PARAMETERS RETURNED
C
   PLACLG, SPACLG, BDWIDT, BDLENT, BDWEIT
C
PARAMETERS INTERNAL
C
   DAT    :TEMPORARY STORAGE FOR INPUT PARAMETERS
   WK     :ARRAY SUBSCRIPT FOR DAT
   LINE   :LINE SPACING PARAMETER FOR SCREEN WRITES
C
EXTERNAL REFERENCES
C
   DRWFRM, LINFD, DUMODL
C
CALLING ROUTINES
C
   MAIN
C
C******************************************************************************
C
DIMENSION DAT(10)
C
COMMON /BB/ PLACLG, SPACLG, BDWIDT, BDLENT, BDWEIT
C
CALL DRWFRM
IFK=1
   CALL DUMODL(IFK)
C
LINE=1
CALL LINFD(LINE,4)
WRITE(5,100)
C
   * DESCRIBES PROGRAM INPUTS
C
   CALL LINFD(LINE,8)
   WRITE(5,111) PLACLG
   WRITE(5,112) SPACLG
   WRITE(5,113) BDWIDT

189
WRITE(5,114) BDLNT
WRITE(5,115) BDWEIT
C
C     * PRINT CURRENT VALUES
C
301 CALL LINFED(LINE,2)
WRITE(5,101)
READ(5,*,ERR=301) DAT(1)
IF(DAT(1).EQ.0) DAT(1)=PLACLO
C
C     * READ IN VALUE OF PLACLO
C
302 CALL LINFED(LINE,1)
WRITE(5,102)
READ(5,*,ERR=302) DAT(2)
IF(DAT(2).EQ.0) DAT(2)=SPACLO
C
C     * READ IN VALUE OF SPACLO
C
303 CALL LINFED(LINE,1)
WRITE(5,103)
READ(5,*,ERR=303) DAT(3)
IF(DAT(3).EQ.0) DAT(3)=BDWIDT
C
C     * READ IN VALUE OF BDWIDT
C
304 CALL LINFED(LINE,1)
WRITE(5,104)
READ(5,*,ERR=304) DAT(4)
IF(DAT(4).EQ.0) DAT(4)=BDLEN
C
C     * READ IN VALUE OF BDLEN
C
305 CALL LINFED(LINE,1)
WRITE(5,105)
READ(5,*,ERR=305) DAT(5)
IF(DAT(5).EQ.0) DAT(5)=BDWEIT
C
C     * READ IN VALUE OF BDWEIT
C
C
100 FORMAT(12X,'DEFINE NEW FRAME PARAM. ',/,
  12X,'INPUT OF "O"',/,
  12X,'DEFAULTS TO OLD VALUE.')
FH=102 FORMAT(12X,'INPUT NEW LEG PLACEMENT ',*)
103 FORMAT(12X,'INPUT NEW LEG SPACING ',*)
104 FORMAT(12X,'INPUT NEW FRAME WIDTH ',*)
105 FORMAT(12X,'INPUT NEW FRAME LENGTH ',*)

20 PLACLO=DAT(1)
SPACLO=DAT(2)
BDWIDT=DAT(3)
BDLEN=DAT(4)
BDWEIT=DAT(5)
* CONVERTS DAT ARRAY TO ACTUAL PARAMETERS

```
CALL DRWFRM
```

```
LINE=1
CALL LINFD(LINE,0)
WRITE(5,110)
WRITE(5,111) PLACLQ
WRITE(5,112) SPACLQ
WRITE(5,113) BDWDIT
WRITE(5,114) BDLENT
WRITE(5,115) BDWEIT
```

* ECHOES THE INPUT DATA

```
110 FORMAT(12X,'NEW FRAME PARAMETERS')
111 FORMAT(12X,'(1) LEG PLACEMENT = ',F9.3)
112 FORMAT(12X,'(2) LEG SPACING = ',F9.3)
113 FORMAT(12X,'(3) FRAME WIDTH = ',F9.3)
114 FORMAT(12X,'(4) FRAME BDLENT = ',F9.3)
115 FORMAT(12X,'(5) FRAME WEIGHT = ',F9.3)
```

```
306 CALL LINFD(LINE,2)
WRITE(5,120)
120 FORMAT(12X,'ARE VALUES CORRECT [Y,N]? ',*)
READ(5,200,ERR=306) ANS
200 FORMAT(1A1)
IF(ANS.EQ.'Y') RETURN
```

* IF INPUT DATA IS CORRECT RETURN—ELSE CHANGE THE INCORRECT VALUES

```
307 CALL LINFD(LINE,1)
10 WRITE(5,121)
121 FORMAT(12X,'INPUT LINE NO., NEW VALUE ',*)
READ(5,*,ERR=307) KK, DAT(KK)
```

* CHANGE AN INCORRECT VALUE

```
308 CALL LINFD(LINE,1)
WRITE(5,122)
122 FORMAT(12X,'ANOTHER CHANGE [Y,N]? ',*)
READ(5,200,ERR=308) ANS
IF(ANS.EQ.'Y') GO TO 10
GO TO 20
```

* RE-ECHO INPUT AFTER CHANGES ARE COMPLETE

END
SUBROUTINE CHPARA

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE IS USED TO INTERACTIVELY CHANGE THE MASS PARAMETERS OF THE LEG.

PARAMETERS INPUT

THETAM : ANGLE LOCATION OF CENTER OF MASS OF DRIVE CRANK. (RAD.)
GAMMA : ANGLE LOCATION OF CENTER OF MASS OF DRIVEN CRANK. (RAD.)
PHIM : ANGLE LOCATION OF CENTER OF MASS OF COUPLER. (RAD.)
LM1 : DISTANCE FROM DRIVE CRANK SUPPORT TO CENTER OF MASS OF DRIVE CRANK. (IN.)
LM2 : DISTANCE FROM DRIVEN CRANK SUPPORT TO CENTER OF MASS OF DRIVEN CRANK. (IN.)
LM3 : DISTANCE FROM COUPLER JOINT ON DRIVEN CRANK TO CENTER OF MASS OF COUPLER. (IN.)
M1 : MASS OF DRIVE CRANK (LBF. SEC2/IN.)
M2 : MASS OF DRIVEN CRANK (LBF. SEC2/IN.)
M3 : MASS OF COUPLER (LBF. SEC2/IN.)
JC01 : MASS MOMENT OF INERTIA OF DRIVE CRANK (LBF. IN. SEC2.)
JC02 : MASS MOMENT OF INERTIA OF DRIVEN CRANK (LBF. IN. SEC2.)
JC03 : MASS MOMENT OF INERTIA OF COUPLER (LBF. IN. SEC2.)

PARAMETERS RETURNED

THETAM, GAMMA, PHIM, LM1, LM2, LM3, JC01, JC02, JC03, M1, M2, M3

PARAMETERS INTERNAL

DAT : TEMPORARY STORAGE FOR INPUT PARAMETERS
DAT1:
KK : ARRAY SUBSCRIPT FOR DAT AND DAT1
I :
RTDD : CONVERT RADIANS TO DEGREES
IFL : FLAG FOR CONTROLLING CHANGING LOCATION OR VALUE
   : 0 TO CHANGE LOCATION
   : 1 TO CHANGE VALUE
   : 2 TO CHECK VALUES
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

EXTERNAL REFERENCES

DRWFRM, CHPAR1, CHPAR2, CHPAR4, CHPAR6, CHQED1, LINFED

CALLING ROUTINES

MAIN
PROGRAM PAGE = 39

C*****************************************************************************
C
C      DIMENSION DAT(12), DAT1(12)
C
C      COMMON /H/ THETAM, GAMMAM, PHIM, LM1, LM2, LM3, JC91, JC62, JC63,
C                M1, M2, M3
C
C      REAL LM1, LM2, LM3, JC91, JC62, JC63, M1, M2, M3
C
C      IFL=0
C      PI=4.0*ATAN(1.)
C      RTDD=180./PI
C      CALL DRWFRM(IPAGE)
C
C      DAT(1)=LM1
C      DAT(2)=LM2
C      DAT(3)=LM3
C      DAT(4)=M1
C      DAT(5)=M2
C      DAT(6)=M3
C      DAT1(1)=THETAM*RTDD
C      DAT1(2)=GAMMAM*RTDD
C      DAT1(3)=PHIM*RTDD
C      DAT1(4)=JC91
C      DAT1(5)=JC62
C      DAT1(6)=JC63
C
C      DO 1 I=1, 6
C      DAT(I+6)=DAT(I)
C      1
C
C      * IN CASE OF DEFAULT TO OLD VALUE STORE OLD VALUE
C
C      CALL CHPAR1
C
C      * DRAWS ILLUSTRATION FIGURE OF LEG PARAMETERS
C
C      CALL CHQED1(1913, B50, 1860, 20.)
C      LINE=26
C      CALL LINFEED(LINE, 1)
C      WRITE(5, 105)
C      105 FORMAT(TS6, 'CURRENT VERSION')
C      CALL CHPAR2(1, DAT(1), DAT1(1))
C      CALL CHPAR2(2, DAT(2), DAT1(2))
C      CALL CHPAR2(3, DAT(3), DAT1(3))
C
C      * DRAWS LATEST VERSION OF LEG
C      * AND SHOW LOCATION OF CURRENT MASS LOCATION
C
C      CALL CHPAR3(DAT, DAT1, LINE)
C
C      * LISTS LATEST VERSION OF LEG PARAMETERS
C
C      IF(1-IFL) 10, 20, 30
C      10
C      CALL LINFEED(LINE, 2)
C      WRITE(5, 100)
C
C      30
100 FORMAT(12X, 'CHANGE LOCATION OF MASS [Y,N]?', ',*, ERR=30) ANS
200 FORMAT(1A1)
   IF(ANS.EQ.'Y') CALL CHPARA(DAT, DAT1, IFL, LINE)
   IF(IFL.EQ.1) GO TO 6
C
   * GO TO LOCATION ROUTINE
C
20 CALL LINFE(LINE, 3)
   WRITE(5, 140)
140 FORMAT(12X, 'CHANGE VALUE OF MASS [Y,N]?', ',*, ERR=20) ANS
   IF(ANS.EQ.'Y') CALL CHPAR6(DAT, DAT1, IFL, LINE)
   IF(IFL.EQ.2) GO TO 6
C
   * GO TO MASS SELECTION ROUTINE
C
10 CALL LINFE(LINE, 2)
   WRITE(5, 145)
145 FORMAT(12X, 'ARE VALUES CORRECT [Y,N]?', ',*, ERR=10) ANS
   IF(ANS.EQ.'Y') RETURN
C
   * CHECK USERS INPUT
C
85 CALL LINFE(LINE, 3)
   WRITE(5, 150)
150 FORMAT(12X, 'INPUT LINE NO. VALUE VALUE', '//, T24, ', ',*, ERR=85) KK, DAT(KK), DAT1(KK)
   IF(DAT(KK).EQ.0.) DAT(KK) = DAT(KK+6)
   IF(DAT1(KK).EQ.0.) DAT1(KK) = DAT1(KK+6)
C
   * DIRECTIONS FOR ENTERING
C
300 CALL LINFE(LINE, 1)
   WRITE(5, 155)
155 FORMAT(12X, 'ANOTHER CHANGE [Y,N]?', ',*, ERR=300) ANS
   IF(ANS.EQ.'Y') GO TO 85
C
   * ENTER ANOTHER CHANGE?
C
6 LM1 = DAT(1)
   LM2 = DAT(2)
   LM3 = DAT(3)
   M1 = DAT(4)
   M2 = DAT(5)
   M3 = DAT(6)
   THETAM = DAT1(1)/RTOD
   GAMMA = DAT1(2)/RTOD
   PHIM = DAT1(3)/RTOD
   JC01 = DAT1(4)
   JC02 = DAT1(5)
   JC03 = DAT1(6)
   GO TO 5
C
   * RETURN TO TOP OF ROUTINE
194
C

END
PROGRAM PAGE = 42

SUBROUTINE CHPARA

******************************************************************************

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DRAWS A MASTER LINKAGE ON SCREEN INDICATING
DESCRIPTION OF VARIABLES USED IN CHPARA

PARAMETERS INPUT

NONE

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

IH : HORIZONTAL WIDTH OF CHARACTER
IV : VERTICAL LENGTH OF CHARACTER

EXTERNAL REFERENCES

BUSH, PLOTIO ROUTINES

CALLING ROUTINES

CHPARA

******************************************************************************

CALL SWIND0(0.4096, 0.3120)
CALL MOVABS(1373.1150)
CALL DRWREL(840.0)
CALL DRWREL(0.915)
CALL MOVABS(1373.1450)
CALL DRWREL(840.0)
CALL DRWREL(0.300)

* DRAW BOX AROUND FIGURE AND LABEL

CALL SWIND0(1383.800, 330.800)
CALL VWIND0(0.20, 0.1, 0.20)

* USE LOWER LEFT CORNER OF DRAWING FRAME

CALL BUSH(.25, 13., 14., 180.)
CALL BUSH(.25, 7., 18., 180.)

* DRAW BUSHINGS AT PIN LOCATIONS

CALL MOVEA(7.4, 12.)
CALL DRAWA(7., 18.)
CALL DRAWA(11., 10.)

196
CALL DRAWA(10., 0.)
CALL MOVEA(13.5, 10.5)
CALL DRAWA(13., 14.)
CALL DRAWA(17., 9.)
CALL DRAWA(11., 10.)
CALL DRAWA(14., 7.)

* DRAWS MAJOR OUTLINE OF LINKAGE

CALL CIAxis(1)
CALL CHRSII(4)
CALL CSIZE(IH, IV)
IH=FLOA(TIH)/3.
IV=FLOAT(IV)/4.
CALL MOVEA(14., 7.)
CALL MOVREL(-IH, -IV)
CALL ANCHO(111)
CALL MOVEA(13.5, 10.5)
CALL MOVREL(-IH, -IV)
CALL ANCHO(111)
CALL MOVEA(7.4, 12.)
CALL MOVREL(-IH, -IV)
CALL ANCHO(111)

* DRAWS "o" AT EACH CENTER OF MASS

CALL MOVEA(8.6, 14.8)
CALL DRAWA(7.3, 14.4)
CALL ARROW(0.7, 3.14, 4., 7.10.)

CALL MOVEA(14.4, 12.2)
CALL DRAWA(13.2, 11.8)
CALL ARROW(0.13, 2.11, 8., 7.15.)

CALL MOVEA(13.2, 9.6)
CALL DRAWA(12.6, 8.4)
CALL ARROW(0.12, 6.8, 4., 7.60.)

* LOCATES ANGLE MEASUREMENT

CALL MOVEA(11., 18.)
CALL DRAWA(10., 18.)
CALL DRAWA(13.2, 12.4)
CALL ARROW(0.13, 2.12, 4., 7.120.)

* POINTER FOR L-1

CALL MOVEA(18., 7.)
CALL DRAWA(17., 7.)
CALL DRAWA(14., 12.)
CALL ARROW(0.14, 12., 7.300.)

* POINTER FOR A-1

CALL MOVEA(3.2, 11.4)
CALL DRAWA(4.2, 11.4)
CALL DRAWA(7.2, 16.)

197
CALL ARROW(0, 7, 2, 16, 7, 240)
* POINTER FOR L-2
CALL MOVEA(3, 8)
CALL DRAWA(4, 8)
CALL DRAWA(7, 8, 14, 4)
CALL ARROW(0, 7, 8, 14, 4, 7, 240)
* POINTER FOR A-2
CALL MOVEA(17, 6, 2)
CALL DRAWA(16, 2, 2)
CALL DRAWA(13, 8)
CALL ARROW(0, 13, 8, 7, 290)
* POINTER FOR L-3
CALL MOVEA(18, 2, 4)
CALL DRAWA(17, 2, 4)
CALL DRAWA(13, 2, 9, 2)
CALL ARROW(0, 13, 2, 9, 2, 7, 310)
* POINTER FOR A-3
CALL MOVEA(11, 2, 18)
CALL ANCHO(76)
CALL ANCHO(45)
CALL ANCHO(49)
* LABEL L-1
CALL MOVEA(18, 2, 7)
CALL ANCHO(65)
CALL ANCHO(45)
CALL ANCHO(49)
* LABEL A-1
CALL MOVEA(9, 11, 4)
CALL ANCHO(76)
CALL ANCHO(45)
CALL ANCHO(50)
* LABEL L-2
CALL MOVEA(7, 8)
CALL ANCHO(65)
CALL ANCHO(45)
CALL ANCHO(50)
* LABEL A-2
CALL MOVEA(17, 8, 2)
CALL ANCHO(76)
CALL ANCHO(45)
CALL ANCHO(51)
* LABEL L-3
CALL MOVEA(18, 4, 4.)
CALL ANCHO(65)
CALL ANCHO(45)
CALL ANCHO(51)

* LABEL A-3
LINE=19
CALL LINFD(LINE, 1)
WRITE(5, 101)
101 FORMAT(T52, 'MASTER LEO MODEL')
CALL LINFD(LINE, 2)
WRITE(5, 100)
100 FORMAT(T48, '"o" DENOTES MASS CENTER', /
 & T52, 'OF LINK.')

* LABEL DIAGRAM
CALL CIZAXIS(0)
CALL TSND
RETURN
END
SUBROUTINE CHPAR2(I, RLEN, ANG)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

This routine is used to compute (X, Y) location of the center of mass of the link using the cursor mode.

PARAMETERS INPUT

AX : X LOCATION OF DRIVE CRANK SUPPORT (VIRTUAL CORD)
AY : Y " " " " " "
BX : X LOCATION OF DRIVE CRANK END (VIRTUAL CORD)
BY : Y " " " " " "
CX : X LOCATION OF DRIVEN CRANK END (VIRTUAL CORD)
CY : Y " " " " " "
DX : X LOCATION OF DRIVEN CRANK SUPPORT (VIRTUAL CORD)
DY : Y " " " " " "
EX : X LOCATION OF FOOT POINT (VIRTUAL CORD)
EY : Y " " " "
I : CONTROL PARAMETER FOR LINK DESIGNATION
   =10  LINK 1 (DRIVE LINK)
   =20  LINK 2 (DRIVEN LINK)
   =30  LINK 3 (COUPLER LINK)
RLEN : DISTANCE TO CENTER OF MASS OF LINK
ANG : ANGLE LOCATION OF CENTER OF MASS

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

X1 : X LOCATION OF POINT ON LINK
X2 : " " " " " "
Y1 : Y LOCATION OF POINT ON LINK
Y2 : " " " " " "
X3 : X LOCATION OF CENTER OF MASS
Y3 : Y LOCATION OF CENTER OF MASS

EXTERNAL REFERENCES

ANGLE1, PLOT10 ROUTINES

CALLING ROUTINES

CHPARA

END

200
COMMON /CC/ AX,AY,BX,BY,CX,CY,DX,DY,EX,EY
C
GO TO (10,20,30), I
C
* CONTROL PARAMETER DEFINING WHICH LINK
C
10 CONTINUE
  X1=AI
  X2=BX
  Y1=AY
  Y2=BY
C
GO TO 40
C
* DEFINES X,Y FOR LINK 1
C
20 CONTINUE
  X1=DX
  X2=CX
  Y1=DY
  Y2=CY
C
GO TO 40
C
* DEFINES X,Y FOR LINK 2
C
30 CONTINUE
  X1=CX
  X2=BX
  Y1=CY
  Y2=BY
C
GO TO 40
C
* DEFINES X,Y FOR LINK 3
C
40 CONTINUE
  ANG0=ANG0
  CALL ANGLE1(ANG1,X1,Y1,X2,Y2)
  ANG1=ANG1
  ANG0=ANG0*.ATAN(1.)/180.
  ANG2=ANG0+ANG
  X3=X1+RLEN*COS(-ANG2)
  Y3=Y1+RLEN*SIN(-ANG2)
  ANG=ANG0

C
* COMPUTE LOCATION OF CENTER OF MASS
C
CALL MOVEA(X3,Y3)
CALL CSIZE(IH,IV)
CALL MOVREL(-IH/3,-IV/4)
CALL ANCHO(111)
CALL ANCHO(I+48)
C
* OUTPUT "O" AND NO. AT CENTER OF MASS

201
RETURN
END
SUBROUTINE CHPAR3(DAT, DAT1, LINE)

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE LISTS THE PREVIOUS VALUES OF MASS, LOCATION OF
MASS, AND JCO FOR THE LINKAGE.

PARAMETERS INPUT

DAT : TEMPORARY ARRAY STORAGE OF INPUT PARAMETERS
DAT1 : " " " " " " " "
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

LINFED

CALLING ROUTINES

CHPARA

DIMENSION DAT(12), DAT1(12)

LINE=1
CALL LINFED(LINE, 10)
WRITE(5, 100)
100 FORMAT(12X, 'CHANGING LEG MASS PARAMETERS', //,
* 12X, 'FROM CURRENT VALUES', //, 21X, 'LOCATION', //,
* T24, 'LENGTH', T36, 'ANGLE', //,
* T12, 'NO.', T16, 'LINK', T26, 'L', T38, 'A')

* LABEL THE DATA BEING ECHOED

WRITE(5, 105) DAT(1), DAT(1), DAT(2), DAT(2), DAT(3), DAT(3)
105 FORMAT(T12, ' (1)', T17, '1', T20, 2(1X, E11.4), /,
* T12, ' (2)', T17, '2', T20, 2(1X, E11.4), /,
* T12, ' (3)', T17, '3', T20, 2(1X, E11.4))

* ECHO LOCATION DATA

CALL LINFED(LINE, 8)
WRITE(5, 110)

203
110 FORMAT(22X,'VALUE',/,'T12','NO.',
         'T16','LINK','T24','MASS','T37','JCG')
    WRITE(5,115)DAT(4),DAT(1),DAT(5),DAT(1),DAT(6),DAT(1)
115 FORMAT(T12,'(4)',T17,'1',T20,2E11.4)/,
         T12,'(5)',T17,'2',T20,2E11.4)/,
         T12,'(6)',T17,'3',T20,2E11.4)
C
C        * ECHO MASS DATA
C
RETURN
END
SUBROUTINE CHPAR4(DAT,DAT1,IFL,LIN
E)

C**********************************************************************************************

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE IS USED TO CHANGE THE LOCATION OF THE CENTER
OF MASS OF A LINK USING KEYBOARD OR CURSOR INPUT.

PARAMETERS INPUT

DAT   :TEMPORARY ARRAY STORAGE OF INPUT PARAMETERS
DAT1  :
LINE  :LINE SPACING PARAMETER FOR SCREEN WRITES

PARAMETERS RETURNED

DAT, DAT1
IFL   :FLAG DESIGNATING CHANGE OF MASS LOCATION HAS OCCURRED

PARAMETERS INTERNAL

I      :DO LOOP COUNTER
RE   N  :DISTANCE TO CENTER OF MASS OF LINK
ANG   :ANGLE LOCATION OF CENTER OF MASS

EXTERNAL REFERENCES

CHPAR5, LINFED

CALLING Routines

CHPARA

C**********************************************************************************************

DIMENSION DAT(12),DAT1(12)

300 CALL LINFED(LINE,2)
WRITE(5,100)
100 FORMAT(1X,'BY CURSOR OR KEYBOARD [C,K]? ','
READ(5,200,ERR=300) ANG
200 FORMAT(1A1)
   IF(ANG.EQ. 'C') 90 TO 20

C
* DESIDE ON CURSOR OR KEYBOARD FOR LOCATING
C
CALL LINFED(LINE,6)
WRITE(5,110)
110 FORMAT(1X,'LOCATE CENTER OF MASS',/.
   $ 12X,'INPUT OF "0"',/.
   $ 12X,'DEFAULTS TO OLD VALUE',/.
   $ 12X,'INPUT LENGTH ANGLE')
* INSTRUCTIONS FOR INPUT

DO 10 I=1,3
301 CALL LINFED(LINE,1)
WRITE(5,120) I
120 FORMAT(12X,'LINK ',II,'.',I6,'&')
READ(5,*,ERR=301) DAT(I),DAT1(I)
IF(DAT(I).EQ.0.) DAT(I) = DAT(I+6)
IF(DAT1(I).EQ.0.) DAT1(I) = DAT1(I+6)
10 CONTINUE

* KEYBOARD INPUT OF LOCATION OF MASS

IFL=1
RETURN

CALL LINFED(LINE,5)
WRITE(5,130)
130 FORMAT(12X,'LOCATE MASS C.O. ON CURRENT',/,
* 12X,'LEG VERSION AND PRESS "SPACE"',/,
* T24,'LENGTH',T37,'ANGLE')

* INSTRUCTIONS FOR CURSOR USE

DO 70 I=1,3
CALL LINFED(LINE,0)
WRITE(5,140) I
140 FORMAT(12X,'LINK ',II,'.')
CALL CHPAR5(I,RLEN,ANG)
DAT(I)=RLEN
DAT1(I)=ANG
CALL LINFED(LINE,-1)
WRITE(5,150) RLEN,ANG
70 CONTINUE

* ECHO CURSOR INPUT

* CHPAR4 CONVERTS CURSOR INPUT TO LENGTH AND ANGLE

LINE=LINE+2
302 CALL LINFED(LINE,1)
WRITE(5,135)
135 FORMAT(12X,'"CR" TO CONTINUE',/)
READ(5,200,ERR=302) IFL
IFL=1
RETURN
END
SUBROUTINE CHFAR5(I, RLEN, ANG)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE IS USED TO INPUT LOCATION OF CENTER OF MASS A LINK USING THE CURSOR

PARAMETERS INPUT

  I    : CONTROL VARIABLE FOR LINK NO.
  AX   : X LOCATION OF DRIVE CRANK SUPPORT (VIRTUAL CORD)
  AY   : Y    
  BX   : X LOCATION OF DRIVE CRANK END (VIRTUAL CORD)
  BY   : Y    
  CX   : X LOCATION OF DRIVEN CRANK END (VIRTUAL CORD)
  CY   : Y    
  DX   : X LOCATION OF DRIVEN CRANK SUPPORT (VIRTUAL CORD)
  DY   : Y    
  EX   : X LOCATION OF FOOT POINT (VIRTUAL CORD)
  EY   : Y    

PARAMETERS RETURNED

  RLEN : DISTANCE TO CENTER OF MASS OF LINK
  ANG  : ANGLE LOCATION OF CENTER OF MASS

PARAMETERS INTERNAL

  X1   : X LOCATION OF POINT ON LINK
  X2   : "    
  Y1   : Y LOCATION OF POINT ON LINK
  Y2   : "    
  X3   : X LOCATION OF CENTER OF MASS
  Y3   : Y LOCATION OF CENTER OF MASS
  ANG1 : TEMP ANGLE FOR COMPUTING ANG.
  ANG2 : "    

EXTERNAL REFERENCES

  ANGLE1, PLOT10 Routines

CALLING Routines

  CHFAR4

COMMON /CC/ AX, BX, BY, CX, CY, DX, DY, EX, EY

GO TO(10, 20, 30), I
* CONTROL PARAMETER DEFINING WHICH LINK

10 CONTINUE
   X1=AX
   X2=BX
   Y1=AY
   Y2=BY

20 CONTINUE
   X1=DX
   X2=CX
   Y1=DY
   Y2=CY

QD TO 40

30 CONTINUE
   X1=CX
   X2=BX
   Y1=CY
   Y2=BY

QD TO 40

40 CONTINUE

CALL VCURSR(J,X3,Y3)
CALL BEL
CALL MOVEA(X1,Y1)
CALL DRAMA(X3,Y3)
CALL CSIZE(IH,IV)
CALL MOVREL(-IH/3,-IV/3)
CALL ANCHO(42)
CALL ANCHO(I+4B)

* REGISTER LOCATION USING CURSOR
* OUTPUT A "*" AND THE NO. AT THE POINT

RLEN=SQRT((X3-X1)**2+(Y3-Y1)**2)

* COMPUTES LENGTH OF RLEN

CALL ANGLE1(ANG1,X1,Y1,X2,Y2)
CALL ANGLE1(ANG2,X1,Y1,X3,Y3)

ANG=ANG1-ANG2
ANG=ANG*180./4.0/ATAN(1.)

* COMPUTES ANGLE ANG IN DEGREES
C
RETURN
END
SUBROUTINE CHPAR6(DAT, DAT1, IFL, LINE)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTIN ALLOWS INTERACTIVE INPUT OF THE MASS AND MASS MOMENT OF INERTIA OF EACH LEG LINK.

PARAMETERS INPUT

DAT : TEMPORARY ARRAY STORAGE OF INPUT PARAMETERS
DAT1 : " " " " " "
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

PARAMETERS RETURNED

DAT, DAT1
IFL : FLAG DESIGNATES MASS HAS BEEN CHANGED

PARAMETERS INTERNAL

J : ARRAY SUBSCRIPT COUNTER

EXTERNAL REFERENCES

LINFED
CALLING Routines

CHPARA

DIMENSION DAT(12), DAT1(12)

CALL LINFED(LINE, 6)
WRITE(5, 100)
100 FORMAT(12X, 'INPUT NEW VALUE OF MASS AND', /
* 12X, 'MOMENT OF INERTIA 'JQC''/, /
* 12X, 'INPUT OF 'O''/, /
* 12X, 'DEFAULTS TO OLD VALUE', //,
* 12X, 'INPUT MASS 'JQC')

* INSTRUCTIONS FOR INPUT

DO 10 I=4, 6
300 CALL LINFED(LINE, 1)
J=I-3
WRITE(5, 120) J
120 FORMAT(12X, 'LINK ', 'I1', ', ', ')
READ(5, 100) DAT(I), DAT1(I)
IF(DAT(I).EQ.0.) DAT(I) = DAT(I+6)
IF(DAT1(I).EQ.0.) DAT1(I) = DAT1(I+6)

210
10 CONTINUE

* INPUT OF MASS AND JCG

IFL=2
RETURN
END
SUBROUTINE CHSTRD

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE IS USED TO CHANGE THE STRIDE AND FOOT PLACEMENT AND TO CHANGE THE INITIAL SPEED OF THE MACHINE.

PARAMETERS INPUT

PI : PI = 4.0 * ATAN(1.)
FOOTST : LENGTH OF LEG STRIDE (IN.)
EXPLAC : DISTANCE FROM DRIVE CRANK SUPPORT TO TOE-DOWN POSITION — POSITIVE RIGHT — (IN.)
SHORTL : AMOUNT DRIVEN CRANK IS SHORTENED FOR RETURN PHASE OF LEG (IN.)
SPEED : AVERAGE SPEED OF MACHINE (IN./SEC.)
INCRE : NUMBER OF INCREMENTS USED IN TIME STEPPING

PARAMETERS RETURNED

FOOTST, EXPLAC, SPEED, INCRE, SHORTL
EXPICK : DISTANCE FROM DRIVE CRANK SUPPORT TO TOE-OFF POSITION — POSITIVE RIGHT — (IN.)
PERIOD : TIME FOOT IS IN CONTACT WITH GROUND.
= 0.5 * TOTAL LEG PERIOD (SEC.)
INC1 : INCREMENT OF "INCRE" AT WHICH POSITION IS SAVED FOR DRAWING
DELTIM : INCREMENT OF TIME STEP (SEC.)

PARAMETERS INTERNAL

DAT : TEMPORARY ARRAY STORAGE OF INPUT PARAMETERS
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

EXTERNAL REFERENCES

CHST1, DWF2RM, LINFED

CALLING ROUTINES

MAIN

DIMENSION DAT(10)

COMMON / CH/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
* INC, INC1, DELTIM, SHORTL, BL2

PI = 4.0 * ATAN(1.0)
CALL DWFMRM(IPAGE)
CALL CHSTR:

LINE=1
CALL LNFED(LINE, 4)
WRITE(5, 100)

* DESCRIBES PROGRAM INPUTS

CALL LNFED(LINE, 8)
WRITE(5, 111) FOOTST
WRITE(5, 112) EXPLAC
WRITE(5, 113) SHORTL
WRITE(5, 114) SPEED
WRITE(5, 115) INCRE

* PRINT CURRENT VALUES

301 CALL LNFED(LINE, 2)
WRITE(5, 101)
READ(5, *, ERR=301) DAT(1)
IF(DAT(1), EQ, 0) DAT(1)=FOOTST

302 CALL LNFED(LINE, 1)
WRITE(5, 102)
READ(5, *, ERR=302) DAT(2)
IF(DAT(2), EQ, 0) DAT(2)=EXPLAC

303 CALL LNFED(LINE, 1)
WRITE(5, 103)
READ(5, *, ERR=303) DAT(3)
IF(DAT(3), EQ, 0) DAT(3)=SHORTL

304 CALL LNFED(LINE, 1)
WRITE(5, 104)
READ(5, *, ERR=304) DAT(4)
IF(DAT(4), EQ, 0) DAT(4)=SPEED

305 CALL LNFED(LINE, 1)
WRITE(5, 105)
READ(5, *, ERR=305) DAT(5)
IF(DAT(5), EQ, 0) DAT(5)=INCRE

* READ IN VALUE OF INCRE

100 FORMAT(12X, 'DEFINE NEW STRIDE PARAM.'),/,
PROGRAM PAGE = 60

* 12X, 'INPUT OF "0",/,
* 12X, 'DEFAULTS TO OLD VALUE ')
101 FORMAT(12X, 'INPUT NEW STRIDE LENGTH ',0)
102 FORMAT(12X, 'INPUT NEW TOE-DOWN POS. ',0)
103 FORMAT(12X, 'INPUT NEW LINK 2 SHORT ',0)
104 FORMAT(12X, 'INPUT NEW AVERAGE SPEED ',0)
105 FORMAT(12X, 'INPUT NEW NO. OF TIME INC. ',0)

C
20 FOOTST=DAT(1)
   EXPLAC=DAT(2)
   SHORTL=DAT(3)
   SPEED=DAT(4)
   INCRE=DAT(5)

C *
C C * CONVERTS DAT ARRAY TO ACTUAL PARAMETERS
C C C
C CALL DRWFRM(IPAGE)
C CALL CHSTR1

C C LINE=1
C CALL LINFEH(LINE,8)
C WRITE(5,110)
C WRITE(5,111) FOOTST
C WRITE(5,112) EXPLAC
C WRITE(5,113) SHORTL
C WRITE(5,114) SPEED
C WRITE(5,115) INCRE

C C *
C C C *
C C C C ECHO'S THE INPUT DATA
C C C
C 110 FORMAT(12X, 'NEW STRIDE PARAMETERS')
C 111 FORMAT(12X, '(1) STRIDE LENGTH = ',F9.3)
C 112 FORMAT(12X, '(2) TOE-DOWN POS. = ',F9.3)
C 113 FORMAT(12X, '(3) LINK 2 SHORT = ',F9.3)
C 114 FORMAT(12X, '(4) AVERAGE SPEED = ',F9.3)
C 115 FORMAT(12X, '(5) NO. OF TIME INC. = ',I5)

C C EXPICK=EXPLAC-FOOTST
C PERIOD=FOOTST/SPEED
C INC!=(INCRE/40
C DELTIM=PERIOD/FLOAT(INCRE)

C C *
C C C C COMPUTE PARAMETERS FOR TIMING
C C
C 306 CALL LINFEH(LINE,2)
C WRITE(5,120)
C 120 FORMAT(12X, 'ARE VALUES CORRECT [Y,N]? ',0)
C READ(5,200,ERR=306) ANS
C 200 FORMAT(1A1)
C IF(ANS.EQ. 'Y') RETURN

C C *
C C C * IF INPUT DATA IS CORRECT RETURN—ELSE CHANGE
C C C * THE INCORRECT VALUES.
CALL LINFED(LINE,1)
WRITE(5,121)
FORMAT(12X,'INPUT LINE NO., NEW VALUE ',*)&
READ(5,*ERR=307) KK,DAT(KK)

* CHANGE AN INCORRECT VALUE

CALL LINFED(LINE,1)
WRITE(5,122)
FORMAT(12X,'ANOTHER CHANGE [Y,N]? ',*)&
READ(5,200,ERR=308) ANS
IF(ANS.EQ.'Y') GO TO 10
GO TO 20

* RE-ECHO INPUT AFTER CHANGES ARE COMPLETE

END
SUBROUTINE CHSTR1

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DRAWS THE TOE-DOWN AND TOE-OFF POSITIONS OF
THE LEG FOR ROUTINE CHSTRD

PARAMETERS INPUT

PERIOD : TIME FOOT IS IN CONTACT WITH GROUND.
L2 : LENGTH OF DRIVEN CRANK

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

BL2 : TEMPORARY STORAGE FOR L2
TIME1 : TIME AT WHICH TOE-DOWN/TOE-OFF OCCUR
THEPLA : DRIVE CRANK ANGLE AT TOE-DOWN
THEPIC : DRIVE CRANK ANGLE AT TOE-DOWN
EX : X-POSITION OF FOOT POINT
Ey : Y-POSITION OF FOOT POINT
EX1 : TEMP. STORAGE OF EX
EX2 :
Ey1 : " " " Ey

EXTERNAL REFERENCES

PICKOF, CHGE01, CHGE02, PLOT10 Routines

CALLING ROUTINES

CHSTRD

COMMON /C/ L2,THETA,GAMMA,PHI
COMMON /G/ PI,FOOTST,EXPLAC,EXPICK,SPEED,PERIOD,
INCRE,INCI,DELTIM,SHORTL,SL2
COMMON /CC/ AX,AY,BX,BY,CX,CY,DX,DY,EX,EY

REAL L2

SL2=L2
TIME1=0.
CALL PICKOF(TIME1)
THEPLA=THETA*180./PI
TIME1=PERIOD
CALL PICKOF(TIME1)
THEPIC=THETA*180./PI

216
* COMPUTE THETA AT TOE-DOWN AND TOE-OFF

CALL CHQED1(1800, 310, 2500, THEPLA)
EY1=EY
EX1=EX

* DRAW LEQ AT THETA TOE-DOWN

CALL CHQED2(1)

* LABELS LEQ POINTS

CALL CHQED1(1800, 310, 2500, THEPIC)
EX2=EX

* DRAW LEQ AT THETA TOE-OFF

CALL MOVEA(EX1, EY1)
CALL DASHA(EX2, EY1, 43)

* DRAW GROUND LINE

RETURN
END
SUBROUTINE CIRCLE (XC, YC, THO, THF, R, IPEN)

******************************************************************************

AUTHOR: GARY L. KINZEL

THIS ROUTINE PLOTS A CIRCLE OR AN ARC CENTERED AT (XC, YC) AND OF RADIUS R. THO IS THE STARTING ANGLE OF THE ARC; THF THE STOPPING ANGLE, BOTH OF WHICH ARE MEASURED CCW FROM THE POSITIVE HORIZONTAL AXIS IN DEGREES. IPEN = 0 draws a SOLID ARC, IPEN NOT EQUAL TO 0 draws a DASHED ARC.

SEE PLOT 10 MANUAL FOR CODE TO CONTROL DASH SIZE.

XC, YC, THO, THF, & R ARE REAL ARGUMENTS. IPEN IS AN INTEGER.

******************************************************************************

CHECK FOR VALID DRAWING PARAMETERS SPECIFIED IN ARGUMENT LIST.

IF(THO. GE. THF) RETURN
IF(R. EQ. 0.0) RETURN

CONVERSION FACTOR FROM DEGREES TO RADIANS.

RAD=ATAN(1.0)/45.0

DIVIDE CIRCULAR ARC INTO NO MORE THAN 360 CHORDS.

DT=(THF-THO)/360.

1 DT=2.*DT
IF(ABS(DT). LT. 2.0) GO TO 1

CONVERSION FROM DEGREES TO RADIANS.

DT=DT*RAD
T=THO*RAD

MOVE TO STARTING POINT OF CIRCLE DRAWING.

X=XC+R*COS(T)
Y=YC+R*SIN(T)
CALL MOVEA (X, Y)
TF=THF*RAD

INCREMENT ANGLE AND ITERATE DRAWING CHORDS UNTIL STOPPING ANGLE IS REACHED.

2 T=T+DT
IF(T . GT. TF) T=TF
X=XC+R*COS(T)
Y=YC+R*SIN(T)

IF(IPEN. EQ. 0) CALL DRAWA (X, Y)
IF(IPEN. NE. 0) CALL DASHA (X, Y, L)

218
IF (T.EQ.TF) QQ TO 3
   QQ TO 2
   CALL TSEND
   CALL ANMODE
   RETURN
END
SUBROUTINE CNTMAS

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE COMPUTES THE CENTER OF MASS OF THE MACHINE
IN THE X-DIRECTION. IT MUST BE CALLED EXACTLY 80 TIMES
CORRESPONDING TO THE EIGHTY FRAMES THE LEG IS DRAWN.

PARAMETERS INPUT

LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT —LEFT POS.— (IN.)
NUM1 : ARRAY SUBSCRIPT FOR STORING DRAWING VERSION OF LEG
BTHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG (RAD.)
BGAMMA : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEG (RAD.)
BPHI : ARRAY OF COUPLER ANGLES FOR DRAWING LEG (RAD.)
BL2 : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEG (IN.)
THETAN : ANGLE LOCATION OF CENTER OF MASS OF DRIVE
         CRANK, CW FROM DRIVE CRANK (RAD.)
GAMMAN : ANGLE LOCATION OF CENTER OF MASS OF DRIVEN
         CRANK, CW FROM DRIVEN CRANK (RAD.)
PHIM : ANGLE LOCATION OF CENTER OF MASS OF COUPLER
       CW FROM COUPLER CRANK (RAD.)
LM1 : DISTANCE FROM DRIVE CRANK SUPPORT TO CENTER
      OF MASS OF DRIVE CRANK (IN.)
LM2 : DISTANCE FROM DRIVEN CRANK SUPPORT TO CENTER
      OF MASS OF DRIVEN CRANK (IN.)
LM3 : DISTANCE FROM COUPLER JOINT ON DRIVEN CRANK
      TO CENTER OF MASS OF COUPLER (IN.)
M1 : MASS OF DRIVE CRANK (LBF. SEC2/IN.)
M2 : MASS OF DRIVEN CRANK (LBF. SEC2/IN.)
M3 : MASS OF COUPLER (LBF. SEC2/IN.)

PARAMETERS RETURNED

XMASS : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE

PARAMETERS INTERNAL

COSTHM : COSINE OF ANGLE LOCATIN C. O. OF LINK 1
COSGAM : " " " " " " " 2
COSPHI : " " " " " " " 3
TMLEG : TOTAL MASS OF LEG

EXTERNAL REFERENCES

NONE

CALLING ROUTINES

LEPOS

***************************************************************************
PROGRAM PAGE = 67

C
C DIMENSION BTHETA(80), BGOAMMA(80), BPHI(80), BL2(80), XMASS(80)
C
C COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMOM /B/ NUM1, BTHETA, BGOAMMA, BPHI, BL2, XMASS
COMMON /H/ THETAM, GAMMAM, PHIM, LM1, LM2, LM3, JG01, JG02, JG03,
M1, M2, M3
COONOM /BB/ PLACLO, SPACLO, BDWIDT, BDLENT, BDWEIT
C
REAL L1, L3, L5, LX, LY, LM1, LM2, LM3, M1, M2, M3, JG01, JG02, JG03
C
* JG01, JG02, JG03 ARE NOT USED IN THIS ROUTINE

C C COSTHM=COS(BTHETA(NUM1)+THETAM)
COSGAM=COS(BGOAMMA(NUM1)+GAMMAM)
COSPHM=COS(BPHI(NUM1)+PHIM)
TMLEQ=M1+M2+M3
C
* VARIABLES REQUIRED FOR EXECUTION

C XMASS(NUM1)=[(LM1+LM2)*COSTHM*M1+LM2*COSGAM*M2+
(BL2(NUM1)*COS(BGOAMMA(NUM1)))+LM3*COSPHM*M3]/TMLEQ
C
C * COMPUTES CENTER OF MASS OF LEG WITH RESPECT TO THE
C * POINT D ON LINK 2.  NUM1 CORRESPONDS TO THE LEQ
C * POSITION IN THE DRAWING.

C IF(NUM1.NE.80) RETURN
C
C * TESTS FOR THE LAST VALUE OF XMASS(J) TO BE COMPUTED
C * IF LAST VALUE THEN COMPUTE CENTER OF MASS OF BODY
C
DO 10 J=1, 40
XMASS(J)=(BDWEIT/386.0*(SPACLO+LX/2.0)+XMASS(J)+
3.0*XMASS(J+40)+6.0*SPACLO)*TMLEQ)/
(BDWEIT/386.0+6.0*TMLEQ)
XMASS(J+40)=XMASS(J)
10 CONTINUE
C
C * COMPUTES BODY CENTER OF GRAVITY IN X-DIRECTION
C * FROM POINT D OF FIRST LEQ.
C * THE BODY WEIGHT (BDWEIT) IS DIVIDED BY Q=386.0 IN/SEC.
C * TO PUT IT INTO EQUILVALENT MASS UNITS

RETURN
END
SUBROUTINE COMACC(T,Y1,Y2,Y3,LINE)

**DESCRIPTION**

This routine computes the angular acceleration of the fixed crank of the four-bar. The 9x9 matrix equation is solved.

**PARAMETERS INPUT**

- **IFIX**: Fix length L2 at extended length
- **T**: Time in Runge Kutta step
- **Y1**: Theta in Runge Kutta step
- **Y2**: THDOTT in Runge Kutta step
- **LINE**: Line spacing parameter for screen writes (IN.)
- **L2**: Length of driven crank
- **THETA**: Drive crank angle measured from POS. X (RAD.)
- **HORIZONTAL**: CW --- (CW POS.)
- **THDOTT**: Angular velocity of drive crank
- **EXPLAC**: Distance from drive crank support to toe-down position --- positive right --- (IN.)
- **SPEED**: Average speed of machine (IN./SEC.)
- **INCI**: Increment of "INCRE" at which position is saved for drawing
- **AX**: Acceleration of body in x-direction (IN./SEC2)
- **AY**: Acceleration of body in y-direction (IN./SEC2)

**PARAMETERS RETURNED**

- **Y3**: THDOTT (THEACC) in Runge Kutta step

**PARAMETERS INTERNAL**

- **A**: Matrix array of 9x9 equation coefficients
- **B**: Matrix array of 9x1 equation solution
- **IERR**: Error indicator in matrix solution
- **FAR**: Radial force at joint A
- **FAT**: Tangential " " " " A
- **FBR**: Radial " " " B
- **FBT**: Tangential " " " B
- **FCR**: Radial " " " C
- **FCT**: Tangential " " " C
- **FDR**: Radial " " " D
- **FDT**: Tangential " " " D
- **THEACC**: Angular acceleration of drive crank
- **TAF**: Magnitude of force at joint A
- **TBF**: " " " " " B
- **TCF**: " " " " " C
- **TDF**: " " " " " D

**EXTERNAL REFERENCES**

VERACC, ANGLE, VELOC, ACCEL, LINE8
CALLING ROUTINES

RKSTEP

******************************************************************************

DIMENSION A(9,9), B(9,1)

COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /D/ THDCT1, QADOT1, PHDOT1
COMMON /E/ THDOT2, QADOT2, PHDOT2
COMMON /Q/ PI, FOOTST, EXPLAC, EXPIK, SPEED, PERIOD,
* INCRES, INC1, DTLIM, SHORTL, SL2
COMMON /H/ THETAM, GAMMAN, PHIM, LM1, LM2, LM3, JG01, JG02, JG03,
* M1, M2, M3
COMMON /AA/ RA, RB, RC, RD, MUA, MUB, MUJ, MUD, FX, FY, AX, AY, BI, Q
COMMON /BB/ PLAQLG, SPAQLG, BDWDT, BDLNCT, BDWEIT
COMMON /DD/ TAF, TBF, TCF, TDF

REAL L2

TIME=T
THETA=Y1
THDCT1=Y2

* DEFINES THE VARIABLES PASSED TO THE ROUTINE
CALL VERACC(AX, AY, TIME)

* DETERMINES THE VERTICAL ACCELERATION OF BODY
CALL ANGLE
CALL VELOC
CALL ACCEL

* DETERMINES POSITION ANALYSIS OF LEG

ITEST2=0

30 CALL VALUES(A, B)

* SETS VALUES FOR MATRIX EQUATION
CALL LINEQ(A, B, IERR)
IF(IERR.EQ.1) CALL LINFED(LINE, 2)
IF(IERR.EQ.1) WRITE(5, 300) IERR
300 FORMAT(12X, 'ERROR IN SOLUTION ROUTINE DF',
* 12X, '9x9 MATRIX EQUATION. IERR= ', I5)

* SOLVES 9x9 MATRIX EQUATION FOR PIN FORCES AND THETA DOT2

FAR=B(1,1)
FAT=B(2,1)
FBR=B(3,1)
FBT=B(4,1)
PROGRAM PAGE = 70

FCR=B(5,1)
FCT=B(6,1)
FDR=B(7,1)
FDT=B(8,1)
THEACC=B(9,1)

C
C  * STORES RESULTS FROM MATRIX ROUTINE SOLUTION
C
IF(ITEST2 EQ 1) GO TO 35
TAF=SQRT(FAR**2+FA7**2)
TBF=SQRT(FBR**2+FBT**2)
TCF=SQRT(FCR**2+FCT**2)
TDT=SQRT(FDR**2+FDT**2)
ITEST2=1
GO TO 30
35 CONTINUE

C
C  * ITERATE THE MAGNITUDE OF THE FORCE AT THE
C  * JOINTS FOR FRICTION TORQUE PREDICTION
C
Y3=THEACC
RETURN
END
SUBROUTINE CREDAT

DESCRIPTION

THIS ROUTINE CREATES A DATA FILE AND WRITES RESULTS AND PARAMETERS FOR A PROGRAM RUN TO THE FILE

PARAMETERS INPUT

L1 : LENGTH OF DRIVE CRANK (IN.)
L2 : LENGTH OF DRIVEN CRANK (IN.)
L3 : LENGTH OF COUPLER (IN.)
L5 : LENGTH OF COUPLER SHANK (IN.)
LY : X DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT --LEFT POS.-- (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT --UPWARD POS.-- (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
SHORTL : AMOUNT DRIVEN CRANK IS SHORTENED FOR RETURN PHASE OF LEQ. (IN.)
Num1 : ARRAY SUBSCRIPT FOR STORING DRAWING VERSION OF LEG
BTHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG (RAD.)
Bgam : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEG (RAD.)
Bphi : ARRAY OF COUPLER ANGLES FOR DRAWING LEG (RAD.)
BLS : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEG (IN.)
Xmass : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE
Thetah : ANGLE LOCATION OF CENTER OF MASS OF DRIVE CRANK, CM FROM DRIVE CRANK. (RAD.)
@gam : ANGLE LOCATION OF CENTER OF MASS OF DRIVEN CRANK, CM FROM DRIVEN CRANK. (RAD.)
Phim : ANGLE LOCATION OF CENTER OF MASS OF COUPLER. (RAD.)
Lm1 : DISTANCE FROM DRIVE CRANK SUPPORT TO CENTER OF MASS OF DRIVE CRANK. (IN.)
Lm2 : DISTANCE FROM DRIVEN CRANK SUPPORT TO CENTER OF MASS OF DRIVEN CRANK. (IN.)
Lm3 : DISTANCE FROM COUPLER JOINT ON DRIVEN CRANK TO CENTER OF MASS OF COUPLER (IN.)
M1 : MASS OF DRIVE CRANK (LB. SEC2/IN.)
M2 : MASS OF DRIVEN CRANK (LB. SEC2/IN.)
M3 : MASS OF COUPLER (LB. SEC2/IN.)
Jc01 : MASS MOMENT OF INERTIA OF DRIVE CRANK (LB. SEC2/IN.)
Jc02 : MASS MOMENT OF INERTIA OF DRIVEN CRANK (LB. SEC2/IN.)
Jc03 : MASS MOMENT OF INERTIA OF COUPLER (LB. SEC2/IN.)
Ra : BEARING RADIUS AT JOINT A (DRY CRK SUP) (IN.)
Rb : BEARING RADIUS AT JOINT B (DRY CRK END) (IN.)
Rc : BEARING RADIUS AT JOINT C (DRYN CRK SUP) (IN.)
Rd : BEARING RADIUS AT JOINT D (DRVN CRK END) (IN.)
Mua : BEARING FRICTION COEFFICIENT AT JOINT A (IN.)
Mub : BEARING FRICTION COEFFICIENT AT JOINT B (IN.)
Muc : BEARING FRICTION COEFFICIENT AT JOINT C (IN.)
Mud : BEARING FRICTION COEFFICIENT AT JOINT D (IN.)
Fox : FORCE AT FOOT IN X-DIRECTION (RIGHT POS)
FOY : FORCE AT FOOT IN Y-DIRECTION (UP POS)
FOOTST : LENGTH OF LEG STRIDE (IN.)
EXPLAC : DISTANCE FROM DRIVE CRANK SUPPORT TO TOE-DOWN SUPPORT —POSITIVE RIGHT— (IN.)
SPEED : AVERAGE SPEED OF MACHINE (IN./SEC.)
INCRE : NUMBER OF INCREMENTS USED IN TIME STEPPING
SI : ANGLE FOR ACCELERATION OF GRAVITY
S : POSITIVE FOR DOWNHILL SLOPE (RAD.)
G : ACCELERATION OF GRAVITY (386 IN./SEC2)
PLACLQ : DISTANCE FROM FRAME EDGE TO DRIVEN CRANK SUPPORT (IN.)
SPAACLQ : DISTANCE BETWEEN LEGS (IN.)
BDWIDT : WIDTH OF FRAME (IN.)
BDLENT : LENGTH OF FRAME (IN.)
BDWEIT : WEIGHT OF FRAME (LBS.)

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

MDATE : VARIABLE FOR COMPUTING THE DATE
NAME : NAME OF DATA FILE BEING CREATED
TITL : TITLE OF PRESENT DATA FILE
RTOD : CONVERSION FACTOR (RADIANS TO DEGREES)
A : TEMPORARY STORAGE OF BTETHA
B : " " " BQAMMA
C : " " " BPHI
TIME : TIME
K : ARRAY SUBSCRIPT
J : DO LOOP CONTROL
I : DO LOOP CONTROL

EXTERNAL REFERENCES

DRWFMR, LINFD

CALLING ROUTINES

MAIN

******************************************************************************

DIMENSION BTHETA(BO), BQAMMA(BO), BPHI(BO), BL2(BO), XMASS(BO)

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMOM /B/ NUM1, BTHETA, BQAMMA, BPHI, BL2, XMASS
COMMOM /C/ L2, THETA, GAMMA, PHI
COMMOM /D/ THDOT1, QADOT1, PHDOT1
COMMOM /E/ THDOT2, QADOT2, PHDOT2
COMMOM /F/ COSH, SINH, COSQA, SINQA, COSPH, SINPH
COMMOM /G/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
* INCRE, INC1, DELTIM, SHORTL, BL2
COMMOM /H/ TTHETAH, GAMH, PHIM, LM1, LM2, LM3, JC01, JC02, JCG3,
* M1, M2, M3

226
COMMON /AA/ RA, RB, RC, RD, MUA, MUB, MUC, MUD, FOX, FOY, AX, AY, BI, Q
COMMON /BB/ PLAACLQ, SPAACLQ, BDWDIT, BDLENT, BDWEIT
COMMON /CC/ TITL(15)
C
REAL L1, L2, L3, L5, LX, LY, LM1, LM2, LM3, M1, M2, M3, JC01, JC02, JC03,
    MUA, MUB, MUC, MUD
CHARACTER*9 MDAT
BYTE NAME(12)
C
REWIND 7
C
RTOD=180./4./ATAN(1.)
PHIL=PHIL*RTOD
THETAM=THETAM*RTOD
GAMHAM=GAMHAM*RTOD
PHIM=PHIM*RTOD
BI=SI*RTOD
C
* CONVERT RADIANS TO DEGREES FOR PRINTING
C
LINE=1
CALL DWRFRM(IPAGE)
C
CALL LINFED(LINE, 2)
WRITE(5,310)
310 FORMAT(12X, 'CREATING DATA FILE')
C
* HEADING FOR CREATING DATA FILE
C
25 CALL LINFED(LINE, 2)
WRITE(5,102)
102 FORMAT(12X, 'ENTER OUTPUT FILE NAME: ', 5X)
READ(5,202, ERR=25) LENGTH, NAME(I), I=1, LENGTH
202 FORMAT(9,12A1)
OPEN(UNIT=7, NAME=NAME, TYPE='NEW', ERR=25)
C
* ASSIGN NAME TO AND
C
* OPEN FILE FOR CREATING DATA FILE
C
301 CALL LINFED(LINE, 7)
WRITE(5,300)
300 FORMAT(12X, 'INPUT THE TITLE FOR THIS CASE',/,
    12X, 'LESS THAN 60 CHAR. ',/,'T14', 'CASE TITLE:',
    12X, 'T15', ' ',/)
C
READ(5,200, ERR=301) (TITL(K), K=1, 15)
200 FORMAT(15A4)
C
* READ THE TITLE FOR THIS CASE
C
CALL DATE(MDATE)
C
* COMPUTE THE DATE
C
WRITE(7,105)
105 FORMAT(12X, 'THIS DATA FILE IS USED IN CONJUNCTION WITH',
    ' THE PROGRAM "DUNE" ',/,'T15', 'WHICH SIMULATES THE',
    12X, ' ',/)
227
$ ' ACTION OF A PASSIVE 4-BAR LINKAGE WALKING',//,
$ T15, 'MACHINE.', /////
C C
* PRINT THE DESCRIPTION OF THE DATA FILE
C C
WRITE(7, 100) (TITL(K), K=1, 15)
100 FORMAT(T5,'CASE TITLE: ',15A4,///)
C C
* PRINT OUT TITLE
C C
WRITE(7, 101) MDATE
101 FORMAT(T5,'DATE: ',1A9,///)
C C
* PRINT OUT DATE
C C
WRITE(7, 110) L1, L2, L3, LX, LY, L5
* T53, 'LX', T65, 'LY', T77, 'L5', //, T9, 6(1X, E11.4), ///)
C C
* PRINT THE GEOMETRY DATA
C C
WRITE(7, 115) LM1, LM2, LM3, THETAM, GAMMAM, PHIM
115 FORMAT(///, T5, 'LEG MASS LOCATION', //, T15, 'LM1', T26, 'LM2',
* T36, 'LM3', T51, 'THETAM', T63, 'GAMMAM', T76, 'PHIM', //,
* T9, 6(1X, E11.4), ///)
C C
* PRINT THE LEG MASS LOCATION DATA
C C
WRITE(7, 120) M1, M2, M3, JC01, JC02, JC03
120 FORMAT(///, T5, 'LEG MASS VALUES', //, T15, 'M1', T27, 'M2', T39, 'M3',
* T50, 'JC01', T62, 'JC02', T74, 'JC03', //, T9, 6(1X, E11.4), ///)
C C
* PRINT THE LEG MASS VALUE DATA
C C
WRITE(7, 125) MUA, MUB, MUC, MUD
125 FORMAT(///, T5, 'LINK BEARING COEFFICIENTS', //, T14, 'MUA', T26, 'MUB',
* T38, 'MUC', T50, 'MUD', //, T9, 4(1X, E11.4), ///)
C C
* PRINT LINK BEARING COEFFICIENTS
C C
WRITE(7, 130) RA, RB, RC, RD
130 FORMAT(///, T5, 'BEARING RADIi', //, T15, 'RA', T27, 'RB', T39, 'RC',
* T51, 'RD', //, T9, 4(1X, E11.4), ///)
C C
* PRINT BEARING RADIUSI
C C
WRITE(7, 135) PLACLO, SPACL0, BDWIT, BDLEN, BDWEIT
135 FORMAT(///, T5, 'FRAME PARAMETERS', //, T13, 'PLACLO', T25, 'SPACL0',
* T37, 'BDWIT', T49, 'BDLEN', T61, 'BDWEIT', //, T9,
* 5(1X, E11.4), ///)
C C
* PRINT FRAME PARAMETERS
C C
WRITE(7, 140) SHORTL, FOOTST, EXPLAC, SPEED
140 FORMAT(///, T5, 'STRIDE PARAMETERS', //, T13, 'SHORTL', T25, 'FOOTST',
* T37, 'EXPLAC', T49, 'SPEED', //, T9, 4(1X, E11.4), ///)
C
* PRINT STRIDE PARAMETERS

WRITE(7,145) PHIL,Q,SI,INCRE
145 FORMAT(//,T5, 'MISCELLANEOUS PARAMETERS',//,T14, 'PHIL',
     $     T25, 'GRAVITY', T35, 'GRAVITY ANG.', T49, 'INCRE',//,
     $     T9, 3(IX,E11.4), T50, I4, //)

* PRINT OUT MISCELLANEOUS PARAMETERS Q, SI, INCRE

PHIL=PHIL/RTOD
THETAM=THETAM/RTOD
GAMMAM=GAMMAM/RTOD
PHIM=PHIM/RTOD
SI=SI/RTOD

* CONVERT RADIANS BACK TO DEGREES

WRITE(7,150)
150 FORMAT(//,T5, 'TIME HISTORY OF PROBLEM ABOVE',//,T8, 'I',
     $     T15, 'TIME', T26, 'TTHETA', T38, 'T5AMMA', T51, 'T3HI',
     $     T62, 'BL2', T74, 'XMASS',//)

* PRINT THE HEADING FOR THE TIME HISTORY SOLUTION

DO 10 J=1,2
   DO 10 I=1,40
     K=I
     IF(J.EQ.2) K=I+40
     A=TTHETA(K)/RTOD
     B=T5AMMA(K)/RTOD
     C=T3HI(K)/RTOD
     TIME=(I-1)*DELTIM*INCI
     WRITE(7,155) K, TIME, A, B, C, BL2(K), XMASS(K)
   155 FORMAT(T6, I3, T10.6, (IX, E11.4))

    IF(J.EQ.2) GO TO 10
  WRITE(30, 500) TIME
  WRITE(40, 500) A
  WRITE(50, 500) B
    500 FORMAT(5X, F20.5)

* PRINT RESULTS TO DATA FILE FOR GRAPH

10 CONTINUE

* COMPUTE AND PRINT THE POSITION AS A FUNCTION

* OF TIME WITH ANGLES EXPRESSED IN DEGREES

CLOSE(UNIT=7)

* CLOSE DATA FILE

CALL LINPFD(LINE,1)
WRITE(5,315) (NAME(I), I=I,LENGTH)
315 FORMAT(12X, 'DATA IN FILE: ', 12A1)

* PRINT MESSAGE INDICATING FILE FORMED
C
302  WRITE(5, 320)
320  FORMAT(1X, 'HIT "CR" TO CONTINUE ', *),
     READ(5, 200, ERR=302)dummy
C
   * WAIT FOR "CR"
C
RETURN
END
SUBROUTINE DRWFRM

J-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DRAW THE SCREEN FRAME FOR ALL OUTPUT TO THE
SCREEN. THE TITLE IS OUTPUT IN THE UPPER LEFT BOX. THE
PAGE NO. IS WRITTEN IN THE LOWER LEFT CORNER.

PARAMETERS INPUT

NONE

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

IPAGE : PAGE NUMBER FOR SCREEN OUTPUT

EXTERNAL REFERENCES

PLOT10 ROUTINES

CALLING ROUTINES

MAIN, CHOEDM, CHPARAM, CHSTRD, CHBDR, CHFRM.

CALL SWINDO(0.4096, 0.3120)
CALL VWINDO(0., 0.4096, 0., 0.3120.)

SETS SCREEN COORDINATES

CALL NEWPAQ
IPAGE=IPAGE+1

CALL MOVEA(273., 235.)
CALL DRAWR(3600., 0.)
CALL DRAWR(0., 2500.)
CALL DRAWR(-3600., 0.)
CALL DRAWR(0., -2500.)

CALL MOVER(1100., 0.)
CALL DRAWR(0., 2500.)

CALL MOVER(0., -400.)
CALL DRAWR(-1100., 0.)

DRWRS RECTANGULAR FRAME
CALL MOVEA(455.,285.)
CALL CHRS1Z(4)
CALL ANMODE
WRITE(5,100) IPAGE

100 FORMAT(1H+, 'PAGE ', I2)

C
C OUTPUTS PAGE NO. TO LOWER LEFT OF PAGE.
C
CALL MOVEA(500.,2620.)
CALL CHRS1Z(2)
CALL ANMODE
WRITE(5,101)

101 FORMAT(1H+, 2X, 'D.U.W.E. ',/ ,12X, 'SIMULATION')

C
C WRITES HEADING IN BOX OF FRAME.
C
RETURN
END
SUBROUTINE DRAWLEG (I1, I2, X, X1, X2, IFR, XM, XR, YM, YR, IXM, IYM, IR)

******************************************************************************

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DRAWS THE DUNE VEHICLE AND SUPPORT TRIANGLE. CONTROL PARAMETERS ARE USED TO SUPPRESS OUTPUT SUCH THAT ONLY ONE LEG MAY BE DRAWN.

PARAMETERS INPUT

I1 : ARRAY SUBSCRIPT FOR CORRECT LEG POSITION
I2 : " " " " " "
X : VIRTUAL COORDINATE OF JOINT D OF FIRST LEG DRAWN
X1 : " " " " " SECOND " "
X2 : " " " " " THIRD " "
IFR : FLAG TO SUPPRESS DRAWING OF SECOND AND THIRD LEGS
XM : MINIMUM X VIRTUAL SCREEN COORD.
XR : RANGE OF X VIRTUAL SCREEN COORD.
YM : MINIMUM Y " " " "
YR : RANGE OF Y " " " "
IXM : MINIMUM X SCREEN COORD.
IYM : MINIMUM Y SCREEN COORD.
IR : RANGE X,Y SCREEN COORD.
L1 : LENGTH OF DRIVE CRANK
L2 : LENGTH OF DRIVEN CRANK
L3 : LENGTH OF COUPLER
L5 : LENGTH OF COUPLER SHANK
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT --LEFT POS.--
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT --UPWARD POS.--
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK
BTHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG
BPHI : ARRAY OF COUPLER ANGLES FOR DRAWING LEG
BL2 : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEG
IMASS : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE
PACLAL : DISTANCE FROM FRAME EDGE TO DRIVEN CRANK SUPPORT
BPAACL : DISTANCE BETWEEN LEGS
BDWIDT : WIDTH OF FRAME
BDLEN : LENGTH OF FRAME
BDWEIT : WEIGHT OF FRAME

PARAMETERS RETURNED

AX : X LOCATION OF DRIVE CRANK SUPPORT (VIRTUAL CORD)
AY : Y " " " " " " " "
BX : X LOCATION OF DRIVE CRANK END (VIRTUAL CORD)
BY : Y " " " " " " " "
CX : X LOCATION OF DRIVEN CRANK END (VIRTUAL CORD)
CY : Y " " " " " " " "
DX : X LOCATION OF DRIVEN CRANK SUPPORT (VIRTUAL CORD)
PROGRAM PAGE = 80

C       DY : Y     " " " " " " "
C       EX : X LOCATION OF FOOT POINT (VIRTUAL CORD)
C       EY : Y     " " " " " " 

PARAMETERS INTERNAL

C       FX : VIRTUAL COORDS. OF LOWER LEG POINT TO REPRESENT
C       FY : RIGID COUPLER
C       DX : VIRTUAL COORDS. OF JOINT D OF LEGS OF MACHINE
C       DX1 :
C       DX2 :
C       I3 : TEMPORARY ARRAY SUBSCRIPT FOR LEG DRAWING POSITION
C       I4 :
C       PXI : VIRTUAL COORDS. FOR DRAWING LEG MOUNTING PLATES.
C       PY1 :
C       PY2 :
C       PX3 :
C       PX4 :
C       FRY : VIRTUAL COORDINATES OF LEFT EDGE OF FRAME
C       FYX :
C       YPOS1 : TEMPORARY FOOT LOCATIONS FOR DRAWING SUPPORT
C       YPOS2 : QUADRATIC WHEN BOTH LEG SETS ARE IN CONTACT
C       YPOS9 : WITH GROUND
C       EX2 :
C       EX3 :
C       AAX :
C       BBX :
C       CXX :
C       EEX :
C       EEY :
C       EXY :
C       C0 : LOCATION OF C.O. OF MACHINE

EXTERNAL REFERENCES

C PLOT10 ROUTINES

C CALLING ROUTINES

C DRWTYP, CANIMA

*******************************************************************************

C
C DIMENSION BTHETA(80), BGAMMA(80), BPHI(80), BL2(80), XMASS(80)

C COMMON /A/ L1, L2, L3, LX, LY, PHIL
C COMMON /B/ NUM1, BTHETA, BGAMMA, BPHI, BL2, XMASS
C COMMON /BB/ PLACLQ, SPACLQ, BDWIDT, BDLENT, BDWEIT
C COMMON /CC/ AX, AY, BX, CY, DX, DY, EX, EY
C
C REAL L1, L2, L3, LX, LY
C
C DX=X
C DX1=X1
C DX2=X2

234
PROGRAM PAGE = B1

* SETS THE X-POSITION OF LEG FROM PARAMETERS PASSED
* TO ROUTINE.

DY=BL2(I2)*SIN(BOAMMA(I2))+L5*SIN(BPHI(I2)+PHIL)

* DETERMINES Y HEIGHT OF BODY

CALL SHINDO(IXM, IR, IYM, IR)
CALL VINDO(IXM, XR, YM, YR)

* SETS SCREEN AND VIRTUAL COORDINATES

IF(I1=2)16, 17, 17

* DECIDES WHICH LEGS TO DRAW AND WHERE (LEG PHASE)

17 DX1=X
   DX=X1
   QO TO 18

16 DX=X

* THE FIRST LEGS DX-POINT IS DRAWN AT X=PLACL0

18 I4=I2

* FOLLOWING DO LOOP DRAWS ONE FRAME OF WALKER

DD 13 I3=1, 3
   IF(I3=2)14, 15, 19
15 DX=DX1
   I4=I2+40
   QO TO 14
19 DX=DX2
   I4=I2
   IF(I1.EQ.2) I4=I2+40

* DECIDES X PLACEMENT OF LEGS

14 AX=DX+LX
   BX=AX+L1*COS(BTHETA(I4))
   CX=DX+BL2(I4)*COS(BOAMMA(I4))
   FX=CX+L5/4.*COS(BPHI(I4)+PHIL)
   EX=CX+L5*COS(BPHI(I4)+PHIL)
   AY=DX-LY
   BY=AY-L1*SIN(BTHETA(I4))
   CY=DX-BL2(I4)*SIN(BOAMMA(I4))
   FY=CY-L5/4.*SIN(BPHI(I4)+PHIL)
   EV=CY-L5*SIN(BPHI(I4)+PHIL)

* COMPUTES POINTS ON LEG LINKAGE REQUIRED FOR DRAWING.
* F POSITION CORRESPONDS TO POINT ON LOWER LEG MEMBER
* WHICH CAUSES DESIRED SLANT IN LEG DURING STRIDE.

CALL MOVEA(AX, AY)
CALL DRAWA(BX, BY)

235
CALL DRAWA(CX, CY)
CALL DRAWA(DX, DY)
CALL MOVEA(BX, BY)
CALL DRAWA(FX, FY)
CALL DRAWA(EX, EY)
CALL DRAWA(FX, FY)
CALL DRAWA(CX, CY)

* DRAWS ACTUAL LEGS

IF(IFR.EQ.5) RETURN

* RETURN AFTER DRAWING ONLY ONE LEG.
* ALLOWS THIS ROUTINE TO BE USED FOR DRAWING
* LEGS ON SCREEN FOR CHANGING PARAMETERS.

IF(IFR.NE.0) GO TO 12
CALL CIZAXIS(0)
CALL CHRSIZ(3)
CALL POINTA(EX, EY)
CALL CHRSIZ(4)
CALL CIZAXIS(2)

* PUTS POINT & FOOT POINT

12 S=DY/10.0
  PX1=DX-S
  PY1=DY+S
  PY2=PY1-S
  PX3=AX-S
  PY3=AY-B
  PX4=AX+S
  CALL MOVEA(PX1, PY1)
  CALL DRAWA(PX1, PY2)
  CALL DRAWA(PX3, PY3)
  CALL DRAWA(PX4, PY3)
  CALL DRAWA(PX4, PY1)

* DRAWS LEG MOUNTING PLATES

CALL CHRSIZ(4)
CALL CSIZE(IH, IV)
CALL MOVEA(AX, AY)
CALL MOVREL(-IH/3, -IV/4)
CALL ANCHD(111)
CALL MOVEA(DX, DY)
CALL MOVREL(-IH/3, -IV/4)
CALL ANCHD(111)

* DRAWS "O" AT BEARING POINTS.

FRY=DY+S
FRX=0.0
CALL MOVEA(FRX, FRY)
CALL DRAWR(BDLENT, 0.0)

* DRAWS SINGLE LINE (TOP OF THE BODY FRAME)
* BEGINNING X POSITION OF FRAME DRAWN AT X=0.0

13 CONTINUE

IF(IFR.EQ.2) RETURN

* DRAW DUWE WITHOUT SUPPORT TRIANGLE

AX=PLACL0+LX
BX=AX+L1*COS(BTHETA(I2))
CX=PLACL0+BL2(I2)*COS(BGAMMA(I2))
EX=CX+L5*COS(BPHI(I2)+PHIL)

* LOCATES X-POSITION OF FOOT IN CONTACT WITH GROUND TO DRAW SUPPORT TRIANGLE.

YPOS1=XM+XR-ABS(XM)
YPOS2=YPOS1-BDWIDT
YPOS3=YPOS1-BDWIDT/2.0

EX2=EX+SPACL0
EX3=EX+2*SPACL0

IF(I1-2) 20,21,21

* DECIDES WHICH PHASE THE TRIANGLE IS IN.

21 YPOS2=YPOS1
YPOS1=YPOS2-BDWIDT

20 IF(I2.NE.1) GO TO 25

AAAX=PLACL0+LX
BBX=AAAX+L1*COS(BTHETA(I2+40))
CCX=PLACL0+BL2(I2+40)*COS(BGAMMA(I2+40))
EEEX=CCX+L5*COS(BPHI(I2+40)+PHIL)
EEEX2=EEEX+SPACL0
EEEX3=EEEX+2*SPACL0
CALL MOVEA(EX,YPOS2)
CALL DRAWA(EX,YP0S2)
CALL DRAWA(EEEX,YP0S1)
CALL DRAWA(EEEX2,YP0S1)
CALL DRAWA(EEEX3,YP0S1)
GO TO 30

* DRAWS SUPPORT QUADRILATERAL AT INSTANT.
* ALL SIX LEGS ARE ON GROUND.

25 CALL MOVEA(EX,YP0S2)
CALL DRAWA(EX2,YP0S1)
CALL DRAWA(EX3,YP0S2)
CALL DRAWA(EX,YP0S2)

* DRAWS THE SUPPORT TRIANGLE

30 C0=XMASS(I4)+PLACL0
CALL MOVEA(C0,YP0S3)
CALL CSIZE(INH,IV)

237
CALL MOVREL(-IH/3,-IV/4)
CALL ANCHO(42)

C
CC  * OUTPUTS STAR AT CENTER OF MASS.

C
CALL HOME
RETURN
END
SUBROUTINE DRWYP (IFL, IFR, LINE)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE SETS THE PROPER PARAMETERS AND DIRECTS
PROGRAM FLOW FOR ANIMATION OF FRAME BY FRAME DRAWING OF LEG.

PARAMETERS INPUT

L1 : LENGTH OF DRIVE CRANK  (IN.)
L3 : LENGTH OF COUPLER       (IN.)
L5 : LENGTH OF COUPLER SHANK  (IN.)
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT --LEFT POS.--   (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT --UPWARD POS.-- (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
ETHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG (RAD.)
BGAMMA : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEG (RAD.)
BPHI : ARRAY OF COUPLER ANGLES FOR DRAWING LEG (RAD.)
BL2 : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEG (IN.)
XMASS : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE
BDWDIT : WIDTH OF FRAME    (IN.)
BDLENT : LENGTH OF FRAME   (IN.)

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

DY : VIRTUAL Y-COORD. OF JOINT D
DX : VIRTUAL COORDS. OF JOINT D OF LEGS OF MACHINE
DX1 : "    "   "    "   "    "   "    "
DX2 : "    "   "    "   "    "   "    "
HEIGHT : VIRTUAL HEIGHT FOR SCALING SCREEN
WIDTH : VIRTUAL WIDTH   "    "
SCALE : SCALE FOR VIRTUAL SCREEN COORDINATES
I1 : ARRAY SUBSCRIPT FOR LEG POSITION DRAWING
I2 : "    "   "    "   "    "   "    "
IL : INPUT FROM KEYBOARD CONTROLS FRAME BY FRAME DRAW
IM : MINIMUM X-VIRTUAL COORD.
XR : RANGE X-VIRTUAL COORD.

EXTERNAL REFERENCES

LINFED, DRWFMR, DRWLEG, CANIMA, PLOTIO Routines

CALLING ROUTINES

**************************************************************************
DIMENSION BTHETA(BO), BCGAMA(BO), BPFI(BO), BL2(BO), XMASS(BO)
COMMON /A/ L1, L2, L3, L5, LX, LY, PHIL
COMMON /B/ NUM1, BTHETA, BCGAMA, BPFI, BL2, XMASS
COMMON /BB/ PLACLG, SPACLG, BDWIOT, BDLENT, BDWEIT

REAL L1, L2, L3, L5, LX, LY

DY=BL2(1)*SIN(BCGAMA(1))+L5*SIN(BPFI(1)+PHIL)

* DETERMINES Y HEIGHT OF THE BODY FOR INITIAL POSITION

SCALE=1.0
HEIGHT=2.*(DY+BDWIOT)
WIDTH=3.0/2.0*BDLENT
IF (HEIGHT.GT. WIDTH) SCALE=HEIGHT/WIDTH
XM=-1.0/4.0*BDLENT*SCALE
XR=WIDTH*SCALE

* DETERMINES SCALING OF SCREEN COORDINATES
* DRAWING TO BE CENTERED WITH RESPECT TO X-AXIS

DX=PLACLG
DX1=SPACLG+PLACLG
DX2=2*SPACLG+PLACLG

* PARAMETERS REQUIRED FOR ROUTINE DRWLEG

IF (IFL.EQ.1) CALL DRWLEG(1, IFR, DX, DX1, DX2, 2, XM, XR, XM, XR, 1373, 310, 2500)

IF (IFL.EQ.1) RETURN

* IF IFL=1, RETURN AFTER DRAWING ONLY ONE FRAME

300 CALL LINFED(LINE, 4)
WRITE(5, 100)
100 FORMAT(12X, 'ANIMATE LEG [Y, N]? ', Y) READ(5, 102, ERR=300) ANS
102 FORMAT(A1)
IF (ANS.EQ. 'Y') GO TO 15

* WANT TO ANIMATE OR JUST DRAW THE LEG

DO 9 I=1, 4
CALL DRWFRM
LINE=1
CALL LINFED(LINE, 4)
WRITE(5, 101)
9 FORMAT(12X, 'HIT "C" FOR NEXT POSITION. ', Y, 12X, 'INPUT "S" TO STOP.')

* DRAW DUNE WITHOUT SUPPORT TRIANGLE

DO 10 I1=1, 2

240
DO 11 I2=1,40
    CALL DRWLEG(I1, I2, DX, DX1, DX2, 1, XM, XR, XM, XR, 1373, 310, 2500)
C
    CALL TINPUT(IL)
C
    IF (IL.EQ.83) GO TO 8
11    CONTINUE
10    CONTINUE
9    CONTINUE
C
    * REPEATEDLY DRAWS LEG
C
8    CALL LINFED(LINE, 1)
    WRITE(5,100)
    READ(5,102,ERR=8) ANS
    IF (ANS.EQ. 'Y') GO TO 15
C
    * ASKS TO ANIMATE LEG AFTER YOU HAVE DRAWN IT
C
    RETURN
C
15    CALL CANIMA(XM, XR)
C
    * CAUSES LEG TO BE ANIMATED
C
    RETURN
END
SUBROUTINE DUMOYL(IFK)

C******************************************************************************C
C 3-MAY-82 AUTHOR: FRANK BROWN
C
DESCRIPTION
C
THIS ROUTINE DRAWS THE MOST CURRENT VERSION OF THE DUNE.
THE FIRST TIME IT IS CALLED, IT DRAWS A SAMPLE DUNE.

PARAMETERS INPUT
C
L1 : LENGTH OF DRIVE CRANK (IN.)
L3 : LENGTH OF COUPLER (IN.)
L5 : LENGTH OF COUPLER SHANK (IN.)
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT --LEFT POS.-- (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT --UPWARD POS.-- (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
BTHTA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG (RAD.)
BGAMMA : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEG (RAD.)
BPHI : ARRAY OF COUPLER ANGLES FOR DRAWN LEG (RAD.)
BL2 : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEG (IN.)
XMASS : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE
PLACLQ : DISTANCE FROM FRAME EDGE TO DRIVEN
     CRANK SUPPORT (IN.)
SPACLQ : DISTANCE BETWEEN LEGS (IN.)
BDWIDT : WIDTH OF FRAME (IN.)
BDLENT : LENGTH OF FRAME (IN.)

PARAMETERS RETURNED
C
NONE

PARAMETERS INTERNAL
C
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

EXTERNAL REFERENCES
C
DRHTYP

CALLING Routines
C
MAIN, CHFRAM

******************************************************************************C
C
COMMON /A/ L1,L3,L5,LX,LY,PHIL
COMMON /B/ NUMI,BHTHA(80),BGAMMA(80),BPHI(80),
           BL2(80),XMASS(80)
COMMON /BB/ PLACLQ,SPACLQ,BDWIDT,BDLENT,BDWEIT
C
REAL L1,L2,L3,L5,LX,LY

242
PROGRAM PAGE = 89

C DTOR=4.0*ATAN(1.0)/180.
IF(IFK.EQ.0) GO TO 10

C * DRAW SAMPLE LEG THE FIRST TIME CALLED
C * ELSE DRAW CURRENT VERSION

C LINE=1
CALL DRWTYPE(1,1,LINE)
IFK=1
RETURN

C * DRAW CURRENT VERSION

C 10 BTHETA(1)=27.4*DTOR
   BGAAMA(1)=48.4*DTOR
   BPHI(1)=8.92*DTOR
   BL2(1)=8.224
   BTHETA(41)=110.6*DTOR
   BGAAMA(41)=106.7*DTOR
   BPHI(41)=33.5*DTOR
   BL2(41)=8.224
   L1=6.774
   L3=3.772
   L3=10.54
   LX=3.17
   LY=3.62
   PHIL=90.0*DTOR
   PLACL=12.41
   SPACL=22.
   BNDIT=36.
   BDLNT=72.

C * SAMPLE CONFIGURATION OF DUNE
C CALL DRWTYPE(1,1,LINE)
C
C * DRAW JUST THE DUNE OUTLINE ON SCREEN
C IFK=2
C
C * SET FLAG SO FURTHER DRAWS ARE FOR CURRENT VERSION
C RETURN
END
SUBROUTINE GRDLEG

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DETERMINES THE POSITION OF THE LEG IN CONTACT WITH THE GROUND. THE FOOT POSITION IS CALCULATED AND THEN THE CRANK ANGLE IS SOLVED BY ITERATION.

PARAMETERS INPUT

L2 : LENGTH OF DRIVEN CRANK (IN.)
INC1 : INCREMENT OF "INCRE" AT WHICH POSITION IS SAVED FOR DRAWING
DELTIM : INCREMENT OF TIME STEP (SEC.)

PARAMETERS RETURNED

NONE

PARAMETERS INTERNAL

THETA : DRIVE CRANK ANGLE MEASURED FROM HORIZON
GAMMA : DRIVEN CRANK ANGLE " " "
PHI : COUPLER LINK ANGLE " " "

EXTERNAL REFERENCES

LEGPOS

CALLING ROUTINES

ANALYS

DIMENSION BTHETA(80), B GAMMA(80), BPHI(80), BL2(80)
COMMON /B/ NUM1, BTHETA, BGAMMA, BPHI, BL2
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /G/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
          INCRE, INC1, DELTIM, SHORTL, SL2

REAL L2

ITERATION OF THE POSITION AND VELOCITY IS NECESSARY

STHETA=THETA
SSL2=L2

* SAVE VARIABLES FOR RETURN

L2=SL2

244
* FIX L2 TO EXTENDED LENGTH

DO 15 NUM1=1,40
TIME=(NUM1-1)*DELTIM*INC1

* COMPUTE TIME FOR LEG POSITION

CALL QRLEG1(TIME, ORTHET, ORTHED, ORTHDD)
THETA=ORTHET

* COMPUTE QRLEG THETA AT PRESENT TIME

CALL ANGLE

* COMPUTE REMAINING LINKAGE ANGLES

CALL LEPOSB(L2, THETA, GAMMA, PHI)

* STORE POSITIONS USING LEPODS

15 CONTINUE

THETA=STHETA
L2=SSL2
CALL ANGLE

* RETURN VALUES TO VALUES PREVIOUS TO CALL

RETURN
END
SUBROUTINE QRLGQ1(TIME, ORTHET, ORTHED, ORTHDD)

*********************************************************************************************

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE CALCULATES THE POSITION, VELOCITY, AND
ACCELERATION OF THE DRIVE CRANK BASED ON CONSTANT VELOCITY
AT THE FOOT.

PARAMETERS INPUT

L1  : LENGTH OF DRIVE CRANK  (IN.)
L2  : LENGTH OF DRIVEN CRANK  (IN.)
L3  : LENGTH OF COUPLER  (IN.)
L5  : LENGTH OF COUPLER SHAFT  (IN.)
LX  : X DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT —LEFT POS.—  (IN.)
LY  : Y DISTANCE FROM DRIVE CRANK SUPPORT TO
     DRIVEN CRANK SUPPORT —UPWARD POS.—  (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHAFT  (RAD.)
EXPICK : DISTANCE FROM DRIVE CRANK SUPPORT TO
         TOE-OFF POSITION —POSITIVE RIGHT—  (IN.)
SPEED : AVERAGE SPEED OF MACHINE  (IN./SEC.)

PARAMETERS RETURNED

THETA : DRIVE CRANK ANGLE MEASURED FROM POS. X
        : HORIZONTAL —CW—  (RAD.)
THDOT1 : ANGULAR VELOCITY OF DRIVE CRANK  (CW POS.)

PARAMETERS INTERNAL

COTH  : COSINE OF DRIVE CRANK ANGLE
COSGA : " " DRIVEN CRANK ANGLE
COSP : " " COUPLER
SIANGA : " " DRIVEN CRANK ANGLE
SIANGZ : " " COUPLER
COSPH : COSINE OF (PHI+PHIL)
SINPH : SINE OF "
THERG : BRACKETS ON THETA PASSED TO SECANT
THEG : " " "
PI  : PI

EXTERNAL REFERENCES

SECANT, ANGLE

CALLING Routines

ANALYSIS, PICKOF

*********************************************************************************************
PROGRAM PAGE = 93

C
C COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /D/ THDOT1, QADOT1, PHDOT1
COMMON /F/ COSTH, SINTH, COSGA, SINQA, COSPH, SINPH
COMMON /G/ PI, FOOTST, EXPLAC, EXICK, SPEED, PERIOD,
          INCRE, INC1, DELTIM, SHORTL, BL2
C
REAL L1, L2, L3, L5, LX, LY
C
STHETA=THETA
STHETD=THDOT1
STHETDO=THDOT2
SSL2=L2
C
C * STORE VALUES INPUT TO ROUTINE
C
C L2=BL2
C
C * FIX L2 IN THE EXTENDED POSITION
C
EX=EXPLAC*SPEED*TIME
PI=4*ATAN(1.0)
DELTHE=5.0/180.0*PI
THETA=0
THELOW=THETA
10 CALL ANGLE
COSPHA=COS(PHI+PHIL)
ERROR=-L2*COSGA-L5*COSPHA+LX+EX
IF(ERROR) 13, 12, 11
11 THEHIGH=THETA
THELOW=THELOW+DELTHE
GO TO 10
12 CONTINUE
C
C * COMPUTES THETA
C
CALL ANGLE
COSPHA=COS(PHI+PHIL)
SINPHA=SIN(PHI+PHIL)
C
C * COMPUTE GAMMA, PHI, AND COSINE AND SINE OF
C * FOOT ANGLE
C
THDOT1=SPEED*(L2*SINQA*L3*COSPH-L2*COSGA*L3*SINPH)
   *(L1*SINQA*(L1*SINTH*L3*COSPH-L1*COSTH*L3*SINPH)
   +L3*SINPHA*(L1*COSTH*L2*SINQA-L1*SINTH*L2*COSGA)))
C
C * COMPUTES THDOT1
C
CALL VELOC
A=L3*COSPH*L1*SINTH-L3*SINPH*L1*COSTH
B=L2*SINQA*L3*COSPH-L2*COSGA*L3*SINPH
\[ C = L_2 \cdot \sin \alpha \cdot L_1 \cdot \cos \theta - L_2 \cdot \cos \alpha \cdot L_1 \cdot \sin \theta \]
\[ D = L_3 \cdot \cos \phi \cdot (\text{THDDT1} + 2 \cdot \sin \theta - \text{QADOT1} + 2 \cdot \sin \phi) \]
\[ E = L_2 \cdot \sin \alpha \cdot (\text{THDDT1} + 2 \cdot \sin \theta + \text{QADOT1} + 2 \cdot \sin \phi) \]

\[ \text{THDDT2} = \left( \frac{\text{AX} - L_2 \cdot \sin \alpha \cdot D / B - L_2 \cdot \text{QADOT1} + 2 \cdot \cos \alpha - L_5 \cdot \sin \phi \cdot A}{E / B - L_5 \cdot \text{PHDDT1} + 2 \cdot \cos \phi} \right) / (L_2 \cdot \sin \alpha \cdot A / B + L_5 \cdot \sin \phi \cdot C / B) \]

- **COMPUTES THDDT2**

\[ \text{ORTHET} = \text{THETA} \]
\[ \text{ORTHED} = \text{THDDT1} \]
\[ \text{ORTHDD} = \text{THDDT2} \]

- **TRANSFERS VALUES TO BE RETURNED**

\[ L_2 = \text{SSL2} \]
\[ \text{THETA} = \text{STHETA} \]
\[ \text{THDDT1} = \text{STHETD} \]
\[ \text{THDDT2} = \text{STHEDD} \]

- **RETURN VALUES TO PREVIOUS VALUES BEFORE ROUTINE CALL**

CALL ANGLE
CALL VELOC
CALL ACCEL

- **PUT COMMON BLOCKS BACK PROPERLY**

RETURN

END
SUBROUTINE ORLEG2(TIME, ORTHET, ORTHED, ORTHDD)

C*******************************************************************************
C
C 3-MAY-82 AUTHOR: FRANK BROWN
C
DESCRIPTION
C
THIS ROUTINE USES SPLINE FIT DATA TO GET THE POSITION
VELOCITY AND ACCELERATION OF THE GROUND LEG POSITION.
The data must be stored in data files FOR010.DAT
and FOR020.DAT. THE SPLINE COEFFICIENTS ARE STORED
IN FOR010.DAT. THE TIME/POSITION HISTORY IS STORED
IN FOR020.DAT.

PARAMETERS INPUT

N : NUMBER OF ENTRIES IN DATA FILE TO BE READ
A : SPLINE COEFFICIENT OF CUBED TERM
B : " " SQUARED TERM
C : " " LINEAR TERM
D : " " CONSTANT TERM
X : VALUE OF TIME READ FROM DATA FILE
Y : VALUE OF THETA READ FROM DATA FILE
TIME : VALUE OF TIME INPUT TO ROUTINE

PARAMETERS RETURNED

ORTHET : VALUE OF THETA FROM ACTUAL EXPERIMENT TIME
ORTHED : VALUE OF THDOTT1 FROM ACTUAL EXPERIMENT TIME
ORTHDD : VALUE OF THDOTT2 FROM ACTUAL EXPERIMENT TIME

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING ROUTINES

ANALYS, PICKOF

*******************************************************************************

DIMENSION A(51), B(51), C(51), D(51), X(50), Y(50)

READ(10, *) N

DO 10 I=1, N
READ(10, *) A(I), B(I), C(I), D(I)
READ(20, *) X(I), Y(I)
10 CONTINUE
* READ THE SPLINE COEFFICIENTS AND THE X, Y VALUES
* FROM DATA FILES FOR010.DAT AND FOR020.DAT RESPECTIVELY

DO 20 I=1,N
  IF(TIME.GE.X(I).AND.TIME.LE.X(I+1)) GO TO 30
20 CONTINUE

* FIND CORRECT SPLINE FIT RANGE

WRITE(5,100)
100 FORMAT(12X,'TIME INTERVAL NOT FOUND IN SPLINE',/
     * 12X,'FIT VALUE SET TO LAST TIME INTERVAL.')

* ERROR MESSAGE IF PROPER TIME NOT FOUND
30 QRTHET=A(I)*(TIME-X(I))**3+B(I)*(TIME-X(I))**2*
    +C(I)*(TIME-X(I))+D(I)

* COMPUTE POSTION THETA
QRT Hed=3*A(I)*(TIME-X(I))**2+2*B(I)*(TIME-X(I))+C(I)

* COMPUTE VELOCITY THD T1
QRT HDD=6*A(I)*(TIME-X(I))+2*B(I)

* COMPUTE ACCELERATION T D T2

REWIND 10
REWIND 20

* REWIND THE DATA FILES

RETURN
END
SUBROUTINE LEGPOS(L2, THETA, GAMMA, PHI)

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE STORES THE POSITION OF 7TH LEG FOR DRAWING PURPOSES.

PARAMETERS INPUT

L2 : LENGTH OF DRIVEN CRANK (IN.)
THETA : DRIVE CRANK ANGLE MEASURED FROM POS. X (RAD.)
GAMMA : DRIVEN CRANK ANGLE MEASURED FROM POS. X (RAD.)
PHI : COUPLER ANGLE MEASURED FROM POS. X (RAD.)

PARAMETERS RETURNED

NUM1 : ARRAY SUBSCRIPT FOR STORING DRAWING VERSION OF LEG
BTHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG (RAD.)
BGAMMA : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEG (RAD.)
BPHI : ARRAY OF COUPLER ANGLES FOR DRAWING LEG (RAD.)
BL2 : ARRAY OF DRIVEN CRANK LENGTHS FOR DRAWING LEG (IN.)
XMASS : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

CNTMAS, DRWTYP

CALLING Routines

ANALYS, QRDLEG

******************************************************************************

DIMENSION BTHETA(80), BGAMMA(80), BPHI(80), BL2(80), XMASS(80)

COMMON /B/ NUM1, BTHETA, BGAMMA, BPHI, BL2, XMASS

REAL L2

BTHETA(NUM1)=THETA
BGAMMA(NUM1)=GAMMA
BPHI(NUM1)=PHI
BL2(NUM1)=L2

* STOR ES POSITION OF LEG FOR EASE OF DRAWING

251
CALL CNTMAS

* COMPUTES THE CENTER OF MASS OF BODY

IF(NUM1.LE.40) RETURN
CALL DRWTYP(1,NUM1-40)

* DRAW LEG AS COMPUTER GOES THROUGH ANALYSIS PHASE.

RETURN
END
PROGRAM PAGE = 99

SUBROUTINE LINEQ (A, B, IERR)

******************************************************************************

AUTHOR: GARY L. KINZEL

DESCRIPTION

THIS ROUTINE SOLVES THE MATRIX EQUATION AX + B = 0, OVERWRITING B WITH
THE SOLUTION MATRIX X. A MUST BE SQUARE AND NON-SINGULAR. B MUST
HAVE THE SAME NUMBER OF ROWS AS A. THE DETERMINANT OF A IS
COMPUTED. BOTH A AND B ARE DESTROYED.

THE METHOD CONSISTS OF GAUSSIAN ELIMINATION FOLLOWED BY BACK
SUBSTITUTIONS. THIS IS MORE EFFICIENT THAN SOLUTION BY MATRIX
INVERSION REGARDLESS OF THE NUMBER OF COLUMNS IN B. BOTH ROWS AND
COLUMNS ARE SEARCHED FOR MAXIMAL PIVOTS. INTERCHANGING OF ROWS OR
COLUMNS OF A IS AVOIDED. CHAPTER 1 OF E.L. STIEFLE, INTRODUCTION TO
NUMERICAL MATHEMATICS. ACADEMIC PRESS, N.Y., 1963, SHOULD BE HELPFUL
IN FOLLOWING THE CODE.

THE CALLING PROGRAM MUST SET A, N, AND B. SOME OF THE PARAMETERS
USED IN THE ROUTINE ARE-

A - THE COEFFICIENT MATRIX

N - THE ORDER OF A

B - THE CONSTANT TERM MATRIX

LTEMP - A BLOCK OF AT LEAST N WORDS OF TEMPORARY INTEGER STORAGE

IN ADDITION TO OVERWRITING B WITH THE SOLUTION MATRIX X, THE
ROUTINE SETS IERR, DET, NPIV, PIV, LPR, AND LPC TO

IERR - 2 IF NO COLUMNS OF X ARE FOUND, THE ELIMINATION PROCESS
BEING HALTED BECAUSE THE CURRENT PIVOT FAILS TO EXCEED
EPS IN MAGNITUDE

0 IF ALL COLUMNS OF X ARE FOUND, NO TROUBLE BEING DETECTED

LPR - THE FIRST NPIV POSITIONS LIST THE PIVOT ROW INDICES IN ORDER
OF USE. A VECTOR OF LENGTH N

LPC - THE FIRST NPIV POSITIONS LIST THE PIVOT COLUMN INDICES IN
ORDER OF USE. A VECTOR OF LENGTH N

DO INITIALIZATIONS

******************************************************************************

DIMENSION A(9,9), B(9,1)

DIMENSION LTEMP(9), LPR(9), LPC(9)

N=9

253
M=1
EPS=0.0
IERR=0
DO 1 I=1,N
LPR(I)=I
1 LPC(I)=I

BEGIN ELIMINATION PROCESS

DO 15 NP=1,N
NPV=NP

SELECT PIVOT

PIV=0.
DO 3 K=NP,N
I=LPR(K)
DO 3 L=NP,N
J=LPC(L)
IF(ABS(A(I,J))-ABS(PIV)) .GE. 2.2

2 KPIV=K
LPIV=L
IPIV=I
JPIV=J
PIV=A(I,J)
CONTINUE

UPDATE DETERMINANT AND PIVOT ROW AND COLUMN LISTS

ITEMP=LPR(NP)
LPR(NP)=LPR(KPIV)
LPR(KPIV)=ITEMP
ITEMP=LPC(NP)
LPC(NP)=LPC(LPIV)
LPC(LPIV)=ITEMP

EXIT IF PIVOT TOO SMALL

IF(EPS-ABS(PIV)) .LE. 5.4,4

4 IERR=2
RETURN

MODIFY PIVOT ROW OF A AND B (ELEMENTS IN PRESENT OR PREVIOUS PIVOT
COLUMNS OF A ARE SKIPPED)

5 IF(NP-N) .GE. 6,8,6
6 NN=NP+1
DO 7 L=NN,N
J=LPC(L)
7 A(IPIV,J)=A(IPIV,J)/PIV
8 DO 9 J=1,M
9 B(IPIV,J)=B(IPIV,J)/PIV

C
MODIFY NON-PIVOT ROWS OF A AND B (ELEMENTS IN PRESENT OR PREVIOUS
PIVOT ROWS OR COLUMNS ARE SKIPPED)
IF(NP-N) 10, 15, 10
10 DO 14 K=NNP, N
   I=LPR(K)
   TEMP=A(I, JPIV)
   IF(TEMP) 11, 14, 11
11 DO 12 L=NNP, N
   J=LPC(L)
12   A(I, J)=A(I, J)+A(IPIV, J)*TEMP
   DO 13 J=1, M
13   B(I, J)=B(I, J)+B(IPIV, J)*TEMP
14 CONTINUE
15 CONTINUE
C
C   END ELIMINATION PROCESS

255
SUBROUTINE LINFED(LINE, NLINE)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE INCREMENTS THE CURSOR POSITION FOR SCREEN WRITING.

PARAMETERS INPUT

LINE  : LINE SPACING PARAMETER FOR SCREEN WRITING
NLINE : NUMBER OF LINES WRITTEN WITH WRITE STATEMENT

PARAMETERS RETURNED

LINE

PARAMETERS INTERNAL

LL  : VALUE FOR SCREEN MOVE

EXTERNAL REFERENCES

PLOTIO ROUTINES

CALLING ROUTINES

MAIN, ANALYS, CHGEOM, CHBEAR, CHSTRD, CHPARA, CHFRAM, DRWTPY

CALL CHRSIZ(4)
LINE=LINE+16
LL=3070-LINE*48
 CALL MOVABS(0, LL)

LINE=LINE+NLINE-15
CALL ANMODE
RETURN
END
PROGRAM PAGE  =  103

SUBROUTINE OPTION(IDOPT,LINE)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DRAWS AND LABELS THE MENU.

PARAMETERS INPUT

NONE

PARAMETERS RETURNED

IDOPT : PARAMETER DESIGNATING SELECTION CHOSEN
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

LINFED, PLOT10 ROUTINES

CALLING ROUTINES

MAIN

IF(IDOPT.EQ.0) GO TO 5

* SKIP THE WRITING IF USER ENTERED WRONG SELECTION

LINE=1
CALL LINFED(LINE,1)
WRITE(5,100)
CALL LINFED(LINE,1)
WRITE(5,101)
CALL LINFED(LINE,1)
WRITE(5,102)
CALL LINFED(LINE,1)
WRITE(5,103)
CALL LINFED(LINE,1)
WRITE(5,104)
CALL LINFED(LINE,1)
WRITE(5,105)
CALL LINFED(LINE,1)
WRITE(5,106)
CALL LINFED(LINE,1)
WRITE(5,107)
CALL LINFED(LINE,5)

257
PROGRAM PAGE = 104

WRITE(5, 108)
CALL LINFD(LINE, 7)
WRITE(5, 109)

* WRITE MENU SELECTIONS TO SCREEN

100 FORMAT(12X, "[ ] RUN STORED VERSION?"
101 FORMAT(12X, "[ ] CHANGE LEG GEOMETRY?"
102 FORMAT(12X, "[ ] CHANGE LEG PARAMETERS?"
103 FORMAT(12X, "[ ] CHANGE BEARING PARAMETERS?"
104 FORMAT(12X, "[ ] CHANGE STRIDE PARAMETERS?"
105 FORMAT(12X, "[ ] CHANGE FRAME PARAMETERS?"
106 FORMAT(12X, "[ ] RUN NEW PARAMETERS?"
107 FORMAT(12X, "[ ] CREATE DATA FILE?"
108 FORMAT(12X, "[ ] EXIT?"
109 FORMAT(12X, 'ALIGN CURSOR IN BRACKETS FOR',/,
 * 12X, 'OPTION. HIT "SPACE" TO REGISTER. ')
110 FORMAT(12X, 'REALIGN CURSOR AND RE-REGISTER. ')

5 CALL SWINDD(0.4096, 0.3120)
   CALL SWINDD(0.640, 0.640)

* SET UP COORDINATES FOR CURSOR LOCATION

12 CALL VCURSR(IDUM, XCORD, YCORD)

IOPT=0
IF(YCORD .LE. 460. .AND. YCORD .GE. 450. ) IOPT=1
IF(YCORD .LE. 440. .AND. YCORD .GE. 430. ) IOPT=2
IF(YCORD .LE. 420. .AND. YCORD .GE. 410. ) IOPT=3
IF(YCORD .LE. 400. .AND. YCORD .GE. 390. ) IOPT=4
IF(YCORD .LE. 380. .AND. YCORD .GE. 370. ) IOPT=5
IF(YCORD .LE. 360. .AND. YCORD .GE. 350. ) IOPT=6
IF(YCORD .LE. 340. .AND. YCORD .GE. 330. ) IOPT=7
IF(YCORD .LE. 320. .AND. YCORD .GE. 310. ) IOPT=8
IF(YCORD .LE. 300. .AND. YCORD .GE. 290. ) IOPT=9

* SET IOPT

IF(IOPT .EQ. 0) GO TO 10
J=IOPT+2
CALL LINFD(JJ, 1)
WRITE(5, 111)

111 FORMAT(13X, '*')

* OUTPUT '*' IN SELECTED BOX

IF(IOPT .NE. 0) RETURN

* IF CURSOR NOT ALLIGNED IN BRACKET
* RE-INPUT WITH CURSOR.

10 LINE=LINE-4
CALL LINFD(LINE, -1)
LINE=LINE+4
WRITE(5, 110)
CALL BELL

258
GO TO 12
END
SUBROUTINE PARAM

C
C-------------------------------
C
C 3-MAY-82  AUTHOR: FRANK BROWN
C
C DESCRIPTION
C
C THIS ROUTINE STORES A SAMPLE VERSION OF ALL DUWE PARAMETERS
C FOR THE INITIAL PROGRAM.
C
C PARAMETERS INPUT
C
C  NONE
C
C PARAMETERS RETURNED
C
C L1 : LENGTH OF DRIVE CRANK (IN.)
C L2 : LENGTH OF DRIVEN CRANK (IN.)
C L3 : LENGTH OF COUPLER (IN.)
C L5 : LENGTH OF COUPLER SHANK (IN.)
C LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO
C : DRIVEN CRANK SUPPORT --LEFT POS.-- (IN.)
C LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO
C : DRIVEN CRANK SUPPORT --UPWARD POS.-- (IN.)
C PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
C SHORTL : AMOUNT DRIVEN CRANK IS SHORTENED FOR
C : RETURN PHASE OF LEQ. (IN.)
C THETA : DRIVE CRANK ANGLE MEASURED FROM POS. X
C : HORIZONTAL --CW-- (RAD.)
C GAMMA : DRIVEN CRANK ANGLE MEASURED FROM POS. X
C : HORIZONTAL --CW-- (RAD.)
C PHI : COUPLER ANGLE MEASURED FROM POS. X
C : HORIZONTAL --CW-- (RAD.)
C THDOT1 : ANGULAR VELOCITY OF DRIVE CRANK (CW POS.)
C QADOT1 : = " " DRIVEN CRANK (CW POS.)
C PHDOT1 : = " " COUPLER (CW POS.)
C THDOT2 : ANGULAR ACCELERATION OF DRIVE CRANK (CW POS.)
C QADOT2 : ANGULAR ACCELERATION OF DRIVEN CRANK (CW POS.)
C PHDOT2 : ANGULAR ACCELERATION OF COUPLER (CW POS.)
C THETAM : ANGLE LOCATION OF CENTER OF MASS OF DRIVE
C : CRANK. CW FROM DRIVE CRANK. (RAD.)
C Gammam : ANGLE LOCATION OF CENTER OF MASS OF DRIVEN
C : CRANK. CW FROM DRIVEN CRANK. (RAD.)
C Phim : ANGLE LOCATION OF CENTER OF MASS OF COUPLER.
C : CW FROM COUPLER CRANK. (RAD.)
C LM1 : DISTANCE FROM DRIVE CRANK SUPPORT TO CENTER
C : OF MASS OF DRIVE CRANK. (IN.)
C LM2 : DISTANCE FROM DRIVEN CRANK SUPPORT TO CENTER
C : OF MASS OF DRIVEN CRANK. (IN.)
C LM3 : DISTANCE FROM COUPLER JOINT ON DRIVEN CRANK
C : TO CENTER OF MASS OF COUPLER (IN.)
C M1 : MASS OF DRIVE CRANK (LBF.SEC2/IN.)
C M2 : MASS OF DRIVEN CRANK (LBF.SEC2/IN.)
C M3 : MASS OF COUPLER (LBF.SEC2/IN.)
C JG01 : MASS MOMENT OF INERTIA OF DRIVE CRANK (LBF.IN.SEC2.)
C JG02 : MASS MOMENT OF INERTIA OF DRIVEN CRANK (LBF.IN.SEC2.)

260
C  JCG3 : MASS MOMENT OF INERTIA OF COUPLER  (LBF-IN-SEC2.)
C  RA  : BEARING RADIUS AT JOINT A (DRV CRK SUP)  (IN.)
C  RB  : BEARING RADIUS AT JOINT B (DRV CRK END)   (IN.)
C  RC  : BEARING RADIUS AT JOINT C (DRVN CRK SUP)  (IN.)
C  RD  : BEARING RADIUS AT JOINT D (DRVN CRK END)  (IN.)
C  MUA : BEARING FRICTION COEFFICIENT AT JOINT A
C  MUB : BEARING FRICTION COEFFICIENT AT JOINT B
C  MUC : BEARING FRICTION COEFFICIENT AT JOINT C
C  MUD : BEARING FRICTION COEFFICIENT AT JOINT D
C  FGX : FORCE AT FOOT IN X-DIRECTION (RIGHT POS)
C  FGY : FORCE AT FOOT IN Y-DIRECTION (UP POS)
C  PI  : PI=4.0*ATAN(1.)
C
C  FOOTST : LENGTH OF LEG STRIDE  (IN.)
C  EXPLAC : DISTANCE FROM DRIVE CRANK SUPPORT TO
C           : TOE-DOWN POSITION —POSITIVE RIGHT—  (IN.)
C  EXPICK : DISTANCE FROM DRIVE CRANK SUPPORT TO
C           : TOE-OFF POSITION —POSITIVE RIGHT—  (IN.)
C  SPEED : AVERAGE SPEED OF MACHINE  (IN./SEC.)
C  PERIOD : TIME FOOT IS IN CONTACT WITH GROUND.
C           : =0.5*TOTAL LEG PERIOD  (SEC.)
C  INCRE : NUMBER OF INCREMENTS USED IN TIME STEPPING
C  INC1 : INCREMENT OF "INCRE" AT WHICH POSITION IS
C           : SAVED FOR DRAWING
C  DELTIM : INCREMENT OF TIME STEP  (SEC.)
C  AX  : ACCELERATION OF BODY IN X-DIRECTION  (IN./SEC2)
C  AY  : ACCELERATION OF BODY IN Y-DIRECTION  (IN./SEC2)
C  SI  : ANGLE FOR ACCELERATION OF GRAVITY  (RAD.)
C  G   : ACCELERATION OF GRAVITY  (386 IN./SEC2)
C  PLACLO : DISTANCE FROM FRAME EDGE TO DrIVEN
C           : CRANK SUPPORT  (IN.)
C  SPACLO : DISTANCE BETWEEN LEGS  (IN.)
C  BDWIDT : WIDTH OF FRAME  (IN.)
C  BDLENF : LENGTH OF FRAME  (IN.)
C  BDWEIT : WEIGHT OF FRAME  (LBS.)

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING ROUTINES

ANALYS

*******************************************************************************

COMMON /A/ L1, L3, L5, Lx, Ly, PHIL
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /D/ THDOT1, GADOT1, PHDOT1
COMMON /E/ THDOT2, GADOT2, PHDOT2
COMMON /G/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
           INCRE, INC1, DELTIM, SHORTL, SL2

261
COMMON /H/ THETAM, GAMMA, PHIM, LM1, LM2, LM3, JG1, JG2, JG3, M1, M2, M3
COMMON /AA/ RA, RB, RC, RD, MUA, MUB, MUC, MUD, FOX, FOY, AX, AY, SI, G
COMMON /BB/ PLACLQ, SPACLQ, BDWIDT, BDLENT, BDWEIT

* REAL L1, L2, L3, L5, LX, LY, LM1, LM2, LM3, M1, M2, M3, JG1, JG2, JG3, MUA, MUB, MUC, MUD

C DATA L1, L2, L3, L5, LX, LY/6. 774, 8. 224, 3. 772, 10. 539, 3. 17, 3. 62/

C DATA RA, RB, RC, RD/1., 1., 1., 1./
DATA AX, AY, SI, THETA, GAMMA, PHIM/0., 0., 0., 0., 0., 0., 0./
DATA FOX, FOY, PHIL/0., 0., 1., 571., 386./
C DATA LM1, LM2, LM3/1. 46, 2. 1, 2. 02/
DATA THETAM, GAMMA, PHIM/75., 63., 78.

C * DATA MUA, MUB, MUC, MUD/1., 1., 1., 1./
DATA SPACLQ, BDWIDT, BDLENT, BDWEIT/22., 36., 72., 15./
DATA THDOTT1, GADOT1, PHDOT1, THDOTT2, GADOT2, PHDOT2/

C * 0. 0, 0. 0, 0. 0, 0. 0, 0. 0, 0. 0/

C PI=4*ATAN(1. 0)
C FOOTST=12. 0
C EXPLAC=1. 055
C EXPK=EXPLAC-FOOTST

C SPEED=6. 0
C PERIOD=FOOTST/SPEED
C INCRE=80

C * INCRE MUST BE DIVISIBLE BY 40 AND LESS THAN OR EQUAL TO 80
C * THIS IS BECAUSE OF DRAWING ROUTINE.
C * THE DRAWING ROUTINE DRAWS ONLY 40 FRAMES PER
C * CYCLE/2.

C INC1=INCRE/40
C DELTIM=PERIOD/INCRE
C SHORTL=0. 1875
C
C PLACLQ=(BDLENT-2*SPACLQ-LX)/2. 0
C
THETAM=THETAM/180. 0*PI
GAMMA=GAMMA/180. 0*PI
PHIM=PHIM/180. 0*PI

C RETURN
C END

262
SUBROUTINE PICKOF(TIME1)

3-MAY-62 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE DETERMINES THE ANGLE "THETA" AND ANGULAR
ROTATION "THETA DOT" FOR A GIVEN (X-DISPLACEMENT) PICKOFF
POSITION OF THE FOOT AND A GIVEN FOOT VELOCITY (X-DIREC.)

PARAMETERS INPUT

PERIOD : TIME FOOT IS IN CONTACT WITH GROUND.
: = 0.5*TOTAL LEG PERIOD

PARAMETERS RETURNED

THETA : DRIVE CRANK ANGLE MEASURED FROM POS. X
: HORIZONTAL ---CW---

THDOT1 : ANGULAR VELOCITY OF DRIVE CRANK (CW POS.)

PARAMETERS INTERNAL

EXTERNAL REFERENCES

ANGLE, VELOC, ACCEL, PICKOF

CALLING ROUTINES

ANALYS

COMMON /A/ L1,L3,L5,LX,LY,PHIL
COMMON /C/ L2,THETA,GAMMA PHI
COMMON /D/ THDOT1,QADOT1,PHDOT1
COMMON /F/ COSTH,SINH,COSG,SGAI,COSPH,SPHI
COMMON /G/ PI,FOOTST,EXPLAC,EXPICK,SPEED,PERIOD,
* INCRE,INCI,DELTIM,SHORTL,SL2

REAL L1,L2,L3,L5,LX,LY

SSL2=L2
L2=SL2

* SAVE PARAMETERS FOR RETURN

CALL QRLEG1(TIME1, ORTHET, ORTHED, ORTHDD)
THETA=ORTHET
THDOT1=ORTHED

* COMPUTE THETA AND THDOT1
L2 = SSL2

C

* RETURN L2 TO ORIGINAL VALUE

C

RETURN

END
SUBROUTINE RDDAT(LINE)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE READS THE PARAMETER VALUES AND POSITION RESULTS FROM A DATA FILE.

PARAMETERS INPUT

NONE

PARAMETERS RETURNED

L1 : LENGTH OF DRIVE CRANK (IN.)
L2 : LENGTH OF DRIVEN CRANK (IN.)
L3 : LENGTH OF COUPLER (IN.)
L5 : LENGTH OF COUPLER SHANK (IN.)
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT ---LEFT POS.--- (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO DRIVEN CRANK SUPPORT ---UPWARD POS.--- (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
SHORTL : AMOUNT DRIVEN CRANK IS SHORTENED FOR RETURN PHASE OF LEG. (IN.)
NUM1 : ARRAY SUBSCRIPT FOR STORING DRAWING VERSION OF LEG
BTHETA : ARRAY OF DRIVE CRANK ANGLES FOR DRAWING LEG (RAD.)
BAMMA : ARRAY OF DRIVEN CRANK ANGLES FOR DRAWING LEG (RAD.)
BPHI : ARRAY OF COUPLER ANGLES FOR DRAWING LEG (RAD.)
BL2 : ARRAY OF DRIVE CRANK LENGTHS FOR DRAWING LEG (IN.)
XMASS : ARRAY OF LOCATIONS OF CENTER OF MASS OF MACHINE
THETAM : ANGLE LOCATION OF CENTER OF MASS OF DRIVE CRANK. CW FROM DRIVE CRANK. (RAD.)
GAMMAM : ANGLE LOCATION OF CENTER OF MASS OF DRIVEN CRANK. CW FROM DRIVEN CRANK. (RAD.)
PHIM : ANGLE LOCATION OF CENTER OF MASS OF COUPLER. CW FROM COUPLER CRANK. (RAD.)
LM1 : DISTANCE FROM DRIVE CRANK SUPPORT TO CENTER OF MASS OF DRIVE CRANK. (IN.)
LM2 : DISTANCE FROM DRIVEN CRANK SUPPORT TO CENTER OF MASS OF DRIVEN CRANK. (IN.)
LM3 : DISTANCE FROM COUPLER JOINT ON DRIVEN CRANK TO CENTER OF MASS OF COUPLER (IN.)
M1 : MASS OF DRIVE CRANK (LBF.SEC2/IN.)
M2 : MASS OF DRIVEN CRANK (LBF.SEC2/IN.)
M3 : MASS OF COUPLER (LBF.SEC2/IN.)
JCQ1 : MASS MOMENT OF INERTIA OF DRIVE CRANK (LBF. IN. SEC2.)
JCQ2 : MASS MOMENT OF INERTIA OF DRIVEN CRANK (LBF. IN. SEC2.)
JCQ3 : MASS MOMENT OF INERTIA OF COUPLER (LBF. IN. SEC2.)
RA : BEARING RADIUS AT JOINT A (DRV CRK SUP) (IN.)
RB : BEARING RADIUS AT JOINT B (DRV CRK END) (IN.)
RC : BEARING RADIUS AT JOINT C (DRV CRK SUP) (IN.)
RD : BEARING RADIUS AT JOINT D (DRV CRK END) (IN.)
MUA : BEARING FRICTION COEFFICIENT AT JOINT A
C MUB : BEARING FRICTION COEFFICIENT AT JOINT B
C MUC : BEARING FRICTION COEFFICIENT AT JOINT C
C MUD : BEARING FRICTION COEFFICIENT AT JOINT D
C FGX : FORCE AT FOOT IN X-DIRECTION (RIGHT POS)
C FGY : FORCE AT FOOT IN Y-DIRECTION (UP POS)
C PI : PI=4.0*ATAN(1.)
C FOOTST : LENGTH OF LEG STRIDE (IN.)
C EXPLAC : DISTANCE FROM DRIVE CRANK SUPPORT TO TOE-DOWN POSITION --POSITIVE RIGHT-- (IN.)
C EXPICK : DISTANCE FROM DRIVE CRANK SUPPORT TO TOE-OFF POSITION --POSITIVE RIGHT-- (IN.)
C SPEED : SPEED OF MACHINE (IN./SEC.)
C PERIOD : TIME FOOT IS IN CONTACT WITH GROUND. =0.5*TOTAL LEG PERIOD (SEC.)
C INCRE : NUMBER OF INCREMENTS USED IN TIME STEPPING
C INC1 : INCREMENT OF "INCRE" AT WHICH POSITION IS SAVED FOR DRAWING (SEC.)
C DELTIM : INCREMENT OF TIME STEP (SEC.)
C AX : ACCELERATION OF BODY IN X-DIRECTION (IN./SEC^2)
C AY : ACCELERATION OF BODY IN Y-DIRECTION (IN./SEC^2)
C BI : ANGLE FOR ACCELERATION OF GRAVITY (RAD.)
C O : ACCELERATION OF GRAVITY (386 IN./SEC^2)
C PLAACL : DISTANCE FROM FRAME EDGE TO DRIVEN CRANK SUPPORT (IN.)
C SPPACL : DISTANCE BETWEEN LEGS (IN.)
C BDWIDT : WIDTH OF FRAME (IN.)
C BDLEN T : LENGTH OF FRAME (IN.)
C BDWEIT : WEIGHT OF FRAME (LBS.)
C PARAMETERS INTERNAL
C NAME : NAME OF DATA FILE
C DTR : TRANSFORMATION FROM DEGREES TO RADIANS
C EXTERNAL REFERENCES
C NONE
C CALLING ROUTINES
C MAIN
C
C*********************************************
C
C DIMENSION BTHETA(80), BGAMMA(80), BPHI(80), BL2(80), XMASS(80)
C
C COMMON /A/ L1, L3, L5, LX, LY, PHIL
C COMMON /B/ NUM1, BTHETA, BGAMMA, BPHI, BL2, XMASS
C COMMON /C/ L2, THETA, GAMMA, PHI
C COMMON /D/ THDT1, QADOT1, PHDOT1
C COMMON /E/ THDT2, QADOT2, PHDOT2
C COMMON /F/ COTH, BINTH, COSGA, SINGA, COSPH, SINPH
C COMMON /G/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
C INC1, INC0, DELTIM, SHORTL
C COMMON /H/ THETAM, GAMMA1, PHIM, LM1, LM2, LM3, JGQ1, JGQ2, JGQ3,
PROGRAM PAGE = 113

* M1, M2, M3
COMMON /AA/ RA, RB, RC, RD, MUA, MUB, MUC, MUD, FGX, FOY, AX, AY, SI, Q
COMMON /BB/ PLACLG, SPACLG, BDWIDT, BDLENT, BDWEIT
COMMON /CC/ TITL, DATE
C
REAL L1, L2, L3, L5, LX, LY, LM1, LM2, LM3, M1, M2, M3, JG01, JG02, JG03,
   MUA, MUB, MUC, MUD
C
BYTE NAME(12)
C
REWIND 7
C
   * MAKE SURE DATA FILE IS AT THE TOP
C
LINE=1
CALL DRWFRM(IPAGE)
C
CALL LINFED(LINE, 2)
WRITE(5, 100)
100 FORMAT(12X, 'READING DATA FILE')
C
   * READING FOR READING DATA FILE
C
15 CALL LINFED(LINE, 4)
WRITE(5, 105)
105 FORMAT(12X, 'ENTER INPUT FILE NAME: ', 9)
READ(5, 201, ERR=15) LENGTH, (NAME(I), I=1, LENGTH)
201 FORMAT(12A1)
OPEN(UNIT=7, NAME=NAME, TYPE='OLD', ERR=15, READONLY)
C
   * INSERT THE NAME OF THE FILE READING FROM
   * AND OPEN UNIT 7 AS THE FILE TO READ FROM
C
READ(7, 200, ERR=300)
200 FORMAT(/////////////////////////////)
PROGRAM PAGE = 114

* READ BEARING RADII

READ(7, 215, ERR=300) PLACLQ, SPACLQ, BDWIDT, BDLEN, BDWEIT
215 FORMAT(T9, 5(1X, E11.4), //////////)

* READ FRAME PARAMETERS

READ(7, 210, ERR=300) BHORTL, FOOTST, EXPLAC, SPEED

* READ STRIDE PARAMETERS

READ(7, 220, ERR=300) PHIL, Q, SI, INCRE
220 FORMAT(T9, 3(1X, E11.4), T50, I4, //////////)

* READ MISCELLANEOUS DATA

THETAM = THETAM * DTOR
GAMMA = GAMMA * DTOR
PHIM = PHIM * DTOR
PHIL = PHIL * DTOR
SI = SI * DTOR

* CONVERT DEGREES TO RADIANS

EXPICK = EXPLAC - FOOTST
PERIOD = FOOTST / SPEED
INCI = INCRE / 40
DELTIM = PERIOD / FLOAT(INCRE)

* COMPUTE PARAMETERS FOR TIMING

DO 10 I = 1, 80
READ(7, 230, ERR=300) K, TIME, A, B, C, BL2(I), XMASS(I)
230 FORMAT(T6, 13, T10.6(1X, E11.4))
BTHETA(I) = A * DTOR
BGAMMA(I) = B * DTOR
BPHI(I) = C * DTOR
10 CONTINUE

* READ AND CONVERT TO RADIANS THE SOLUTION

* OF PROBLEM BEING READ

CLOSE(UNIT=7)

* CLOSE FILE 7 BEFORE EXITING READING ROUTINE

RETURN

CALL LINFED(LINE, 2)
WRITE(5, 301) (NAME(I), I = 1, LENGTH)
301 FORMAT(12X, 'ERROR READING FROM DATA', 12X, 'FILE: ', 12AI)
* REWIND 7
GO TO 19

268
C
C       * IF ERROR IN DATA FILE RE-INPUT DATA FILE
C
END
SUBROUTINE RKSTEP(H, T, Y1, Y2, LINE)

C*************************************************************************
C
3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE PERFORMS A 4TH-ORDER RUNGE KUTTA SOLUTION, ONE
STEP IN TIME.

PARAMETERS INPUT

H : WIDTH OF TIME STEP (SEC.)
T : TIME (SEC.)
Y1 : VALUE OF THETA PASSED TO ROUTINE
Y2 : VALUE OF THDOT1 PASSED TO ROUTINE
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

PARAMETERS RETURNED

Y1, Y2

PARAMETERS INTERNAL

K1 : RUNGE KUTTA COEFFICIENTS
K2
K3
K4
Y1HOLD : TEMP STORAGE OF Y1 AND Y2
Y2HOLD
T2 : INTERMEDIATE TIME STEP
T3
Y11 : INTERMEDIATE VALUES OF Y1 AND Y2
Y21
Y12
Y22
Y13
Y23

EXTERNAL REFERENCES

F

CALLING Routines

ANALYS, TOEDOWN

C*************************************************************************
C
DIMENSION K1(2), K2(2), K3(2), K4(2)
REAL K1, K2, K3, K4
C
Y1HOLD=Y1
Y2HOLD=Y2
C
270
C  Hold values for future calculations
C
K1(1)=H*F(1, T, Y1, Y2)
K1(2)=H*F(2, T, Y1, Y2)

T2=T+H/2.0
Y11=Y1+K1(1)/2.0
Y21=Y2+K1(2)/2.0
K2(1)=H*F(1, T2, Y11, Y21)
K2(2)=H*F(2, T2, Y11, Y21)

Y1=Y1HOLD
Y2=Y2HOLD
Y12=Y1+K2(1)/2.0
Y22=Y2+K2(2)/2.0
K3(1)=H*F(1, T2, Y12, Y22)
K3(2)=H*F(2, T2, Y12, Y22)

Y1=Y1HOLD
Y2=Y2HOLD
T3=T+H
Y13=Y1+K3(1)
Y23=Y2+K3(2)
K4(1)=H*F(1, T3, Y13, Y23)
K4(2)=H*F(2, T3, Y13, Y23)

Y1=Y1HOLD+1.0/6.0*(K1(1)+2*K2(1)+2*K3(1)+K4(1))
Y2=Y2HOLD+1.0/6.0*(K1(2)+2*K2(2)+2*K3(2)+K4(2))

C  Compute new values
C
T=T3
C
RETURN
END
PROGRAM PAGE = 118

FUNCTION F(I, TT, YY1, YY2)

C*****************************************************************************
C 3_MAY_82 AUTHOR: FRANK BROWN
C DESCRIPTION
C THIS ROUTINE IS USED IN CONJUNCTION WITH RKSTEP. IT PROVIDES
C THE FUNCTION VALUES FOR THE DIFFERENTIAL EQUATIONS.
C
PARAMETERS INPUT
I :FLAG DESIGNATED PROPER FUNCTION EVALUATION
TT :TIME
YY1 :VALUE OF THETA PASSED TO ROUTINE
YY2 :VALUE OF THDOT1 PASSED TO ROUTINE

PARAMETERS RETURNED
F :VALUE OF THDOT2 RETURNED TO RKSTEP

PARAMETERS INTERNAL
NONE

EXTERNAL REFERENCES
COMACC

CALLING Routines
RKSTEP

*****************************************************************************

COMMON /E/ THDOT2, QADOT2, PHDOT2

IF(I.EQ.2) GO TO 10
F=YY2
RETURN

10 CALL COMACC(TT, YY1, YY2, YY3, LINE)
F=YY3
IF(TT.EQ.0.) F=THDOT2

* IF TIME=0. THEN F=THDOT2 OF TOEOFF POSITION

RETURN
END
SUBROUTINE SECANT(X1,X2,X,EX)

3-MAY-82 AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE USES THE SECANT METHOD TO
FIND THE ROOTS OF AN ALGEBRAIC EQUATION OF
THE FORM F(X)=0.

PARAMETERS INPUT

X1 : VALUE OF THETA FOR WHICH ROOT FUNCTION IS NEGATIVE
X2 : " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " 

PARAMETERS RETURNED

X : VALUE OF THETA FOR WHICH ROOT FUNCTION IS ZERO

PARAMETERS INTERNAL

E : ERROR TOLERANCE ON ROOT FUNCTION
MAXF : MAXIMUM NUMBER OF ITERATIONS ALLOWED
I : ITERATION COUNTER
XMINUS : TEMPORARY STORAGE OF "X"
XSTOR : =
XMIN : =
XMAX : =
XPLUS : =

EXTERNAL REFERENCES

ERRORF

CALLING ROUTINES

PIECOF

X1 < X2 ARE DEFINED AS THE INTERVAL WHICH
BOUNDS THE ROOT WHERE F(X1) < 0.0 AND
F(X2) > 0.0.
E IS DEFINED AS THE MAX. ALLOWABLE ERROR
BETWEEN SUCCESSIVE ITERATIONS FOR CONVERGENCE.
MAXF IS DEFINED AS THE MAXIMUM NUMBER OF ITERATIONS
ALLOWED.
THE EQUATION F(X) IS ENTERED INTO THE ROUTINE THROUGH
THE FUNCTION SUBPROGRAM ERRORF(X,EX).
E=1.0E-4
MAXF=15

273
PROGRAM PAGE = 120

C
C
I=0
XMINUS=X1
X=X2
XSTOR=X
XMIN=XMINUS
XMAX=X

C
C
\* Initializes variables. \*
C
5 XPLUS=X-(ERRORF(X,EX)*XMINUS-XMINUS)/
* (ERRORF(X,EX)-ERRORF(XMINUS,EX))
C
C
\* Computes next iteration of x. \*
C
IF XPLUS LE XMIN OR XPLUS GE XMAX XPLUS=(XMIN+XMAX)/2.0
C
C
\* Uses interval halving step should the new iteration
\* occur outside the designated interval. \*
C
15 IF ERRORF(XPLUS,EX)*ERRORF(X2,EX) 15,25,20
20 IF XPLUS GE XMIN XMIN=XPLUS
GO TO 25
20 IF XPLUS LE XMAX XMAX=XPLUS
C
C
\* Keeps the interval size minimal. \*
C
25 I=I+1
XMINUS=XSTOR
X=XPLUS
XSTOR=XPLUS
C
C
\* Test=ABS(XMINUS-X) \*
C
\* IF(ABS(XMINUS-X),LE,E)GO TO 45 \*
\* Checks for convergence \*
C
IF(I,GT,MAXF)GO TO 40
GO TO 5
40 RETURN
C
C
45 RETURN
END
FUNCTION ERRORF(X,EX)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS FUNCTION SUBPROGRAM DEFINES THE EQUATION FOR
THE SECANT METHOD.

PARAMETERS INPUT

L2    : LENGTH OF DRIVEN CRANK
L1    : LENGTH OF SHANK
PHI   : COUPLER ANGLE MEASURED FROM HORIZONTAL
PHIL  : ANGLE BETWEEN COUPLER AND SHANK
COSQA : COS(QAMMA)
COSPHA : COS(PHI+PHIL)

PARAMETERS RETURNED

ERRORF : VALUE OF ROOT FUNCTION RETURNED TO SECANT

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING Routines

SECANT

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /C/ L2, THETA, QAMMA, PHI
COMMON /F/ COTH, SINTH, COSQA, SINQA, COSPH, SINPH

REAL L1, L2, L3, L5, LX, LY

THETA=X

CALL ANGLE

COSPHA=COS(PHI+PHIL)
SINPHA=SIN(PHI+PHIL)
ERRORF=-L2*COSQA-L5*COSPHA+LX*EX
RETURN
END
SUBROUTINE TOEDWN(THETOE, THEBRA, THETIM, LINE)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE COMPUTES THE BRAKING HEIGHT OF THE LEG
REQUIRED FOR A SMOOTH TRANSITION OF POSITION AND
VELOCITY AT THE TOEDOWN POSITION. IT ALSO CALLS PICKOF
FOR THE CALCULATION OF THE POSITION AND VELOCITY AT TOEDOWN

THE PROCEDURE USED IS TO SPECIFY THE VELOCITY REQUIRED
AT TOE-DOWN AND SOLVE THE PROBLEM IN REVERSE UNTIL THE
LEG VELOCITY DROPS TO ZERO. THIS IS THE BRAKE POSITION.

PARAMETERS INPUT

EXPLAC : DISTANCE FROM DRIVE CRANK SUPPORT TO
         : TOE-DOWN POSITION —POSITIVE RIGHT— (IN.)
SPEED : SPEED OF MACHINE (IN./SEC.)
DELTIM : INCREMENT OF TIME STEP (SEC.)
LINE : LINE SPACING PARAMETER FOR SCREEN WRITES

PARAMETERS RETURNED

THETOE : DRIVE CRANK POSITION AT TOE-DOWN (RAD.)
THEBRA : DRIVE CRANK POSITION AT BRAKE POSITION (RAD.)
THETIM : TIME REQUIRED FOR LEG TO SWING FROM BRAKE
         : POSITION TO TOE-DOWN POSITION (SEC.)

PARAMETERS INTERNAL

THDOT1 : ANGULAR VELOCITY OF DRIVE CRANK (CW POS.)
THDSAV : TEMPORARY STORAGE OF THDOT1
THDHAL : TEMPORARY STORAGE OF THDOT1
TIME : TIME NUMBER : NUMBER OF TIME STEP

EXTERNAL REFERENCES

PICKOF, RKSTEP, LINFED

CALLING Routines

ANALYS

COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /D/ THDOT1, GADOT1, PHDOT1
COMMON /G/ PI, FOOTST, EXPLAC, EXPICK, SPEED, PERIOD,
         * INCRE, INC1, DELTIM, SHORTL, SL2

C

DTOR=4.0*ATAN(1.0)/180.

276
PROGRAM PAGE = 123

TIME = 0.
CALL PICKOF(TIME)

IFL = 0
THETOE = THETA
THDSAVE = THDOTT1
THDOTT1 = THDOTT1
THDHOI = THDOTT1

* COMPUTES THETA AND THDOTT1 AT TOEDOWN POSITION
* FOR THE REQUIRED SPEED AND FOOT STRIDE

11 TIME = 0.0
NUMBER = -1

4 NUMBER = NUMBER + 1
IF(NUMBER .GE. INCRE) GO TO 20
TIME = DELTIM * NUMBER / 4.0

CALL RKSTEP(DELTIM, TIME, THETA, THDOTT1, LINE)

* ADVANCES SOLUTION OF DIFF. EQU. 1 TIME STEP

IF(THDHOI - THDOTT1) 2, 2, 4

* IF LEG VELOCITY CHANGES SIGN THEN BRAKING HEIGHT IS REACHED

2 THETOIM = TIME
THEBRA = THETA
THETA = THETOE
THDOTT1 = THDSAVE
RETURN

* RETURN THE VALUES COMPUTED WITHIN THE PROGRAM

20 THEBRA = THETOE
THETOIM = 0.0
CALL LINFEQ(LINE, 3)
WRITE(5, 101)
101 FORMAT(12X, 'ITERATION OF THEBRA DOES NOT CONVERGE',
* 12X, 'WITHIN REASONABLE TOLERANCE. ANALYSIS',
* 12X, 'CONT. WITH THEBRA = THETOE')

* ERROR MESSAGE IF ROUTINE FAILS TO CONVERGE

RETURN
END
SUBROUTINE VALUES(A,B)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE ASSIGN THE PROPER VALUES TO THE 9x9 MATRIX EQUATION FOR COMPATIBILITY WITH ROUTINE LINEQ.

PARAMETERS INPUT

L1 : LENGTH OF DRIVE CRANK (IN.)
L2 : LENGTH OF DRIVEN CRANK (IN.)
L3 : LENGTH OF COUPLER (IN.)
L5 : LENGTH OF COUPLER SHANK (IN.)
LX : X DISTANCE FROM DRIVE CRANK SUPPORT TO (IN.)
LY : Y DISTANCE FROM DRIVE CRANK SUPPORT TO (IN.)
PHIL : ANGLE BETWEEN COUPLER AND COUPLER SHANK (RAD.)
THETA : DRIVE CRANK ANGLE MEASURED FROM POS. X (RAD.)
GAMMA : DRIVE CRANK ANGLE MEASURED FROM POS. X (RAD.)
PHI : COUPLER ANGLE MEASURED FROM POS. X (RAD.)
THD1 : ANGULAR VELOCITY OF DRIVE CRANK (CW POS.)
QADD1 : " " " DRIVEN CRANK (CW POS.)
PHD1 : " " " COUPLER (CW POS.)
THETAM : ANGLE LOCATION OF CENTER OF MASS OF DRIVE CRANK, CW FROM DRIVE CRANK. (RAD.)
GAMMAM : ANGLE LOCATION OF CENTER OF MASS OF DRIVEN CRANK. CW FROM DRIVEN CRANK. (RAD.)
PHIM : ANGLE LOCATION OF CENTER OF MASS OF COUPLER. CW FROM COUPLER CRANK. (RAD.)
LM1 : DISTANCE FROM DRIVE CRANK SUPPORT TO CENTER OF MASS OF DRIVE CRANK. (IN.)
LM2 : DISTANCE FROM DRIVEN CRANK SUPPORT TO CENTER OF MASS OF DRIVEN CRANK. (IN.)
LM3 : DISTANCE FROM COUPLER JOINT ON DRIVEN CRANK TO CENTER OF MASS OF COUPLER (IN.)
M1 : MASS OF DRIVE CRANK. (LBFRSEC2/IN.)
M2 : MASS OF DRIVEN CRANK (LBFRSEC2/IN.)
M3 : MASS OF COUPLER (LBFRSEC2/IN.)
JCG1 : MASS MOMENT OF INERTIA OF DRIVE CRANK (LBFRSEC2/IN.)
JCG2 : MASS MOMENT OF INERTIA OF DRIVEN CRANK (LBFRSEC2/IN.)
JCG3 : MASS MOMENT OF INERTIA OF COUPLER (LBFRSEC2/IN.)
RA : BEARING RADIUS AT JOINT A (DRV CRK BUP) (IN.)
RB : BEARING RADIUS AT JOINT B (DRV CRK END) (IN.)
RC : BEARING RADIUS AT JOINT C (DRV CRK BUP) (IN.)
RD : BEARING RADIUS AT JOINT D (DRV CRK END) (IN.)
MUA : BEARING FRICTION COEFFICIENT AT JOINT A (IN.)
MUB : BEARING FRICTION COEFFICIENT AT JOINT B (IN.)
MUC : BEARING FRICTION COEFFICIENT AT JOINT C (IN.)
MUD : BEARING FRICTION COEFFICIENT AT JOINT D (IN.)
PROGRAM PAGE = 125

C       FOX : FORCE AT FOOT IN X-DIRECTION (RIGHT POS)
C       FGY : FORCE AT FOOT IN Y-DIRECTION (UP  POS)
C       PI  : PI=4.0*ATAN(1.)
C       AX  : ACCELERATION OF BODY IN X-DIRECTION (IN./SEC2)
C       AY  : ACCELERATION OF BODY IN Y-DIRECTION (IN./SEC2)
C       BI  : ANGLE FOR ACCELERATION OF GRAVITY
C       G   : POSITIVE FOR DOWNHILL SLOPE (RAD.)
C       Q   : ACCELERATION OF GRAVITY (386 IN./SEC2)

PARAMETERS RETURNED

C       A : ARRAY OF 9x9 MATRIX EQUATION COEFFICIENTS
C       B : ARRAY OF 9x1 MATRIX SOLUTION COEFFICIENTS

PARAMETERS INTERNAL

C       COSTH : COSINE OF DRIVE CRANK ANGLE
C       COSGA : " " DRIVEN CRANK ANGLE
C       COSPH : " " COUPLER
C       SINTH : SINE OF DRIVE CRANK ANGLE
C       SINGA : " " DRIVEN CRANK ANGLE
C       SINPH : " " COUPLER
C       COSTHM : COSINE OF TOTAL ANGLE LOCATING C. Q. OF LINK 1
C       COSGAM : " 2
C       COSPHM : " 3
C       SINTHM : SINE OF TOTAL ANGLE LOCATING C. Q. OF LINK 1
C       SINGAM : " 2
C       SINPHM : " 3
C       COSSI : COSINE OF ANGLE GRAVITY VECTOR MAKES WITH MACHINE
C       SINSI : SINE "
C       COSPHL : COSINE OF TOTAL ANGLE LOCATING FOOT POINT ON COUPLER
C       SINPHL : SINE "
C       APRIME : INTERMEDIATE VALUES FOR CALCULATING MATRIX COEFF.
C       BPRIME : "
C       CPRIME : "
C       DPRIME : "

EXTERNAL REFERENCES

C       NONE

CALLING ROUTINES

C       ANALYS

*************************************************************************

C       DIMENSION A(9,9),B(9,1)
C       COMMON /A/ L1,L3,L5,LX,LY,PHIL
C       COMMON /C/ L2,THETA,GAMMA,PHI
C       COMMON /D/ THDOT1,QADOT1,PHDOT1
C       COMMON /H/ THETAM,GAMMAM,PHIM,LH1,LH2,LH3,JCQ1,JCQ2,JCQ3,
C       M1,M2,M3
C       COMMON /AA/ RA,RB,RC,RO,MAU,MBU,MUC,MUD,FGX,FGY,AX,Ay,BI,G
C       COMMON /DD/ TAF,TBF,TCF,TFD

C

279
REAL L1, L2, L3, L5, LX, LY, LM1, LM2, LM3, M1, M2, M3, JCO1, JCO2, JCO3, 
NUA, MUB, MUC, MUD
C
DO 10 I=1, 9
DO 11 J=1, 9
A(I, J)=0. 0
11 CONTINUE
10 CONTINUE
C
DEFINE INDIVIDUAL KNOWN VALUES OF MATRIX ARRAY "A"

C
COSTH=cos(THETA+THETAM)
SINTH=sin(THETA+THETAM)
COSGAM=cos(GAMMA+GAMMAM)
SINGAM=sin(GAMMA+GAMMAM)
COSPH=cos(PHI+PHIM)
SINPH=sin(PHI+PHIM)
COSTH=cos(THETA)
SINTH=sin(THETA)
COSGAM=cos(GAMMA)
SINGAM=sin(GAMMA)
COSPH=cos(PHI)
SINPH=sin(PHI)
COSSI=cos(SI)
SINSI=sin(SI)
COSPHL=cos(PHI+PHI)
SINPHL=sin(PHI+PHI)
C
DEFINE THE VALUES OF APRIME, BPRIME, CPRIME, DPRIME

APRIME=(L3*COSPH*L1*SINTH-L3*SINPH*L1*COSTH) /
#(L2*SINGA*L3*COSP H-L2*COSGAM*L3*SINPH)

BPRIME=((L3*COSPH)*(THDOT1**2*L1*COSTH-QADOT1**2*L2*COSGA
#-PDOT1**2*L3*COSPH)-(L3*SINPH)*(THDOT1**2*L1*SINTH+
# QADOT1**2*L2*SINGA+PDOT1**2*L3*SINPH))/
#(L2*SINGA*L3*COSP H-L2*COSGAM*L3*SINPH)

CPRIME=(L2*SINGA*L1*COSTH-L2*COSGAM*L1*SINTH) /
#(L2*SINGA*L3*COSP H-L2*COSGAM*L3*SINPH)

DPRIME=((L2*SINGA)*(-THDOT1**2*L1*SINTH+QADOT1**2*L2*SINGA
#+PDOT1**2*L3*SINPH)-(L2*COSGA)*(THDOT1**2*L1*COSTH-
# QADOT1**2*L2*COSGAM-PDOT1**2*L3*COSPH))/
#(L2*SINGA*L3*COSP H-L2*COSGAM*L3*SINPH)

NOW IT IS POSSIBLE TO DEFINE THE NON ZERO A(I, J) OF THE "A" MATRIX.

A(1, 1)=COSTH
A(1, 2)=SINTH
A(1, 3)=COSTH
A(1, 4)=SINTH
A(1, 9)=M1*M1*SINTHM
A(2, 1)=SINTH
A(2, 2)=COSTH
A(2,3)=SINTH
A(2,4)=COSTH
A(2,7)=M1*LM1*COSTH
A(3,1)=COSTH*LM1*SINTH-SINTH*LM1*COSTH
A(3,2)=SINTH*LM1*SINTH+COSTH*LM1*COSTH
A(3,3)=COSTH*LM1*SINTH-COSTH*LM1*SINTH-SINTH*L1
* C0STH+SINTH+LM1*C0STH
A(3,4)=SINTH*L1*SINTH-SINTH*LM1*SINTH+C0STH*L1
* COSTH-C0STH*LM1*C0STH
A(3,9)=JCG1
A(4,5)=COSGA
A(4,6)=SINGA
A(4,7)=COSGA
A(4,8)=SINGA
A(4,9)=M2*LM2*SINGAM*APRIME
A(5,5)=SINGA
A(5,6)=COSGA
A(5,7)=SINGA
A(5,8)=COSGA
A(5,9)=M2*LM2*COSGAM*APRIME
A(6,5)=COSGA*LM2*SINGA-COSGA*LM2*SINGAM-SINGA*L2
* COSGA+SINGA*LM2*COSGAM
A(6,6)=SINGA*L2*SINGA-SINGA*LM2*SINGAM-COSGA*L2*COSGAM
* -COSGA*LM2*COSGAM
A(6,7)=COSGA*LM2*SINGAM-SINGA*LM2*COSGAM
A(6,8)=SINGA*LM2*SINGAM+COSGA*LM2*COSGAM
A(6,9)=JCG2*APRIME
A(7,3)=COSTH
A(7,4)=SINTH
A(7,5)=COSGA
A(7,6)=SINGA
A(7,9)=M3*LM3*SINPHM*CPRIME+M3*L2*SINGA*APRIME
A(8,3)=SINTH
A(8,4)=COSTH
A(8,5)=SINGA
A(8,6)=COSGA
A(8,9)=M3*L2*COSGAM*APRIME+M3*LM3*COSPHM*CPRIME
A(9,3)=SINTH*L3*COSPH-SINTH*LM3*COSPHM
* -COSTH*L3*SINPH+COSTH*LM3*SINPHM
A(9,4)=COSTH*L3*COSPH+COSTH*LM3*COSPHM-SINPH*L3*SINPH
+ SINTH*LM3*SINPHM
A(9,5)=SINGA*LM3*COSPHM+COSGA*LM3*SINPHM
A(9,6)=COSGA*LM3*COSPHM+SINGA*LM3*SINPHM
A(9,9)=JCG3*CPRIME

DEFINE THE PARAMETERS OF THE VECTOR "B"

B(1,1)=M1*G*SINSI-M1*AX+M1*LM1*THD0T1**2*C0STH
B(2,1)=M1*G*C0SSI-M1*AY-M1*LM1*THD0T1**2*SINTH
B(3,1)=TAA(THETA)-MUA*RA+TAF-MUB*RB+TBF
B(4,1)=M2*G*SINSI-M2*AX+M2*LM2*GAD0T1**2*COSGAM
+ M2*LM2*SINGAM*APRIME
B(5,1)=M2*G*C0SSI-M2*AY-M2*LM2*GAD0T1**2*SINGAM
+ M2*LM2*COSGAM*APRIME
B(6,1)=JCO2*BPRIME-TDD(GAMMA)-MUC*RC+TCF
- MUD*RD*TDF
B(7,1)=M3*AX+M3*G*SINSI+M3*LM3*PHD0T1**2*COSPHM


PROGRAM PAGE = 128

\$ +M3*L2*GADOT1##2*COSG0A+M3*L3*SINPHM*DPRI
M3*L2*SINGA*BPRIME+FQX
B* (B, 1)*=-M3*AY-M3*G*COSG1-M3*L3*PHD0T1##2*SINPHM
M3*L2*GADOT1##2*SINGA+M3*L2*COSG0A*BPRIME
M3*L3*COSPHM*DPRI+FCY
B* (9, 1)*=FCY*(L3+COSPHM-L5+COSPHL)-FCY*(L5+SINPHL-L3
* SINPHM)-JCG3*DPRI+MUC*RC*TCF+MUB*R8*TFB
RETURN
END
FUNCTION TAA(THETA)

DESCRIPTION

THIS ROUTINE SPECIFIES THE TORQUE AT THE JOINT OF THE LINKAGE AS A FUNCTION OF "THETA"

PARAMETERS INPUT

THETA : DRIVE CRANK ANGLE OF SWING PHASE LEG

PARAMETERS RETURNED

TAA : APPLIED TORQUE AT SUPPORT END OF DRIVE CRANK

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING ROUTINES

VALUES

TAA = 0.0

RETURN

END
FUNCTION TDD(GAMMA)

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE SPECIFIES THE EXTERNAL TORQUE AT
JOINT "D" AS A FUNCTION OF "GAMMA"

PARAMETERS INPUT

GAMMA : ANGLE OF DRIVEN CRANK OF SWING PHASE LEG

PARAMETERS RETURNED

TDD : APPLIED TORQUE AT SUPPORT END OF DRIVEN CRANK

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING ROUTINES

VALUES

TDD=0.0

RETURN
END
SUBROUTINE VELOC

3-MAY-82  AUTHOR: FRANK BROWN

DESCRIPTION

THIS ROUTINE CALCULATES THE DRIVEN CRANK AND COUPLER ANGULAR VELOCITIES AS A FUNCTION OF THE ANGULAR VELOCITY OF THE FIXED CRANK (THDOT1)

PARAMETERS INPUT

L1 : LENGTH OF DRIVE CRANK  (IN.)
L2 : LENGTH OF DRIVEN CRANK  (IN.)
L3 : LENGTH OF COUPLER  (IN.)
THDOT1 : ANGULAR VELOCITY OF DRIVE CRANK  (CW POS.)
COSTH : COSINE OF DRIVE CRANK ANGLE
COSGA : DRIVEN CRANK ANGLE
COSPHE : COUPLER
SINTH : SINE OF DRIVE CRANK ANGLE
SINGA : DRIVEN CRANK ANGLE
SINPH : COUPLER

PARAMETERS RETURNED

QADOT1 : DRIVEN CRANK  (CW POS.)
PHDOT1 : COUPLER  (CW POS.)

PARAMETERS INTERNAL

NONE

EXTERNAL REFERENCES

NONE

CALLING ROUTINES

ANALYS

COMMON /A/ L1, L3, L5, LX, LY, PHIL
COMMON /C/ L2, THETA, GAMMA, PHI
COMMON /D/ THDOT1, QADOT1, PHDOT1
COMMON /F/ COSTH, SINTH, COSGA, SINGA, COSPH, SINPH

REAL L1, L2, L3, L5, LX, LY

QADOT1 = (THDOT1*L1*SINTH*L3*COSPH - THDOT1*L1*COSTH*L3*SINPH) /
         (L2*SINGA*L3*COSPH - L2*COSGA*L3*SINPH)

* COMPUTE QADOT1—DRIVEN CRANK ANGULAR VELOCITY

285
PHDDTI = (THDOTI*L1*COSTH*L2*SINGA-THDOTI*L1*SINTH*L2*COSGA)/
          (L2*SINGA*L3*COSPH-L2*COSGA*L3*SINPH)

C    * COMPUTE PHDDTI--COUPLER CRANK ANGULAR VELOCITY

RETURN
END
APPENDIX B

SAMPLE SIMULATION RESULTS
THIS DATA FILE IS USED IN CONJUNCTION WITH THE PROGRAM "DUNE" WHICH SIMULATES THE ACTION OF A PASSIVE 4-BAR LINKAGE WALKING MACHINE.

CASE TITLE:   EXPERIMENTAL TEST NO. 1

DATE:  7-JUL-82

LEG GEOMETRY

\[
\begin{array}{cccccccc}
L_1 & L_2 & L_3 & L_4 & L_5 \\
0.6774E+01 & 0.8200E+01 & 0.4050E+01 & 0.3170E+01 & 0.3500E+01 & 0.9900E+01 \\
\end{array}
\]

LEG MASS LOCATION

\[
\begin{array}{cccccccc}
LM_1 & LM_2 & LM_3 & \theta_1 & \gamma_1 & \phi_1 \\
0.1500E+01 & 0.1600E+01 & 0.2080E+01 & 0.7100E+02 & 0.5500E+02 & 0.1150E+03 \\
\end{array}
\]

LEG MASS VALUES

\[
\begin{array}{cccccccc}
M_1 & M_2 & M_3 & J_{C1} & J_{C2} & J_{C3} \\
0.4410E-02 & 0.1050E-01 & 0.3980E-02 & 0.5050E-01 & 0.9370E-01 & 0.6190E-01 \\
\end{array}
\]

LINK BEARING COEFFICIENTS

\[
\begin{array}{cccc}
M_{UA} & M_{UB} & M_{UC} & M_{UD} \\
0.5000E+00 & 0.5000E+00 & 0.5000E+00 & 0.8000E+00 \\
\end{array}
\]

BEARING RADIIS

\[
\begin{array}{cccc}
R_A & R_B & R_C & R_D \\
0.3200E+00 & 0.3200E+00 & 0.3200E+00 & 0.3200E+00 \\
\end{array}
\]
### FRAME PARAMETERS

<table>
<thead>
<tr>
<th>PPOLE</th>
<th>BPPOLE</th>
<th>BDMDT</th>
<th>BDLEN</th>
<th>BDWET</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1241E+02</td>
<td>0.2200E+02</td>
<td>0.3600E+02</td>
<td>0.7200E+02</td>
<td>0.1500E+02</td>
</tr>
</tbody>
</table>

### STRIDE PARAMETERS

<table>
<thead>
<tr>
<th>SHORTL</th>
<th>FOOST</th>
<th>EXPLAC</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2000E+00</td>
<td>0.1106E+02</td>
<td>0.4300E+00</td>
<td>0.5800E+01</td>
</tr>
</tbody>
</table>

### MISCELLANEOUS PARAMETERS

<table>
<thead>
<tr>
<th>PHIL</th>
<th>GRAVITY</th>
<th>GRAVITY ANG.</th>
<th>INCRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9400E+02</td>
<td>0.3860E+03</td>
<td>0.0000E+00</td>
<td>0.80</td>
</tr>
</tbody>
</table>

### TIME HISTORY OF PROBLEM ABOVE

<table>
<thead>
<tr>
<th>I</th>
<th>TIME</th>
<th>BHTEMA</th>
<th>BOAMMA</th>
<th>BPHI</th>
<th>BL2</th>
<th>XMMASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000E+00</td>
<td>0.2856E+02</td>
<td>0.5171E+02</td>
<td>0.4286E+01</td>
<td>0.8200E+01</td>
<td>0.2158E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.4767E-01</td>
<td>0.2920E+02</td>
<td>0.5201E+02</td>
<td>0.4683E+01</td>
<td>0.8200E+01</td>
<td>0.2156E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.9534E-01</td>
<td>0.3023E+02</td>
<td>0.5249E+02</td>
<td>0.5755E+01</td>
<td>0.8200E+01</td>
<td>0.2156E+02</td>
</tr>
<tr>
<td>4</td>
<td>0.1430E+00</td>
<td>0.3145E+02</td>
<td>0.5307E+02</td>
<td>0.6807E+01</td>
<td>0.8200E+01</td>
<td>0.2159E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.1907E+00</td>
<td>0.3263E+02</td>
<td>0.5363E+02</td>
<td>0.7804E+01</td>
<td>0.8200E+01</td>
<td>0.2167E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.2364E+00</td>
<td>0.3375E+02</td>
<td>0.5418E+02</td>
<td>0.6729E+01</td>
<td>0.8200E+01</td>
<td>0.2179E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.2860E+00</td>
<td>0.3481E+02</td>
<td>0.5470E+02</td>
<td>0.9586E+01</td>
<td>0.8200E+01</td>
<td>0.2195E+02</td>
</tr>
<tr>
<td>8</td>
<td>0.3337E+00</td>
<td>0.3587E+02</td>
<td>0.5523E+02</td>
<td>0.1043E+02</td>
<td>0.8200E+01</td>
<td>0.2214E+02</td>
</tr>
<tr>
<td>9</td>
<td>0.3814E+00</td>
<td>0.3699E+02</td>
<td>0.5580E+02</td>
<td>0.1130E+02</td>
<td>0.8200E+01</td>
<td>0.2234E+02</td>
</tr>
<tr>
<td>10</td>
<td>0.4291E+00</td>
<td>0.3821E+02</td>
<td>0.5643E+02</td>
<td>0.1223E+02</td>
<td>0.8200E+01</td>
<td>0.2253E+02</td>
</tr>
<tr>
<td>11</td>
<td>0.4767E+00</td>
<td>0.3949E+02</td>
<td>0.5710E+02</td>
<td>0.1316E+02</td>
<td>0.8200E+01</td>
<td>0.2270E+02</td>
</tr>
<tr>
<td>12</td>
<td>0.5244E+00</td>
<td>0.4081E+02</td>
<td>0.5780E+02</td>
<td>0.1412E+02</td>
<td>0.8200E+01</td>
<td>0.2283E+02</td>
</tr>
<tr>
<td>13</td>
<td>0.5721E+00</td>
<td>0.4213E+02</td>
<td>0.5852E+02</td>
<td>0.1505E+02</td>
<td>0.8200E+01</td>
<td>0.2293E+02</td>
</tr>
<tr>
<td>14</td>
<td>0.6197E+00</td>
<td>0.4346E+02</td>
<td>0.5925E+02</td>
<td>0.1594E+02</td>
<td>0.8200E+01</td>
<td>0.2297E+02</td>
</tr>
<tr>
<td>15</td>
<td>0.6674E+00</td>
<td>0.4481E+02</td>
<td>0.6001E+02</td>
<td>0.1682E+02</td>
<td>0.8200E+01</td>
<td>0.2298E+02</td>
</tr>
<tr>
<td>16</td>
<td>0.7151E+00</td>
<td>0.4620E+02</td>
<td>0.6080E+02</td>
<td>0.1769E+02</td>
<td>0.8200E+01</td>
<td>0.2297E+02</td>
</tr>
<tr>
<td>17</td>
<td>0.7628E+00</td>
<td>0.4762E+02</td>
<td>0.6163E+02</td>
<td>0.1856E+02</td>
<td>0.8200E+01</td>
<td>0.2287E+02</td>
</tr>
<tr>
<td>18</td>
<td>0.8104E+00</td>
<td>0.4915E+02</td>
<td>0.6254E+02</td>
<td>0.1945E+02</td>
<td>0.8200E+01</td>
<td>0.2276E+02</td>
</tr>
<tr>
<td>19</td>
<td>0.8581E+00</td>
<td>0.5084E+02</td>
<td>0.6356E+02</td>
<td>0.2036E+02</td>
<td>0.8200E+01</td>
<td>0.2261E+02</td>
</tr>
<tr>
<td>20</td>
<td>0.9058E+00</td>
<td>0.5275E+02</td>
<td>0.6474E+02</td>
<td>0.2137E+02</td>
<td>0.8200E+01</td>
<td>0.2244E+02</td>
</tr>
<tr>
<td>21</td>
<td>0.9534E+00</td>
<td>0.5485E+02</td>
<td>0.6607E+02</td>
<td>0.2240E+02</td>
<td>0.8200E+01</td>
<td>0.2224E+02</td>
</tr>
<tr>
<td>22</td>
<td>0.1001E+01</td>
<td>0.5704E+02</td>
<td>0.6750E+02</td>
<td>0.2339E+02</td>
<td>0.8200E+01</td>
<td>0.2221E+02</td>
</tr>
<tr>
<td>23</td>
<td>0.1049E+01</td>
<td>0.5923E+02</td>
<td>0.6896E+02</td>
<td>0.2431E+02</td>
<td>0.8200E+01</td>
<td>0.2217E+02</td>
</tr>
<tr>
<td>24</td>
<td>0.1096E+01</td>
<td>0.6133E+02</td>
<td>0.7038E+02</td>
<td>0.2512E+02</td>
<td>0.8200E+01</td>
<td>0.2214E+02</td>
</tr>
<tr>
<td>25</td>
<td>0.1144E+01</td>
<td>0.6333E+02</td>
<td>0.7177E+02</td>
<td>0.2583E+02</td>
<td>0.8200E+01</td>
<td>0.2210E+02</td>
</tr>
<tr>
<td>26</td>
<td>0.1192E+01</td>
<td>0.6531E+02</td>
<td>0.7316E+02</td>
<td>0.2646E+02</td>
<td>0.8200E+01</td>
<td>0.2207E+02</td>
</tr>
<tr>
<td>27</td>
<td>0.1239E+01</td>
<td>0.6733E+02</td>
<td>0.7442E+02</td>
<td>0.2709E+02</td>
<td>0.8200E+01</td>
<td>0.2204E+02</td>
</tr>
<tr>
<td>28</td>
<td>0.1287E+01</td>
<td>0.6946E+02</td>
<td>0.7617E+02</td>
<td>0.2767E+02</td>
<td>0.8200E+01</td>
<td>0.2201E+02</td>
</tr>
</tbody>
</table>

289
THIS DATA FILE IS USED IN CONJUNCTION WITH THE PROGRAM "DUWE" WHICH SIMULATES THE ACTION OF A PASSIVE 4-BAR LINKAGE WALKING MACHINE.

CASE TITLE: EXPERIMENTAL TEST NO. 2

DATE: 7-JUL-82

LEG GEOMETRY

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>LX</th>
<th>LY</th>
<th>L5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6774E+01</td>
<td>0.8200E+01</td>
<td>0.4050E+01</td>
<td>0.3170E+01</td>
<td>0.3500E+01</td>
<td>0.9900E+01</td>
</tr>
</tbody>
</table>

LEG MASS LOCATION

<table>
<thead>
<tr>
<th>LM1</th>
<th>LM2</th>
<th>LM3</th>
<th>THETAM</th>
<th>GAMMAM</th>
<th>PHIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1500E+01</td>
<td>0.1600E+01</td>
<td>0.2080E+01</td>
<td>0.7100E+02</td>
<td>0.5500E+02</td>
<td>0.1150E+03</td>
</tr>
</tbody>
</table>

LEG MASS VALUES

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>JCG1</th>
<th>JCG2</th>
<th>JCG3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4410E-02</td>
<td>0.1050E-01</td>
<td>0.3880E-02</td>
<td>0.5050E-01</td>
<td>0.9370E-01</td>
<td>0.6190E-01</td>
</tr>
</tbody>
</table>

LINK BEARING COEFFICIENTS

<table>
<thead>
<tr>
<th>MUA</th>
<th>MUB</th>
<th>MUC</th>
<th>MUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td>0.8000E+00</td>
</tr>
</tbody>
</table>

BEARING RADI

<table>
<thead>
<tr>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
</tr>
</tbody>
</table>
### FRAME PARAMETERS

<table>
<thead>
<tr>
<th>PLACLO</th>
<th>BPLACLO</th>
<th>BDWIDT</th>
<th>BDOLENT</th>
<th>BDWEIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1241E+02</td>
<td>0.2200E+02</td>
<td>0.3600E+02</td>
<td>0.7200E+02</td>
<td>0.1500E+02</td>
</tr>
</tbody>
</table>

### STRIDE PARAMETERS

<table>
<thead>
<tr>
<th>SHORTL</th>
<th>FDOOTST</th>
<th>EXPLAC</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2000E+00</td>
<td>0.1285E+02</td>
<td>0.1250E+01</td>
<td>0.9000E+01</td>
</tr>
</tbody>
</table>

### MISCELLANEOUS PARAMETERS

<table>
<thead>
<tr>
<th>PHIL</th>
<th>GRAVITY</th>
<th>GRAVITY ANG.</th>
<th>INCRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9400E+02</td>
<td>0.3860E+03</td>
<td>0.0000E+00</td>
<td>80</td>
</tr>
</tbody>
</table>

### TIME HISTORY OF PROBLEM ABOVE

<table>
<thead>
<tr>
<th>I</th>
<th>TIME</th>
<th>BTHTA</th>
<th>BOMMA</th>
<th>BPHTI</th>
<th>BL2</th>
<th>BMMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000E+00</td>
<td>0.2592E+02</td>
<td>0.5051E+02</td>
<td>0.1883E+01</td>
<td>0.8200E+01</td>
<td>0.2156E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.3569E-01</td>
<td>0.2687E+02</td>
<td>0.5094E+02</td>
<td>0.2756E+01</td>
<td>0.8200E+01</td>
<td>0.2152E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.7139E-01</td>
<td>0.2849E+02</td>
<td>0.5168E+02</td>
<td>0.4223E+01</td>
<td>0.8200E+01</td>
<td>0.2150E+02</td>
</tr>
<tr>
<td>4</td>
<td>0.1071E+00</td>
<td>0.3023E+02</td>
<td>0.5249E+02</td>
<td>0.5795E+01</td>
<td>0.8200E+01</td>
<td>0.2148E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.1438E+00</td>
<td>0.3165E+02</td>
<td>0.5316E+02</td>
<td>0.6779E+01</td>
<td>0.8200E+01</td>
<td>0.2150E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.1785E+00</td>
<td>0.3283E+02</td>
<td>0.5373E+02</td>
<td>0.7971E+01</td>
<td>0.8200E+01</td>
<td>0.2154E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.2142E+00</td>
<td>0.3391E+02</td>
<td>0.5426E+02</td>
<td>0.8863E+01</td>
<td>0.8200E+01</td>
<td>0.2162E+02</td>
</tr>
<tr>
<td>8</td>
<td>0.2499E+00</td>
<td>0.3504E+02</td>
<td>0.5482E+02</td>
<td>0.9774E+01</td>
<td>0.8200E+01</td>
<td>0.2173E+02</td>
</tr>
<tr>
<td>9</td>
<td>0.2856E+00</td>
<td>0.3629E+02</td>
<td>0.5544E+02</td>
<td>1.0765E+02</td>
<td>0.8200E+01</td>
<td>0.2185E+02</td>
</tr>
<tr>
<td>10</td>
<td>0.3212E+00</td>
<td>0.3765E+02</td>
<td>0.5614E+02</td>
<td>1.1809E+02</td>
<td>0.8200E+01</td>
<td>0.2200E+02</td>
</tr>
<tr>
<td>11</td>
<td>0.3569E+00</td>
<td>0.3911E+02</td>
<td>0.5690E+02</td>
<td>1.2909E+02</td>
<td>0.8200E+01</td>
<td>0.2216E+02</td>
</tr>
<tr>
<td>12</td>
<td>0.3926E+00</td>
<td>0.4067E+02</td>
<td>0.5773E+02</td>
<td>1.4032E+02</td>
<td>0.8200E+01</td>
<td>0.2232E+02</td>
</tr>
<tr>
<td>13</td>
<td>0.4283E+00</td>
<td>0.4233E+02</td>
<td>0.5862E+02</td>
<td>1.5182E+02</td>
<td>0.8200E+01</td>
<td>0.2247E+02</td>
</tr>
<tr>
<td>14</td>
<td>0.4640E+00</td>
<td>0.4400E+02</td>
<td>0.5959E+02</td>
<td>1.6398E+02</td>
<td>0.8200E+01</td>
<td>0.2260E+02</td>
</tr>
<tr>
<td>15</td>
<td>0.4997E+00</td>
<td>0.4573E+02</td>
<td>0.6063E+02</td>
<td>1.7532E+02</td>
<td>0.8200E+01</td>
<td>0.2271E+02</td>
</tr>
<tr>
<td>16</td>
<td>0.5354E+00</td>
<td>0.4788E+02</td>
<td>0.6178E+02</td>
<td>1.8718E+02</td>
<td>0.8200E+01</td>
<td>0.2280E+02</td>
</tr>
<tr>
<td>17</td>
<td>0.5711E+00</td>
<td>0.4992E+02</td>
<td>0.6300E+02</td>
<td>1.9988E+02</td>
<td>0.8200E+01</td>
<td>0.2286E+02</td>
</tr>
<tr>
<td>18</td>
<td>0.6068E+00</td>
<td>0.5204E+02</td>
<td>0.6430E+02</td>
<td>2.1018E+02</td>
<td>0.8200E+01</td>
<td>0.2288E+02</td>
</tr>
<tr>
<td>19</td>
<td>0.6425E+00</td>
<td>0.5419E+02</td>
<td>0.6565E+02</td>
<td>2.2092E+02</td>
<td>0.8200E+01</td>
<td>0.2288E+02</td>
</tr>
<tr>
<td>20</td>
<td>0.6782E+00</td>
<td>0.5634E+02</td>
<td>0.6704E+02</td>
<td>2.3097E+02</td>
<td>0.8200E+01</td>
<td>0.2287E+02</td>
</tr>
<tr>
<td>21</td>
<td>0.7139E+00</td>
<td>0.5844E+02</td>
<td>0.6843E+02</td>
<td>2.3997E+02</td>
<td>0.8200E+01</td>
<td>0.2282E+02</td>
</tr>
<tr>
<td>22</td>
<td>0.7496E+00</td>
<td>0.6050E+02</td>
<td>0.6982E+02</td>
<td>2.4812E+02</td>
<td>0.8200E+01</td>
<td>0.2275E+02</td>
</tr>
<tr>
<td>23</td>
<td>0.7853E+00</td>
<td>0.6254E+02</td>
<td>0.7122E+02</td>
<td>2.5565E+02</td>
<td>0.8200E+01</td>
<td>0.2267E+02</td>
</tr>
<tr>
<td>24</td>
<td>0.8210E+00</td>
<td>0.6459E+02</td>
<td>0.7265E+02</td>
<td>2.6299E+02</td>
<td>0.8200E+01</td>
<td>0.2266E+02</td>
</tr>
<tr>
<td>25</td>
<td>0.8567E+00</td>
<td>0.6664E+02</td>
<td>0.7412E+02</td>
<td>2.6989E+02</td>
<td>0.8200E+01</td>
<td>0.2243E+02</td>
</tr>
<tr>
<td>26</td>
<td>0.8924E+00</td>
<td>0.6875E+02</td>
<td>0.7565E+02</td>
<td>2.7749E+02</td>
<td>0.8200E+01</td>
<td>0.2228E+02</td>
</tr>
<tr>
<td>27</td>
<td>0.9281E+00</td>
<td>0.7093E+02</td>
<td>0.7725E+02</td>
<td>2.8404E+02</td>
<td>0.8200E+01</td>
<td>0.2213E+02</td>
</tr>
<tr>
<td>28</td>
<td>0.9637E+00</td>
<td>0.7317E+02</td>
<td>0.7893E+02</td>
<td>2.8856E+02</td>
<td>0.8200E+01</td>
<td>0.2202E+02</td>
</tr>
</tbody>
</table>

CASE TITLE: EXPERIMENT CASE NO. 3

DATE: 19-JUL-82

LEG GEOMETRY

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>LX</th>
<th>LY</th>
<th>L5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6774E+01</td>
<td>0.8200E+01</td>
<td>0.4050E+01</td>
<td>0.3170E+01</td>
<td>0.3500E+01</td>
<td>0.9900E+01</td>
</tr>
</tbody>
</table>

LEG MASS LOCATION

<table>
<thead>
<tr>
<th>LM1</th>
<th>LM2</th>
<th>LM3</th>
<th>THETAM</th>
<th>GAMMAM</th>
<th>PHIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1500E+01</td>
<td>0.1600E+01</td>
<td>0.2080E+01</td>
<td>0.7100E+02</td>
<td>0.5500E+02</td>
<td>0.1150E+03</td>
</tr>
</tbody>
</table>

LEG MASS VALUES

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>JGQ1</th>
<th>JGQ2</th>
<th>JGQ3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4410E-02</td>
<td>0.1050E-01</td>
<td>0.3880E-02</td>
<td>0.5050E-01</td>
<td>0.9370E-01</td>
<td>0.6190E-01</td>
</tr>
</tbody>
</table>

LINK BEARING COEFFICIENTS

<table>
<thead>
<tr>
<th>MUA</th>
<th>MUB</th>
<th>MUC</th>
<th>MUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td>0.8000E+00</td>
</tr>
</tbody>
</table>

BEARING RADII

<table>
<thead>
<tr>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
</tr>
</tbody>
</table>

294
### FRAME PARAMETERS

<table>
<thead>
<tr>
<th>PLACLQ</th>
<th>SPACLQ</th>
<th>BDWIDT</th>
<th>DDLNENT</th>
<th>BDWEIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1241E+02</td>
<td>0.2200E+02</td>
<td>0.3600E+02</td>
<td>0.7200E+02</td>
<td>0.1500E+02</td>
</tr>
</tbody>
</table>

### STRIDE PARAMETERS

<table>
<thead>
<tr>
<th>SHORTL</th>
<th>FOOTST</th>
<th>EXPLAC</th>
<th>BPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2000E+00</td>
<td>0.1164E+02</td>
<td>0.8530E+00</td>
<td>0.7867E+01</td>
</tr>
</tbody>
</table>

### MISCELLANEOUS PARAMETERS

<table>
<thead>
<tr>
<th>PHIL</th>
<th>GRAVITY</th>
<th>GRAVITY ANO.</th>
<th>INCRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9400E+02</td>
<td>0.3860E+03</td>
<td>0.0000E+00</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### TIME HISTORY OF PROBLEM ABOVE

<table>
<thead>
<tr>
<th>I</th>
<th>TIME</th>
<th>BTHETA</th>
<th>BOAMMA</th>
<th>BPHI</th>
<th>BL2</th>
<th>XMMASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000E+00</td>
<td>0.2765E+02</td>
<td>0.5129E+02</td>
<td>0.3465E+01</td>
<td>0.8200E+01</td>
<td>0.2158E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.3699E-01</td>
<td>0.2938E+02</td>
<td>0.5209E+02</td>
<td>0.5010E+01</td>
<td>0.8200E+01</td>
<td>0.2155E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.7398E-01</td>
<td>0.3083E+02</td>
<td>0.5277E+02</td>
<td>0.6278E+01</td>
<td>0.8200E+01</td>
<td>0.2155E+02</td>
</tr>
<tr>
<td>4</td>
<td>0.1110E+00</td>
<td>0.3265E+02</td>
<td>0.5325E+02</td>
<td>0.7143E+01</td>
<td>0.8200E+01</td>
<td>0.2159E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.1480E+00</td>
<td>0.3293E+02</td>
<td>0.5378E+02</td>
<td>0.8048E+01</td>
<td>0.8200E+01</td>
<td>0.2161E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.1854E+00</td>
<td>0.3436E+02</td>
<td>0.5448E+02</td>
<td>0.9228E+01</td>
<td>0.8200E+01</td>
<td>0.2174E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.2219E+00</td>
<td>0.3591E+02</td>
<td>0.5525E+02</td>
<td>1.0470E+02</td>
<td>0.8200E+01</td>
<td>0.2183E+02</td>
</tr>
<tr>
<td>8</td>
<td>0.2589E+00</td>
<td>0.3726E+02</td>
<td>0.5594E+02</td>
<td>1.1510E+02</td>
<td>0.8200E+01</td>
<td>0.2196E+02</td>
</tr>
<tr>
<td>9</td>
<td>0.2959E+00</td>
<td>0.3846E+02</td>
<td>0.5656E+02</td>
<td>1.2420E+02</td>
<td>0.8200E+01</td>
<td>0.2210E+02</td>
</tr>
<tr>
<td>10</td>
<td>0.3329E+00</td>
<td>0.3974E+02</td>
<td>0.5723E+02</td>
<td>1.3360E+02</td>
<td>0.8200E+01</td>
<td>0.2223E+02</td>
</tr>
<tr>
<td>11</td>
<td>0.3699E+00</td>
<td>0.4119E+02</td>
<td>0.5800E+02</td>
<td>1.4390E+02</td>
<td>0.8200E+01</td>
<td>0.2239E+02</td>
</tr>
<tr>
<td>12</td>
<td>0.4067E+00</td>
<td>0.4276E+02</td>
<td>0.5886E+02</td>
<td>1.5440E+02</td>
<td>0.8200E+01</td>
<td>0.2252E+02</td>
</tr>
<tr>
<td>13</td>
<td>0.4439E+00</td>
<td>0.4411E+02</td>
<td>0.5978E+02</td>
<td>1.6570E+02</td>
<td>0.8200E+01</td>
<td>0.2263E+02</td>
</tr>
<tr>
<td>14</td>
<td>0.4809E+00</td>
<td>0.4608E+02</td>
<td>0.6073E+02</td>
<td>1.7660E+02</td>
<td>0.8200E+01</td>
<td>0.2272E+02</td>
</tr>
<tr>
<td>15</td>
<td>0.5179E+00</td>
<td>0.4772E+02</td>
<td>0.6169E+02</td>
<td>1.8610E+02</td>
<td>0.8200E+01</td>
<td>0.2279E+02</td>
</tr>
<tr>
<td>16</td>
<td>0.5548E+00</td>
<td>0.4936E+02</td>
<td>0.6264E+02</td>
<td>1.9560E+02</td>
<td>0.8200E+01</td>
<td>0.2284E+02</td>
</tr>
<tr>
<td>17</td>
<td>0.5918E+00</td>
<td>0.5105E+02</td>
<td>0.6349E+02</td>
<td>2.0490E+02</td>
<td>0.8200E+01</td>
<td>0.2286E+02</td>
</tr>
<tr>
<td>18</td>
<td>0.6288E+00</td>
<td>0.5283E+02</td>
<td>0.6479E+02</td>
<td>2.1410E+02</td>
<td>0.8200E+01</td>
<td>0.2286E+02</td>
</tr>
<tr>
<td>19</td>
<td>0.6658E+00</td>
<td>0.5473E+02</td>
<td>0.6600E+02</td>
<td>2.2340E+02</td>
<td>0.8200E+01</td>
<td>0.2284E+02</td>
</tr>
<tr>
<td>20</td>
<td>0.7028E+00</td>
<td>0.5681E+02</td>
<td>0.6735E+02</td>
<td>2.3290E+02</td>
<td>0.8200E+01</td>
<td>0.2278E+02</td>
</tr>
<tr>
<td>21</td>
<td>0.7398E+00</td>
<td>0.5908E+02</td>
<td>0.6865E+02</td>
<td>2.4250E+02</td>
<td>0.8200E+01</td>
<td>0.2270E+02</td>
</tr>
<tr>
<td>22</td>
<td>0.7768E+00</td>
<td>0.6130E+02</td>
<td>0.7050E+02</td>
<td>2.5190E+02</td>
<td>0.8200E+01</td>
<td>0.2260E+02</td>
</tr>
<tr>
<td>23</td>
<td>0.8138E+00</td>
<td>0.6397E+02</td>
<td>0.7222E+02</td>
<td>2.6050E+02</td>
<td>0.8200E+01</td>
<td>0.2256E+02</td>
</tr>
<tr>
<td>24</td>
<td>0.8508E+00</td>
<td>0.6641E+02</td>
<td>0.7395E+02</td>
<td>2.6820E+02</td>
<td>0.8200E+01</td>
<td>0.2252E+02</td>
</tr>
<tr>
<td>25</td>
<td>0.8878E+00</td>
<td>0.6874E+02</td>
<td>0.7544E+02</td>
<td>2.7480E+02</td>
<td>0.8200E+01</td>
<td>0.2248E+02</td>
</tr>
<tr>
<td>26</td>
<td>0.9247E+00</td>
<td>0.7096E+02</td>
<td>0.7728E+02</td>
<td>2.8050E+02</td>
<td>0.8200E+01</td>
<td>0.2245E+02</td>
</tr>
<tr>
<td>27</td>
<td>0.9617E+00</td>
<td>0.7308E+02</td>
<td>0.7886E+02</td>
<td>2.8540E+02</td>
<td>0.8200E+01</td>
<td>0.2241E+02</td>
</tr>
<tr>
<td>28</td>
<td>0.9987E+00</td>
<td>0.7511E+02</td>
<td>0.8040E+02</td>
<td>2.8970E+02</td>
<td>0.8200E+01</td>
<td>0.2238E+02</td>
</tr>
</tbody>
</table>
THIS DATA FILE IS USED IN CONJUNCTION WITH THE PROGRAM "DUNE" WHICH SIMULATES THE ACTION OF A PASSIVE 4-BAR LINKAGE WALKING MACHINE.

CASE TITLE: EXPERIMENT CASE NO. 4

DATE: 20-JUL-82

LEQ GEOMETRY

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>LX</th>
<th>LY</th>
<th>L5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6774E+01</td>
<td>0.8200E+01</td>
<td>0.4050E+01</td>
<td>0.3170E+01</td>
<td>0.3500E+01</td>
<td>0.9900E+01</td>
</tr>
</tbody>
</table>

LEQ MASS LOCATION

<table>
<thead>
<tr>
<th>LM1</th>
<th>LM2</th>
<th>LM3</th>
<th>THETAM</th>
<th>GAMMA</th>
<th>PHIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1500E+01</td>
<td>0.1600E+01</td>
<td>0.2080E+01</td>
<td>0.7100E+02</td>
<td>0.5500E+02</td>
<td>0.1150E+03</td>
</tr>
</tbody>
</table>

LEQ MASS VALUES

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>JC01</th>
<th>JC02</th>
<th>JC03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4410E-02</td>
<td>0.1050E-01</td>
<td>0.3880E-02</td>
<td>0.5050E-01</td>
<td>0.9370E-01</td>
<td>0.6190E-01</td>
</tr>
</tbody>
</table>

LINK BEARING COEFFICIENTS

<table>
<thead>
<tr>
<th>MUa</th>
<th>MUb</th>
<th>Muc</th>
<th>MUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td>0.8000E+00</td>
</tr>
</tbody>
</table>

BEARING RADI

<table>
<thead>
<tr>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
</tr>
</tbody>
</table>
FRAME PARAMETERS

<table>
<thead>
<tr>
<th>PLANLQ</th>
<th>SPCAPLQ</th>
<th>BDWIDT</th>
<th>BDLEN</th>
<th>BDWIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1241E+02</td>
<td>0.2200E+02</td>
<td>0.3600E+02</td>
<td>0.7200E+02</td>
<td>0.1500E+02</td>
</tr>
</tbody>
</table>

STRIDE PARAMETERS

<table>
<thead>
<tr>
<th>SHORTL</th>
<th>FOOTST</th>
<th>EXPLAC</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2000E+00</td>
<td>0.1282E+02</td>
<td>0.1805E+01</td>
<td>0.9662E+01</td>
</tr>
</tbody>
</table>

MISCELLANEOUS PARAMETERS

<table>
<thead>
<tr>
<th>PHI</th>
<th>GRAVITY</th>
<th>GRAVITY ANG.</th>
<th>INCRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9400E+02</td>
<td>0.3860E+03</td>
<td>0.0000E+00</td>
<td>80</td>
</tr>
</tbody>
</table>

TIME HISTORY OF PROBLEM ABOVE

<table>
<thead>
<tr>
<th>I</th>
<th>TIME</th>
<th>BTTHETA</th>
<th>BGAMMA</th>
<th>BPHI</th>
<th>BL2</th>
<th>XMASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000E+00</td>
<td>0.2504E+02</td>
<td>0.5012E+02</td>
<td>0.1660E+01</td>
<td>0.8200E+01</td>
<td>0.2161E+02</td>
</tr>
<tr>
<td>2</td>
<td>0.3316E-01</td>
<td>0.2593E+02</td>
<td>0.5052E+02</td>
<td>0.1691E+01</td>
<td>0.8200E+01</td>
<td>0.2198E+02</td>
</tr>
<tr>
<td>3</td>
<td>0.6633E-01</td>
<td>0.2691E+02</td>
<td>0.5096E+02</td>
<td>0.2799E+01</td>
<td>0.8200E+01</td>
<td>0.2158E+02</td>
</tr>
<tr>
<td>4</td>
<td>0.9949E-01</td>
<td>0.2789E+02</td>
<td>0.5144E+02</td>
<td>0.3764E+01</td>
<td>0.8200E+01</td>
<td>0.2159E+02</td>
</tr>
<tr>
<td>5</td>
<td>0.1327E+00</td>
<td>0.2921E+02</td>
<td>0.5201E+02</td>
<td>0.4800E+01</td>
<td>0.8200E+01</td>
<td>0.2162E+02</td>
</tr>
<tr>
<td>6</td>
<td>0.1658E+00</td>
<td>0.3068E+02</td>
<td>0.5270E+02</td>
<td>0.6146E+01</td>
<td>0.8200E+01</td>
<td>0.2165E+02</td>
</tr>
<tr>
<td>7</td>
<td>0.1990E+00</td>
<td>0.3215E+02</td>
<td>0.5340E+02</td>
<td>0.7400E+01</td>
<td>0.8200E+01</td>
<td>0.2171E+02</td>
</tr>
<tr>
<td>8</td>
<td>0.2321E+00</td>
<td>0.3343E+02</td>
<td>0.5402E+02</td>
<td>0.8686E+01</td>
<td>0.8200E+01</td>
<td>0.2180E+02</td>
</tr>
<tr>
<td>9</td>
<td>0.2653E+00</td>
<td>0.3467E+02</td>
<td>0.5463E+02</td>
<td>0.9474E+01</td>
<td>0.8200E+01</td>
<td>0.2190E+02</td>
</tr>
<tr>
<td>10</td>
<td>0.2985E+00</td>
<td>0.3602E+02</td>
<td>0.5531E+02</td>
<td>1.0555E+02</td>
<td>0.8200E+01</td>
<td>0.2202E+02</td>
</tr>
<tr>
<td>11</td>
<td>0.3316E+00</td>
<td>0.3752E+02</td>
<td>0.5607E+02</td>
<td>1.1711E+02</td>
<td>0.8200E+01</td>
<td>0.2214E+02</td>
</tr>
<tr>
<td>12</td>
<td>0.3648E+00</td>
<td>0.3913E+02</td>
<td>0.5681E+02</td>
<td>1.2911E+02</td>
<td>0.8200E+01</td>
<td>0.2227E+02</td>
</tr>
<tr>
<td>13</td>
<td>0.3980E+00</td>
<td>0.4033E+02</td>
<td>0.5761E+02</td>
<td>1.4116E+02</td>
<td>0.8200E+01</td>
<td>0.2238E+02</td>
</tr>
<tr>
<td>14</td>
<td>0.4311E+00</td>
<td>0.4260E+02</td>
<td>0.5877E+02</td>
<td>1.5378E+02</td>
<td>0.8200E+01</td>
<td>0.2249E+02</td>
</tr>
<tr>
<td>15</td>
<td>0.4643E+00</td>
<td>0.4442E+02</td>
<td>0.5997E+02</td>
<td>1.6657E+02</td>
<td>0.8200E+01</td>
<td>0.2259E+02</td>
</tr>
<tr>
<td>16</td>
<td>0.4975E+00</td>
<td>0.4630E+02</td>
<td>0.6086E+02</td>
<td>1.7776E+02</td>
<td>0.8200E+01</td>
<td>0.2267E+02</td>
</tr>
<tr>
<td>17</td>
<td>0.5306E+00</td>
<td>0.4823E+02</td>
<td>0.6199E+02</td>
<td>1.8982E+02</td>
<td>0.8200E+01</td>
<td>0.2273E+02</td>
</tr>
<tr>
<td>18</td>
<td>0.5638E+00</td>
<td>0.5033E+02</td>
<td>0.6319E+02</td>
<td>2.0005E+02</td>
<td>0.8200E+01</td>
<td>0.2277E+02</td>
</tr>
<tr>
<td>19</td>
<td>0.5969E+00</td>
<td>0.5230E+02</td>
<td>0.6446E+02</td>
<td>2.1114E+02</td>
<td>0.8200E+01</td>
<td>0.2279E+02</td>
</tr>
<tr>
<td>20</td>
<td>0.6301E+00</td>
<td>0.5441E+02</td>
<td>0.6580E+02</td>
<td>2.2199E+02</td>
<td>0.8200E+01</td>
<td>0.2280E+02</td>
</tr>
<tr>
<td>21</td>
<td>0.6633E+00</td>
<td>0.5657E+02</td>
<td>0.6719E+02</td>
<td>2.3192E+02</td>
<td>0.8200E+01</td>
<td>0.2278E+02</td>
</tr>
<tr>
<td>22</td>
<td>0.6964E+00</td>
<td>0.5873E+02</td>
<td>0.6862E+02</td>
<td>2.4118E+02</td>
<td>0.8200E+01</td>
<td>0.2276E+02</td>
</tr>
<tr>
<td>23</td>
<td>0.7296E+00</td>
<td>0.6089E+02</td>
<td>0.7009E+02</td>
<td>2.5077E+02</td>
<td>0.8200E+01</td>
<td>0.2271E+02</td>
</tr>
<tr>
<td>24</td>
<td>0.7628E+00</td>
<td>0.6303E+02</td>
<td>0.7156E+02</td>
<td>2.5836E+02</td>
<td>0.8200E+01</td>
<td>0.2264E+02</td>
</tr>
<tr>
<td>25</td>
<td>0.7959E+00</td>
<td>0.6513E+02</td>
<td>0.7304E+02</td>
<td>2.6643E+02</td>
<td>0.8200E+01</td>
<td>0.2257E+02</td>
</tr>
<tr>
<td>26</td>
<td>0.8291E+00</td>
<td>0.6719E+02</td>
<td>0.7452E+02</td>
<td>2.7405E+02</td>
<td>0.8200E+01</td>
<td>0.2248E+02</td>
</tr>
<tr>
<td>27</td>
<td>0.8622E+00</td>
<td>0.6919E+02</td>
<td>0.7597E+02</td>
<td>2.8160E+02</td>
<td>0.8200E+01</td>
<td>0.2238E+02</td>
</tr>
<tr>
<td>28</td>
<td>0.8954E+00</td>
<td>0.7113E+02</td>
<td>0.7740E+02</td>
<td>2.8909E+02</td>
<td>0.8200E+01</td>
<td>0.2227E+02</td>
</tr>
</tbody>
</table>
THIS DATA FILE IS USED IN CONJUNCTION WITH THE PROGRAM "DUNE" WHICH SIMULATES THE ACTION OF A PASSIVE 4-BAR LINKAGE WALKING MACHINE.

CASE TITLE: EXPERIMENT NO. 3 WITH CHANGES IN BEARING FRICTION

DATE: 24-JUL-82

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>LX</th>
<th>LY</th>
<th>L5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6774E+01</td>
<td>0.8200E+01</td>
<td>0.4050E+01</td>
<td>0.3170E+01</td>
<td>0.3500E+01</td>
<td>0.9900E+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LM1</th>
<th>LM2</th>
<th>LM3</th>
<th>THETAM</th>
<th>GAMMAM</th>
<th>PHIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1500E+01</td>
<td>0.1600E+01</td>
<td>0.2080E+01</td>
<td>0.7100E+02</td>
<td>0.5500E+02</td>
<td>0.1150E+03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>JCG1</th>
<th>JCG2</th>
<th>JCG3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4410E-02</td>
<td>0.1050E-01</td>
<td>0.3880E-02</td>
<td>0.5050E-01</td>
<td>0.9370E-01</td>
<td>0.6190E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MJU</th>
<th>MUB</th>
<th>MUC</th>
<th>MUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5500E+00</td>
<td>0.5500E+00</td>
<td>0.5500E+00</td>
<td>0.6500E+00</td>
</tr>
</tbody>
</table>

BEARING RADI.

<table>
<thead>
<tr>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>RD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
<td>0.3200E+00</td>
</tr>
<tr>
<td>FRAME PARAMETERS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>PLACLQ</td>
<td>0.1241E+02</td>
<td>0.2200E+02</td>
<td>0.3600E+02</td>
</tr>
<tr>
<td>BPAACLQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BDWIDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BDELNT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BDWEIT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STRIDE PARAMETERS</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SHORTL</td>
<td>0.2000E+00</td>
<td>0.1164E+02</td>
<td>0.9530E+00</td>
<td>0.7867E+01</td>
<td></td>
</tr>
<tr>
<td>FOOTST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPLAC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPEED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MISCELLANEOUS PARAMETERS</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIL</td>
<td>0.9400E+02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAVITY</td>
<td>0.3860E+03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAVITY ANG. INCRE</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME HISTORY OF PROBLEM ABOVE</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>TIME</td>
<td>BTHETA</td>
<td>BGAMMA</td>
<td>BPHI</td>
<td>BL2</td>
</tr>
<tr>
<td>1</td>
<td>0.0000E+00</td>
<td>0.2782E+02</td>
<td>0.5137E+02</td>
<td>0.3615E+01</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.3699E-01</td>
<td>0.2926E+02</td>
<td>0.5203E+02</td>
<td>0.4905E+01</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.7398E-01</td>
<td>0.3059E+02</td>
<td>0.5266E+02</td>
<td>0.6069E+01</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>4</td>
<td>0.1110E+00</td>
<td>0.3182E+02</td>
<td>0.5324E+02</td>
<td>0.7117E+01</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>5</td>
<td>0.1480E+00</td>
<td>0.3307E+02</td>
<td>0.5384E+02</td>
<td>0.8167E+01</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>6</td>
<td>0.1849E+00</td>
<td>0.3446E+02</td>
<td>0.5453E+02</td>
<td>0.9309E+01</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>7</td>
<td>0.2219E+00</td>
<td>0.3599E+02</td>
<td>0.5529E+02</td>
<td>0.1053E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>8</td>
<td>0.2589E+00</td>
<td>0.3753E+02</td>
<td>0.5613E+02</td>
<td>0.1179E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>9</td>
<td>0.2959E+00</td>
<td>0.3931E+02</td>
<td>0.5700E+02</td>
<td>0.1305E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>10</td>
<td>0.3329E+00</td>
<td>0.4098E+02</td>
<td>0.5798E+02</td>
<td>0.1425E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>11</td>
<td>0.3699E+00</td>
<td>0.4256E+02</td>
<td>0.5875E+02</td>
<td>0.1534E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>12</td>
<td>0.4069E+00</td>
<td>0.4405E+02</td>
<td>0.5958E+02</td>
<td>0.1633E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>13</td>
<td>0.4439E+00</td>
<td>0.4553E+02</td>
<td>0.6042E+02</td>
<td>0.1726E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>14</td>
<td>0.4809E+00</td>
<td>0.4708E+02</td>
<td>0.6131E+02</td>
<td>0.1824E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>15</td>
<td>0.5179E+00</td>
<td>0.4860E+02</td>
<td>0.6233E+02</td>
<td>0.1923E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>16</td>
<td>0.5549E+00</td>
<td>0.5027E+02</td>
<td>0.6349E+02</td>
<td>0.2031E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>17</td>
<td>0.5919E+00</td>
<td>0.5275E+02</td>
<td>0.6474E+02</td>
<td>0.2137E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>18</td>
<td>0.6289E+00</td>
<td>0.5518E+02</td>
<td>0.6605E+02</td>
<td>0.2238E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>19</td>
<td>0.6659E+00</td>
<td>0.5768E+02</td>
<td>0.6733E+02</td>
<td>0.2338E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>20</td>
<td>0.7028E+00</td>
<td>0.5864E+02</td>
<td>0.6856E+02</td>
<td>0.2407E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>21</td>
<td>0.7398E+00</td>
<td>0.6050E+02</td>
<td>0.6982E+02</td>
<td>0.2481E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>22</td>
<td>0.7768E+00</td>
<td>0.6232E+02</td>
<td>0.7120E+02</td>
<td>0.2555E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>23</td>
<td>0.8138E+00</td>
<td>0.6408E+02</td>
<td>0.7262E+02</td>
<td>0.2633E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>24</td>
<td>0.8508E+00</td>
<td>0.6583E+02</td>
<td>0.7404E+02</td>
<td>0.2710E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>25</td>
<td>0.8878E+00</td>
<td>0.6766E+02</td>
<td>0.7555E+02</td>
<td>0.2781E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>26</td>
<td>0.9248E+00</td>
<td>0.6946E+02</td>
<td>0.7704E+02</td>
<td>0.2842E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>27</td>
<td>0.9618E+00</td>
<td>0.7123E+02</td>
<td>0.7851E+02</td>
<td>0.2901E+02</td>
<td>0.8200E+01</td>
</tr>
<tr>
<td>28</td>
<td>0.9987E+00</td>
<td>0.7291E+02</td>
<td>0.8005E+02</td>
<td>0.2955E+02</td>
<td>0.8200E+01</td>
</tr>
</tbody>
</table>
APPENDIX C

COMPONENT SPECIFICATIONS
DETAIL DRAWING OF DUWE LEG LATCHING ASSEMBLY
DETAIL DRAWING OF DUWE LOAD RING
APPENDIX D

ELECTRICAL WIRING DIAGRAM
DUWE ELECTRICAL WIRING DIAGRAM
APPENDIX E

CONTROL PROGRAM
PROGRAM MNHNL;

(*-----------------------------------------------
---DUWE CONTROL PROGRAM---*

THIS PROGRAM OPERATES THE DUWE USING
THE POSITION CONTROL VARIABLES THETA TOE-OFF,
THETA TOE-DOWN, THETA LATCH. ONLY POSITION
FEEDBACK FROM THE FIXED CRANK ANGULAR POSITION
OF THE MIDDLE LEG OF THE TRIPOD IS REQUIRED
-----------------------------------------------*)

USES APPLESTUFF;

(*--LIBRARY OF THE APPLE--*)

LABEL 1,2,3;

VAR GDS,BK,GD,GDC,GDC2,THTD,THTD,THTB:INTEGER;

(* VARIABLES REQUIRED *)

(*
GDS...TEMPORARY STORAGE VARIABLE
BK....STATE OF THE SOLENOID ACTUATORS
GD....GROUND LEG FLAG
GDC...COMPLIMENT OF GROUND LEG FLAG<RETURN LEG>
GDC2...COMPLIMENT OF GROUND LEG FLAG + 2
THTD..TOE-OFF ENCODER READING
THTD..TOE-DOWN ENCODER READING
THTB..LATCH ENCODER READING *)

PROCEDURE POKE(SELECT,VALUE:INTEGER);
EXTERNAL;

(* POKE THE MEMORY LOCATION SELECT
 WITH VALUE.*)

PROCEDURE POKE1(SELECT,VALUE:INTEGER);
EXTERNAL;

(* POKE THE MEMORY LOCATION OF THE
 SOLENOID ACTUATORS TO TURN THEM ON
 AND OFF *)

FUNCTION PEAK(SELECT:INTEGER): INTEGER;
EXTERNAL;

(* READS THE CONTENTS OF THE MEMORY LOCATION
 SELECT *)
FUNCTION ENCODE(SELECT:INTEGER):INTEGER;
EXTERNAL;

(* READS THE MEMORY LOCATIONS OF THE ENCODERS
AND CONVERTS GRAY CODE TO BINARY CODE *)

PROCEDURE IN109;

(* THIS ROUTINE INITIALIZES THE CONTROL LINES
OF THE D109 FOR ACCESS TO 32 I/O LINES *)

VAR SELECT,VALUE:INTEGER;
BEGIN
VALUE:=0;
FOR SELECT:= 0 TO 31 DO
BEGIN
POKE(SELECT,VALUE) (* DIABLES ALL FUNCTIONS *)
END
END;

PROCEDURE INDUWE( VAR THTO,THTD,THTB,
BK,GD,GDC:INTEGER);

(* THIS ROUTINE INITIALIZES THE DUWE *)

(* VARIABLES
THTO......THETA TOEOFF
THTD......THETA TOEDOWN
THTB......THETA LATCH
BK........BRAKE FLAG
GD........GROUND LEG FLAG
GDC........COMPLIMENT OF GD *)
BEGIN
WRITELN('INPUT THE FOLLOWING PARAMETERS');
WRITELN;
WRITELN;
WRITE('THETA TOEOFF....');
READLN(THTO);
WRITE('THETA TOEDOWN...');
READLN(THTD);
WRITE('THETA BRAKE.....');
READLN(THTB);

BK:=3; (*START WITH RIGHT MID LEG AT *)
GD:=0; (*END OF STROKE AND LEFT LEG*)
GDC:=1; (*AT BEGINNING OF STROKE*)

314
POKE1(0,1); (*TURN ALL SOLENOIDS OFF*)
POKE1(1,1); (* 0,1...LEG SOLENOIDS*)
POKE1(2,1); (* 2,3...BRAKE SOLENOIDS*)
POKE1(3,1)
END;

(*------------------------------------------------*)

(* MAIN PROGRAM *)

(*------------------------------------------------*)

BEGIN

(* -- INITIALIZE DIO9 & DUWE -- *)

INDIO9;
INDUWE(THTO,THTD,THTB,BK,GD,GDC);

WRITELN('HIT CR TO INITIATE PROGRAM');
READLN(GDS); (* DUMMY VAR. START-UP*)

REPEAT

WRITELN('BK = ',BK,' GD = ',GD,' GDC = ',GDC);

CASE BK OF

(* -- LEG RETURNING BACK -- *)

1: BEGIN
IF(ENCODE(GDC)<=THTB) THEN
BEGIN
BK:=2;
GDC2:=GDC+2; (* LATCH LEG AND *)
POKE1(GDC2,0); (* AND EXTEND *)
POKE1(GDC,1) (* ADJUSTABLE LINK *)
END
END;

(* -- LEG LATCHED BRAKE ON -- *)

2: BEGIN
IF(ENCODE(GD)>=THTO) THEN
BEGIN
GDC2:=GDC+2; (* RELEASE LATCHED *)
POKE1(GDC2,1); (* LEG *)
BK:=3
END
END;

315
(* -- LEG SWINGING DOWN -- *)

3: BEGIN
   IF(ENCODE(GDC)=THTD) THEN
      BEGIN
         POKEi(GD,0); (* LIFT SUPPORT LEG *)
         GDS:=GD; (* FOR RETURN PHASE *)
         GD:=GDC;
         GDC:=GDS;
         BK:=1
      END
   END

UNTIL KEYPRESS;

(* -- END OF MAIN PROGRAM -- *)

END.
; MACRO POPS 16 BIT ARGUMENT
;

; .MACRO POP
PLA
STA %1
PLA
STA %1+1
.ENDM

; .FUNC PEAK,1 ; ONE WORD OF PARAMETERS
;
; EXAMINES CONTENTS OF SELECTED MEMORY LOCATION
; C400-->C4FF BY PASSING SELECT = 0-->255 TO LOCATE
;
; FUNCTION PEAK(SELECT: INTEGER): INTEGER;
;
RETURN .EQU 0 ; TEMP VAR FOR RETURN ADDR

POP RETURN ; SAVE PASCAL RETURN ADDR
PLA ; DISCARD 4 BYTES STACK BIAS
PLA ; REQUIRED FOR FUNC ONLY
PLA
PLA
PLA ; GET L.S.BYTE SELECT PARAM
TAX ; STORE LSB IN REG X
PLA ; DISCARD MSB SELECT PARAM
LDA #0
PHA ; PUSH MSB OF RETURN VALUE = 0
LDA 0c400,X
PHA
LDA RETURN+1
PHA
LDA RETURN
PHA
RTS ; RETURN TO PASCAL CALLER
.PROC POKE,2 ; TWO WORDS OF PARAM

;-----------------------------------------------
; ROUTINE TO WRITE TO A MEMORY LOCATION
; VALUE = 0 --> 255
;
; PROCEDURE POKE(SELECT, VALUE: INTEGER);
;-----------------------------------------------
RETURN .EQU 0 ; TEMP RETURN ADDR

POP RETURN ; SAVE PASCAL RETURN ADDRESS
PLA ; GET LSB VALUE PARAM
TAY ; PUT LSB VALUE IN REG Y
PLA ; DISREGARD MSB VALUE
PLA ; GET LSB OF SELECT PARAM
TAX ; PUT LSB SELECT IN REG X
PLA ; DISREGARD MSB OF SELECT
TYA ; LOAD ACC WITH REG Y
STA 0C400,X ; PUT VALUE IN LOCATION
LDA RETURN +1 ; RESTORE PASCAL RETURN ADDR
PHA
LDA RETURN
PHA
RTS ; GO BACK TO PASCAL
.PROC POKE1,2 ;Two Words of Parameters

; Routine to Set or Clear One of the Solenoid Switches on the P8 Optical Isolator
; Bits. Data = 1 to Clear (Reverse Logic)
; Data = 0 to Set
; Procedure POKE1(SELECT,DATA: Integer);

RETURN .EQU 0 ;Temp Return Addr
POP RETURN ;Save Pascal Return Address
PLA ;Get LSB Data 0 or 1
LSR A ;Save Data in Carry
PLA ;Discard MSB Boolean Data
PLA ;Get LSB Select
AND #03 ;Treat It Mod 4
ROL A ;Double, Add Data for Index
TAY ;Put I/O Strobe Index in Y
LDA 0C058,Y ;Activate I/O Strobe
PLA ;Discard MSB Select Param
LDA RETURN+1 ;Restore Pascal Return Addr
PHA
LDA RETURN
PHA
RTS ;Return to Pascal
.FUNC ENCODE,1 ;ONE WORD OF PARAMETERS

; READS THE VALUE OF AN ENCODER. AUTOMATICALLY
; CONVERTS FROM GRAY TO BINARY. ENCODER MUST
; BE ATTACHED TO MEM. LOCAT. C400-->C4FF. LOCAT.
; SELECT = 0-->255. EVEN SELECT FOR CW READ
; ODD SELECT FOR CCW READ.

; FUNCTION ENCODE(SELECT: INTEGER): INTEGER;

RETURN .EQU 0 ;TEMP VAR FOR RETURN ADDR

POP RETURN ;SAVE PASCAL RETURN ADDR
PLA ;DISCARD 4 BYTES STACK BIAS
PLA ;REQUIRED FOR FUNC ONLY
PLA
PLA
PLA
STP 042 ;STORE LSB IN MEM.
LDX 042 ;STORE SELECT IN X
LDY 042 ;+ Y REGISTERS
PLA ;DISCARD MSB SELECT PARAM
LDA 0C400,X ;LOAD ACC WITH ENCODE READ.
EOR #0FF ;COMPLIMENT ARG
TAX ;STORE NEW ARG IN X
PLA ;DISREGARD MSB ARG
LDA #080 ;LOAD ACC 10000000
CLC ;CLEAR CARRY STATUS
STA 041 ;STORE ACC IN MEM
LOOP1 BCC CARCL ;BRANCH IF CARRY CLEAR
TXA ;STORE X IN ACC
EOR 041 ;COMPLIMENT DESIRED BIT
TAX ;RET ACC TO X
CARCL TXA ;PUT X BACK INTO ACC
AND #041 ;DETERMINE BIT STATUS
BNE CARR1 ;BRANCH IF RESULT > 0
LDA #0 ;LOAD ACC WITH 0
STA 040 ;STORE 0 IN 0040
AND #0 ;RESULT = 0
BEQ ROLME ;ALWAYS GO TO ROLME
CARR1 LDA #1 ;LOAD 1 INTO ACC
STA 040 ;STORE ACC IN 0040
ROLME CLC ;CLEAR CARRY BEFORE ROLE
ROR 041 ;ROLL MEM TO RIGHT
LDA #0FF ;LOAD ACC WITH 11111111
AND #041 ;CHECK AND SEE IF DONE
BEQ DONE ;BRANCH IF DONE

320
ROR 040 ;RESTORE CARRY
AND #0
BEQ LOOP1 ;BRANCH TO LOOP1
DONE
CLC ;CLEAR CARRY
TYA ;RECALL SELECT
ROR A ;CHECK SELECT FOREVEN OR ODD
BCS NOCCW ;BRANCH ON ODD SELECT
STX 040 ;STORE RESULT IN 040
LDA #0FF ;LOAD ACC. WITH 255
SEC ;SET CARRY TO 1 FOR SUBTRACT
SBC 040 ;SUBTRACT RESULT FROM 255
TXA ;TRANS NEW RESULT TO X
NOCCW
LDA #0 ;RETURN VALUE
PHA
TXA
PHA
LDA RETURN+1
PHA
LDA RETURN
PHA
RTS ;RETURN TO PASCAL

.END
APPENDIX F

ADDITIONAL FIGURES AND TABLES
LOAD RING CALIBRATION CURVE
15V EXCITATION

VOLTAGE OUT (mV)

32.3
30.0
20.0
10.0
5.0
0.0

FORCE ($1D_f$)

0.0  5.0  10.0  15.0  20.0  25.0

LOAD RING FORCE TRANSDUCER CALIBRATION CURVE
<table>
<thead>
<tr>
<th>AVERAGE CART SPEED (M/SEC) (IN/SEC)</th>
<th>DUTY FACTOR ( \beta )</th>
<th>ROTORY ACT. MOTOR</th>
<th>LINEAR ACT. MOTOR</th>
<th>BRAKE SOLENOID</th>
<th>MOTOR DRIVE CIRCUITS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AVE PWR (WATTS)</td>
<td>ENERGY (JOULES)</td>
<td>AVE PWR (WATTS)</td>
<td>ENERGY (JOULES)</td>
<td>AVE PWR (WATTS)</td>
</tr>
<tr>
<td>0.074 2.9</td>
<td>0.50</td>
<td>4.9</td>
<td>1.2</td>
<td>14.65</td>
<td>3.2</td>
</tr>
<tr>
<td>0.097 3.8</td>
<td>0.52</td>
<td>5.8</td>
<td>1.4</td>
<td>11.90</td>
<td>2.7</td>
</tr>
<tr>
<td>0.141 5.6</td>
<td>0.52</td>
<td>9.2</td>
<td>2.3</td>
<td>12.70</td>
<td>2.5</td>
</tr>
<tr>
<td>0.211 8.3</td>
<td>0.49</td>
<td>21.0</td>
<td>4.7</td>
<td>18.60</td>
<td>2.4</td>
</tr>
<tr>
<td>0.245 9.6</td>
<td>0.47</td>
<td>24.7</td>
<td>3.5</td>
<td>12.40</td>
<td>2.9</td>
</tr>
<tr>
<td>0.267 10.5</td>
<td>0.46</td>
<td>29.2</td>
<td>7.2</td>
<td>22.90</td>
<td>2.2</td>
</tr>
</tbody>
</table>

* AVE PWR = AVERAGE POWER

MONOPOD ENERGY AUDIT; ONE CYCLE, FORWARD STRIDE
ADAPTED FROM VONKHOUT [3]
APPLE COMPUTER WIRING DIAGRAM SHOWING
GAME I/O CONNECTOR

325
**FEATURES**

- Single Apple II card providing 32 digital I/O lines.
- All lines bidirectional: input or output.
- Four 16-bit programmable timer-counters.
- Two 8-bit shift registers.
- Complete Handshake capability for data transfer.
- Full interrupt control capability.

---

*Apple II is a trademark of Apple Computer Inc.*

**SCHEMATIC OF DI09 INTERFACE OPERATION**

326
LAYOUT OF OPTICAL INTERFACE FOR
SWITCHING THE SOLENOIDS

327