<table>
<thead>
<tr>
<th>UNCLASSIFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD NUMBER</td>
</tr>
<tr>
<td>ADB015216</td>
</tr>
<tr>
<td>LIMITATION CHANGES</td>
</tr>
</tbody>
</table>

**TO:**
Approved for public release; distribution is unlimited.

**FROM:**
Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; NOV 1975. Other requests shall be referred to Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH 45433.

**AUTHORITY**
AFFDL ltr 2 May 1979
THIS REPORT HAS BEEN DELIMITED
AND CLEARED FOR PUBLIC RELEASE
UNDER DOD DIRECTIVE 5200.20 AND
NO RESTRICTIONS ARE IMPOSED UPON
ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.
ACTIVE CONTROL SYNTHESIS FOR FLEXIBLE VEHICLES
Volume II KONPACT Program Listing

HONEYWELL
SYSTEMS & RESEARCH CENTER
2600 RIDGWAY PARKWAY
MINNEAPOLIS, MINNESOTA 55413

JULY 1976

TECHNICAL REPORT AFFDL-TR-75-146 FOR PERIOD APRIL 1975 - APRIL 1976

Distribution limited to U. S. Government agencies only; test and evaluation statement applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FCG), Wright-Patterson Air Force Base, Ohio 45433.

COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

Prepared for
U.S. AIR FORCE FLIGHT DYNAMICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433
NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Distribution limited to U. S. Government agencies only; test and evaluation statements applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FGC), Wright-Patterson Air Force Base, Ohio 45433.

This technical report has been reviewed and is approved for publication.

Charles R. Stockdale
Project Engineer

FOR THE COMMANDER

Evard H. Flinn, Chief
Control Criteria Branch
Air Force Flight Dynamics Laboratory

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.
**Title:** ACTIVE CONTROL SYNTHESIS FOR FLEXIBLE VEHICLES, Volume II: KONPACT Program Listing

**Authors:** A. F. Konar, J. K. Mahesh, C. R. Stone, M. Hank

**Performing Organization:** Honeywell Inc., Systems and Research Center, 2600 Ridgway Parkway N.E., Minneapolis, Minnesota 55413

**Controlling Office:** U.S. Air Force Flight Dynamics Laboratory, Wright Patterson Air Force Base, Ohio 45433

**Report Date:** January 1976

**Number of Pages:** 308

**Distribution Statement:** Distribution limited to U.S. Government agencies only; test and evaluation statement applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FGC), Wright-Patterson Air Force Base, Ohio 45433.

**Key Words:**
- Active Control
- C-5A
- Overlay
- CCV (Control Configured Vehicles)
- Optimal Control
- Variable Dimensioning
- Flight
- FLEXSTAB
- Dynamic Data Storage
- Flexible Vehicle
- Modeling
- PRECOMPILER

**Abstract:** KONPACT is a system of computer programs that will design optimal or suboptimal control systems especially for aircraft with lightly damped modes. This program represents advanced computational techniques to perform modern control synthesis, analysis and design of automatic control systems. These programs augment aircraft mathematical models produced from such advanced programs as the FLEXSTAB Level 2.01.00 with control system dynamics and then design and analyze...
The KONPACT Program Listings is the second volume of report prepared under contract F33615-75-C-3046. This report contains the program listings of KONPACT.
FOREWORD

The research described in this report was prepared by Honeywell Inc., Minneapolis, Minnesota 55413, under Air Force Contract F33615-75-C-3046. It was initiated under the AFFDL task number 82190221, "Optimal Control of Flexible Aircraft," project number 8219 "Stability and Control of Aerospace Vehicles." This work was directed by the Control Criteria Branch (FGC), Flight Control Division of the Air Force Flight Dynamics Laboratory and was administered by Mr. Charles R. Stockdale of the Control Criteria Branch. Special thanks to Mr. Robert C. Schwanz of FGC and Mr. Gary Grimes of ASD/ADDP for their continued support toward this contract.

The technical work reported in this volume was conducted by the Research Department at the Systems and Research Center of Honeywell Inc. Dr. A. F. Konar was the Honeywell Program Manager and the principal investigator on this contract. He was assisted by Mr. C. R. Stone, Dr. J. K. Mahesh, and Miss M. Hank. This report covers work from April 1975 to April 1976.

The work under this contract was reported in three volumes entitled, "Active Control Synthesis for Flexible Vehicles."

Volume I. KONPACT Theoretical Description  AD-B015 1984
Volume II. KONPACT Program Listing
Volume III. KONPACT Users Manual  AD-B015 0254
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>3</td>
</tr>
<tr>
<td>III</td>
<td>10</td>
</tr>
</tbody>
</table>

## I INTRODUCTION

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

## II DESCRIPTION OF KONPACT PROGRAMS

<table>
<thead>
<tr>
<th>Overlay Organization</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Flow</td>
<td>4</td>
</tr>
<tr>
<td>Variable Dimensioning</td>
<td>7</td>
</tr>
</tbody>
</table>

## III MODELING PROGRAM (KONPACT-1)

<table>
<thead>
<tr>
<th>Overlay Structure</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description of Main Programs</td>
<td>10</td>
</tr>
<tr>
<td>Program MAIN</td>
<td>10</td>
</tr>
<tr>
<td>Program MAIN1</td>
<td>18</td>
</tr>
<tr>
<td>Program MAIN2</td>
<td>18</td>
</tr>
<tr>
<td>Program MAIN3</td>
<td>18</td>
</tr>
<tr>
<td>Program MAIN4</td>
<td>22</td>
</tr>
<tr>
<td>Program MAIN5</td>
<td>22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description of Basic Subroutines</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subroutine KORG1</td>
<td>22</td>
</tr>
<tr>
<td>Subroutine NAMEL</td>
<td>26</td>
</tr>
<tr>
<td>Subroutine STAMK1</td>
<td>26</td>
</tr>
<tr>
<td>Subroutine SIMK1</td>
<td>26</td>
</tr>
<tr>
<td>Subroutine STAMK2</td>
<td>26</td>
</tr>
<tr>
<td>Subroutine SIMKT</td>
<td>28</td>
</tr>
<tr>
<td>Subroutine TRANSK</td>
<td>28</td>
</tr>
<tr>
<td>Subroutine DFN</td>
<td>28</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Subroutine PHERR</td>
<td>28</td>
</tr>
<tr>
<td>Subroutine STAMK3</td>
<td>30</td>
</tr>
<tr>
<td>Subroutine SIMK</td>
<td>30</td>
</tr>
<tr>
<td>Subroutine QUADK</td>
<td>30</td>
</tr>
<tr>
<td>Subroutine STAMK4</td>
<td>30</td>
</tr>
<tr>
<td>Subroutine SIMK2</td>
<td>34</td>
</tr>
<tr>
<td>Subroutine CONDK</td>
<td>34</td>
</tr>
<tr>
<td>Subroutine MNAME</td>
<td>34</td>
</tr>
<tr>
<td>Subroutine IMRATE</td>
<td>34</td>
</tr>
<tr>
<td>Subroutine DIFFK</td>
<td>35</td>
</tr>
<tr>
<td>Subroutine REDUCE</td>
<td>35</td>
</tr>
<tr>
<td>Subroutine SCAL</td>
<td>35</td>
</tr>
<tr>
<td>Subroutine SHUFF</td>
<td>35</td>
</tr>
<tr>
<td>Subroutine SHUF1</td>
<td>35</td>
</tr>
<tr>
<td>Subroutine SHUF2</td>
<td>36</td>
</tr>
<tr>
<td><strong>Description of Auxiliary Subroutines</strong></td>
<td></td>
</tr>
<tr>
<td>Subroutine QDIO</td>
<td>36</td>
</tr>
<tr>
<td>Subroutine IDRO</td>
<td>36</td>
</tr>
<tr>
<td>Subroutine RSDRD</td>
<td>36</td>
</tr>
<tr>
<td>Subroutine SDRD</td>
<td>37</td>
</tr>
<tr>
<td>Subroutine FILE</td>
<td>37</td>
</tr>
<tr>
<td>Subroutine TPR</td>
<td>37</td>
</tr>
<tr>
<td>Subroutine HPR</td>
<td>37</td>
</tr>
<tr>
<td>Subroutine IDPR</td>
<td>37</td>
</tr>
</tbody>
</table>

vi
TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subroutine MPRS</td>
<td>38</td>
</tr>
<tr>
<td>Subroutine MPRS1</td>
<td>38</td>
</tr>
<tr>
<td>Subroutine ZERO</td>
<td>38</td>
</tr>
<tr>
<td>Subroutine INPT</td>
<td>38</td>
</tr>
<tr>
<td>Subroutine INPT1</td>
<td>38</td>
</tr>
<tr>
<td>Subroutine DEBUG</td>
<td>39</td>
</tr>
<tr>
<td>Subroutine ERRM</td>
<td>39</td>
</tr>
<tr>
<td>Subroutine DERRM</td>
<td>39</td>
</tr>
<tr>
<td>Subroutine DERMS</td>
<td>39</td>
</tr>
<tr>
<td>Subroutine SHIFT</td>
<td>39</td>
</tr>
<tr>
<td>Subroutine TDINVR</td>
<td>39</td>
</tr>
</tbody>
</table>

IV DESIGN PROGRAM (KONPACT-2) | 40
Overlay Structure | 40
Description of Main Programs | 45
  Program MAIN | 45
  Program DIAK | 45
  Program FFOC | 45
  Program DATAK | 45
Description of Basic Subroutines | 47
  Subroutine KORG2 | 47
  Subroutine TIMER | 47
  Subroutine SGUST | 47
  Subroutine CAL1 | 47
  Subroutine STRIC | 47
  Subroutine SHUFL | 47
### TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subroutine SHUF</td>
<td>48</td>
</tr>
<tr>
<td>Subroutine RESP</td>
<td>48</td>
</tr>
<tr>
<td>Subroutine COVAR</td>
<td>48</td>
</tr>
<tr>
<td>Subroutine COSTAT</td>
<td>48</td>
</tr>
<tr>
<td>Subroutine TRANS</td>
<td>49</td>
</tr>
<tr>
<td>Subroutine UNSCR</td>
<td>49</td>
</tr>
<tr>
<td>Subroutine GCAL</td>
<td>49</td>
</tr>
<tr>
<td>Subroutine CAL</td>
<td>49</td>
</tr>
<tr>
<td>Subroutine DDIAK</td>
<td>49</td>
</tr>
<tr>
<td>Subroutine DFFOC</td>
<td>50</td>
</tr>
<tr>
<td>Subroutine DLSA</td>
<td>50</td>
</tr>
<tr>
<td>Subroutine FINK</td>
<td>50</td>
</tr>
<tr>
<td>Description of Auxiliary Subroutines</td>
<td>50</td>
</tr>
<tr>
<td>Subroutine MP</td>
<td>50</td>
</tr>
<tr>
<td>Subroutine OUTP</td>
<td>51</td>
</tr>
<tr>
<td>Subroutine POLES</td>
<td>51</td>
</tr>
<tr>
<td>Subroutine HESSEN</td>
<td>51</td>
</tr>
<tr>
<td>Subroutine QRCALL</td>
<td>51</td>
</tr>
<tr>
<td>Subroutine QR</td>
<td>51</td>
</tr>
<tr>
<td>Subroutine INPTM</td>
<td>51</td>
</tr>
<tr>
<td>Subroutine WTP</td>
<td>52</td>
</tr>
<tr>
<td>Function GRAN</td>
<td>52</td>
</tr>
</tbody>
</table>

V CONCLUSIONS AND RECOMMENDATIONS

Recommendations for Future Software Development Work 53
Conclusions 53
TABLE OF CONTENTS (concluded)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDIX</td>
<td>303</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>308</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interface Between LSA and KONPACT Programs</td>
<td>54</td>
</tr>
<tr>
<td>2</td>
<td>Overlay Structure of KONPACT-1</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Overlay Structure of KONPACT-2</td>
<td>56</td>
</tr>
<tr>
<td>4</td>
<td>Overlay Structure and Subroutines in KONPACT-1</td>
<td>57</td>
</tr>
<tr>
<td>5</td>
<td>Program MAIN Program Listing</td>
<td>58</td>
</tr>
<tr>
<td>6</td>
<td>Program MAIN1 Program Listing</td>
<td>59</td>
</tr>
<tr>
<td>7</td>
<td>Program MAIN2 Program Listing</td>
<td>60 - 61</td>
</tr>
<tr>
<td>8</td>
<td>Program MAIN3 Program Listing</td>
<td>62 - 63</td>
</tr>
<tr>
<td>9</td>
<td>Program MAIN4 Program Listing</td>
<td>64</td>
</tr>
<tr>
<td>10</td>
<td>Program MAIN5 Program Listing</td>
<td>65 - 66</td>
</tr>
<tr>
<td>11</td>
<td>Subroutine KORG1 Flow Chart</td>
<td>67 - 69</td>
</tr>
<tr>
<td>12</td>
<td>Subroutine KORG1 Program Listing</td>
<td>70 - 74</td>
</tr>
<tr>
<td>13</td>
<td>Subroutine NAMEL Flow Chart</td>
<td>75</td>
</tr>
<tr>
<td>14</td>
<td>Subroutine NAMEL Program Listing</td>
<td>76 - 82</td>
</tr>
<tr>
<td>15</td>
<td>Subroutine STAMK1 Flow Chart</td>
<td>83</td>
</tr>
<tr>
<td>16</td>
<td>Subroutine STAMK1 Program Listing</td>
<td>84 - 86</td>
</tr>
<tr>
<td>17</td>
<td>Subroutine SIMK1 Flow Chart</td>
<td>87</td>
</tr>
<tr>
<td>18</td>
<td>Subroutine SIMK1 Program Listing</td>
<td>88 - 101</td>
</tr>
</tbody>
</table>

x
## LIST OF FIGURES (continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Subroutine STAMK2 Flow Chart</td>
<td>102</td>
</tr>
<tr>
<td>20</td>
<td>Subroutine STAMK2 Program Listing</td>
<td>103 - 107</td>
</tr>
<tr>
<td>21</td>
<td>Subroutine SIMKT Flow Chart</td>
<td>108</td>
</tr>
<tr>
<td>22</td>
<td>Subroutine SIMKT Program Listing</td>
<td>109 - 113</td>
</tr>
<tr>
<td>23</td>
<td>Subroutine TRANSK Flow Chart</td>
<td>114</td>
</tr>
<tr>
<td>24</td>
<td>Subroutine TRANSK Program Listing</td>
<td>115</td>
</tr>
<tr>
<td>25</td>
<td>Subroutine DFN Flow Chart</td>
<td>116</td>
</tr>
<tr>
<td>26</td>
<td>Subroutine DFN Program Listing</td>
<td>117 - 119</td>
</tr>
<tr>
<td>27</td>
<td>Subroutine PHERR Flow Chart</td>
<td>120</td>
</tr>
<tr>
<td>28</td>
<td>Subroutine PHERR Program Listing</td>
<td>121</td>
</tr>
<tr>
<td>29</td>
<td>Subroutine STAMK3 Flow Chart</td>
<td>122</td>
</tr>
<tr>
<td>30</td>
<td>Subroutine STAMK3 Program Listing</td>
<td>123 - 127</td>
</tr>
<tr>
<td>31</td>
<td>Subroutine SIMK Flow Chart</td>
<td>128</td>
</tr>
<tr>
<td>32</td>
<td>Subroutine SIMK Program Listing</td>
<td>129 - 136</td>
</tr>
<tr>
<td>33</td>
<td>Subroutine QUADK Flow Chart</td>
<td>137</td>
</tr>
<tr>
<td>34</td>
<td>Subroutine QUADK Program Listing</td>
<td>138 - 139</td>
</tr>
<tr>
<td>35</td>
<td>Subroutine STAMK4 Flow Chart</td>
<td>140</td>
</tr>
<tr>
<td>36</td>
<td>Subroutine STAMK4 Program Listing</td>
<td>141 - 143</td>
</tr>
<tr>
<td>Figure</td>
<td>Subroutine/Program</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>37</td>
<td>SIMK2 Flow Chart</td>
<td>144</td>
</tr>
<tr>
<td>38</td>
<td>SIMK2 Program Listing</td>
<td>145 - 147</td>
</tr>
<tr>
<td>39</td>
<td>CONDK Flow Chart</td>
<td>148 - 149</td>
</tr>
<tr>
<td>40</td>
<td>CONDK Program Listing</td>
<td>150 - 155</td>
</tr>
<tr>
<td>41</td>
<td>MNAME Flow Chart</td>
<td>156</td>
</tr>
<tr>
<td>42</td>
<td>MNAME Program Listing</td>
<td>157 - 161</td>
</tr>
<tr>
<td>43</td>
<td>IMRATE Flow Chart</td>
<td>162</td>
</tr>
<tr>
<td>44</td>
<td>IMRATE Program Listing</td>
<td>163 - 164</td>
</tr>
<tr>
<td>45</td>
<td>DIFFK Flow Chart</td>
<td>165</td>
</tr>
<tr>
<td>46</td>
<td>DIFFK Program Listing</td>
<td>166 - 167</td>
</tr>
<tr>
<td>47</td>
<td>REDUCE Flow Chart</td>
<td>168</td>
</tr>
<tr>
<td>48</td>
<td>REDUCE Program Listing</td>
<td>169 - 171</td>
</tr>
<tr>
<td>49</td>
<td>SCAL Flow Chart</td>
<td>172</td>
</tr>
<tr>
<td>50</td>
<td>SCAL Program Listing</td>
<td>173</td>
</tr>
<tr>
<td>51</td>
<td>SHUFF Flow Chart</td>
<td>174</td>
</tr>
<tr>
<td>52</td>
<td>SHUFF Program Listing</td>
<td>175</td>
</tr>
<tr>
<td>53</td>
<td>SHUFF1 Flow Chart</td>
<td>176</td>
</tr>
<tr>
<td>54</td>
<td>SHUFF1 Program Listing</td>
<td>177</td>
</tr>
<tr>
<td>55</td>
<td>SHUF2 Flow Chart</td>
<td>178</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>56</td>
<td>Subroutine SHUF2 Program Listing</td>
<td>179</td>
</tr>
<tr>
<td>57</td>
<td>Subroutine QDIO Flow Chart</td>
<td>180</td>
</tr>
<tr>
<td>58</td>
<td>Subroutine QDIO Program Listing</td>
<td>181 - 183</td>
</tr>
<tr>
<td>59</td>
<td>Subroutine IDRO Flow Chart</td>
<td>184</td>
</tr>
<tr>
<td>60</td>
<td>Subroutine IDRO Program Listing</td>
<td>185 - 187</td>
</tr>
<tr>
<td>61</td>
<td>Subroutine RSDRD Flow Chart</td>
<td>188</td>
</tr>
<tr>
<td>62</td>
<td>Subroutine RSDRD Program Listing</td>
<td>189 - 192</td>
</tr>
<tr>
<td>63</td>
<td>Subroutine SDRD Flow Chart</td>
<td>193</td>
</tr>
<tr>
<td>64</td>
<td>Subroutine SDRD Program Listing</td>
<td>194 - 195</td>
</tr>
<tr>
<td>65</td>
<td>Subroutine FILE Flow Chart</td>
<td>196</td>
</tr>
<tr>
<td>66</td>
<td>Subroutine FILE Program Listing</td>
<td>197 - 198</td>
</tr>
<tr>
<td>67</td>
<td>Subroutine TPR Flow Chart</td>
<td>199</td>
</tr>
<tr>
<td>68</td>
<td>Subroutine TPR Program Listing</td>
<td>200 - 201</td>
</tr>
<tr>
<td>69</td>
<td>Subroutine HPR Program Listing</td>
<td>202</td>
</tr>
<tr>
<td>70</td>
<td>Subroutine IDPR Program Listing</td>
<td>203</td>
</tr>
<tr>
<td>71</td>
<td>Subroutine MPfIS Program Listing</td>
<td>204 - 205</td>
</tr>
<tr>
<td>72</td>
<td>Subroutine MPRS1 Program Listing</td>
<td>206</td>
</tr>
<tr>
<td>73</td>
<td>Subroutine ZERO Program Listing</td>
<td>207</td>
</tr>
<tr>
<td>74</td>
<td>Subroutine INPT Program Listing</td>
<td>208</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>75</td>
<td>Subroutine INPT1 Program Listing</td>
<td>209</td>
</tr>
<tr>
<td>76</td>
<td>Subroutine DEBUG Program Listing</td>
<td>210</td>
</tr>
<tr>
<td>77</td>
<td>Subroutine ERM Program Listing</td>
<td>211</td>
</tr>
<tr>
<td>78</td>
<td>Subroutine DERRM Program Listing</td>
<td>212</td>
</tr>
<tr>
<td>79</td>
<td>Subroutine DERRMS Program Listing</td>
<td>213</td>
</tr>
<tr>
<td>80</td>
<td>Subroutine SHIFT Program Listing</td>
<td>214</td>
</tr>
<tr>
<td>81</td>
<td>Subroutine TDINVR Program Listing</td>
<td>215-217</td>
</tr>
<tr>
<td>82</td>
<td>Overlay Structure and Subroutines in KONPACT-2</td>
<td>218</td>
</tr>
<tr>
<td>83</td>
<td>Program MAIN Program Listing</td>
<td>219</td>
</tr>
<tr>
<td>84</td>
<td>Program DIAK Program Listing</td>
<td>220-234</td>
</tr>
<tr>
<td>85</td>
<td>Program FFOC Program Listing</td>
<td>235-244</td>
</tr>
<tr>
<td>86</td>
<td>Program DATAK Program Listing</td>
<td>245-246</td>
</tr>
<tr>
<td>87</td>
<td>Subroutine KORG2 Flow Chart</td>
<td>247-248</td>
</tr>
<tr>
<td>88</td>
<td>Subroutine KORG2 Program Listing</td>
<td>249-251</td>
</tr>
<tr>
<td>89</td>
<td>Subroutine TIMER Program Listing</td>
<td>252-256</td>
</tr>
<tr>
<td>90</td>
<td>Subroutine SGUST Program Listing</td>
<td>257</td>
</tr>
<tr>
<td>91</td>
<td>Subroutine CAL1 Program Listing</td>
<td>258-259</td>
</tr>
<tr>
<td>92</td>
<td>Subroutine STRIC Program Listing</td>
<td>260</td>
</tr>
<tr>
<td>93</td>
<td>Subroutine SHUFL Program Listing</td>
<td>261</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>94</td>
<td>Subroutine SHUF Program Listing</td>
<td>262</td>
</tr>
<tr>
<td>95</td>
<td>Subroutine RESP Program Listing</td>
<td>263 - 265</td>
</tr>
<tr>
<td>96</td>
<td>Subroutine COVAR Program Listing</td>
<td>266 - 267</td>
</tr>
<tr>
<td>97</td>
<td>Subroutine COSTAT Program Listing</td>
<td>268</td>
</tr>
<tr>
<td>98</td>
<td>Subroutine TRANS Program Listing</td>
<td>269</td>
</tr>
<tr>
<td>99</td>
<td>Subroutine UNSCR Program Listing</td>
<td>270</td>
</tr>
<tr>
<td>100</td>
<td>Subroutine GCAL Program Listing</td>
<td>271 - 272</td>
</tr>
<tr>
<td>101</td>
<td>Subroutine CAL Program Listing</td>
<td>273 - 274</td>
</tr>
<tr>
<td>102</td>
<td>Subroutine DDIAK Flow Chart</td>
<td>275</td>
</tr>
<tr>
<td>103</td>
<td>Subroutine DDIAK Program Listing</td>
<td>276 - 278</td>
</tr>
<tr>
<td>104</td>
<td>Subroutine DFFOC Flow Chart</td>
<td>279</td>
</tr>
<tr>
<td>105</td>
<td>Subroutine DFFOC Program Listing</td>
<td>280 - 282</td>
</tr>
<tr>
<td>106</td>
<td>Subroutine DLSA Flow Chart</td>
<td>283</td>
</tr>
<tr>
<td>107</td>
<td>Subroutine DLSA Program Listing</td>
<td>284 - 286</td>
</tr>
<tr>
<td>108</td>
<td>Subroutine FINK Flow Chart</td>
<td>287</td>
</tr>
<tr>
<td>109</td>
<td>Subroutine FINK Program Listing</td>
<td>288 - 289</td>
</tr>
<tr>
<td>110</td>
<td>Subroutine MP Program Listing</td>
<td>290</td>
</tr>
<tr>
<td>111</td>
<td>Subroutine OUTP Program Listing</td>
<td>291</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES (concluded)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>112</td>
<td>Subroutine POLES Program Listing</td>
<td>292</td>
</tr>
<tr>
<td>113</td>
<td>Subroutine HESSEN Program Listing</td>
<td>293 - 294</td>
</tr>
<tr>
<td>114</td>
<td>Subroutine QRCALL Program Listing</td>
<td>295 - 296</td>
</tr>
<tr>
<td>115</td>
<td>Subroutine QR Program Listing</td>
<td>297 - 298</td>
</tr>
<tr>
<td>116</td>
<td>Subroutine INPTM Program Listing</td>
<td>299</td>
</tr>
<tr>
<td>117</td>
<td>Subroutine WTP Program Listing</td>
<td>300</td>
</tr>
<tr>
<td>118</td>
<td>Subroutine GRAN Program Listing</td>
<td>301</td>
</tr>
<tr>
<td>119</td>
<td>Program PRECOM Flow Chart</td>
<td>302</td>
</tr>
<tr>
<td>120</td>
<td>Program PRECOM Program Listing</td>
<td>304 - 307</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KONPACT Program Descriptions</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>KONPACT Data Tapes</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Typical Dynamic Storage Map</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>KONPACT-1 Subroutine Summary</td>
<td>11 - 17</td>
</tr>
<tr>
<td>5</td>
<td>Dynamic Storage Map for Program MAIN1</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>Dynamic Storage Map for Program MAIN2</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Dynamic Storage Map for Program MAIN3</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>Dynamic Storage Map for Program MAIN4</td>
<td>23</td>
</tr>
<tr>
<td>9</td>
<td>Dynamic Storage Map for Program MAIN5</td>
<td>24 - 25</td>
</tr>
<tr>
<td>10</td>
<td>Dynamic Storage Map for Program STAMP1</td>
<td>27</td>
</tr>
<tr>
<td>11</td>
<td>Dynamic Storage Map for Program STAMP2</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Dynamic Storage Map for Program STAMP3</td>
<td>31 - 32</td>
</tr>
<tr>
<td>13</td>
<td>Dynamic Storage Map for Program STAMP4</td>
<td>33</td>
</tr>
<tr>
<td>14</td>
<td>KONPACT-2 Subroutine Summary</td>
<td>41 - 44</td>
</tr>
<tr>
<td>15</td>
<td>Dynamic Storage Map for Program DATAK</td>
<td>46</td>
</tr>
</tbody>
</table>
SECTION I

INTRODUCTION

The general objective of this program is to develop techniques and tools necessary for rapid design of an active control system for aircraft with lightly damped structural modes. The synthesis techniques provided here are aimed at reducing the engineering man-hours presently required for flight control system design thus effecting a cost reduction. Improvements in the fatigue life, ride qualities, and/or handling qualities of military aircraft are sought by controlling the lightly damped modes thus improving mission performance.

The present scope of this program is to develop programs to interface the level 2.01.00 FLEXSTAB computer program system with existing Air Force-owned optimal control computer programs. These programs represent advanced computational techniques required to perform quantitative analysis of multi-surface control systems. The resulting interface program system is called "KONPACT - Computer Programs for Active Control Technology." KONPACT provides the capability to model, synthesize, analyze, and design automatic control systems by efficiently working together with FLEXSTAB. It can also be used as a stand-alone program.

The work performed under this contract is reported in three volumes:

Volume I. KONPACT Theoretical Description and Demonstration
Volume II. KONPACT Program Listing
Volume III. KONPACT Users Manual
This document reports the program listings of KONPACT. Complete documentation of KONPACT is beyond the scope of this contract.

Section II presents a brief description of KONPACT programs. The variable dimensioning technique for efficient data storage and memory allocation is discussed here. This approach is used throughout KONPACT-1.

The Modeling Program (KONPACT-1) is described in Section III. The Design Program (KONPACT-2) is described in Section IV. The appendix contains a description of the precompiler program for KONPACT-1.

The analytical techniques and algorithms used in KONPACT are described in Volume I. Volume I also demonstrates how these techniques are applied to flexible aircraft control system design.

User's information on KONPACT is given in Volume III. The input cards are fully described for each program. Brief descriptions of programs and information flow in KONPACT are also presented for completeness. Demonstration examples are included to guide the user in data mechanics.
SECTION II

DESCRIPTION OF KONPACT PROGRAMS

KONPACT is a system of computer programs developed by Honeywell under Air Force Contract No. F33615-75-C-3046. KONPACT uses the state space approach for modeling flight control systems and designs the controllers using optimal control methodology. KONPACT interfaces with the Linear Systems Analysis (LSA) Program of the Level 2 FLEXSTAB Program system developed by Boeing under Air Force Contract No. F33615-72-C-1172 (Reference 1). KONPACT can also be used as a stand-alone program.

KONPACT operates on CDC6000 and CDC7000 series computers and can be easily modified to operate on other computers. KONPACT has been written in Extended Fortran IV language.

In this section, a description of KONPACT programs is presented in terms of overlay organization and information flow.

OVERLAY ORGANIZATION

KONPACT consists of two programs, namely, a modeling program (KONPACT-1) and a design program (KONPACT-2). KONPACT-1 interfaces with FLEXSTAB through the LSA program to obtain the vehicle model and augments the specified dynamics to obtain the state space description (quadruple data) of the flight control system. These data are utilized by KONPACT-2.
which contains the subprograms DIAK and FFOC (documented in Reference 2) to the design of the optimal feedback gains. DIAK stands for Doubly Iterative Algorithm developed by Konar (Reference 5). The DIAK program designs full state feedback optimal controllers. FFOC stands for Fixed Form Optimal Controllers. FFOC stands for Fixed Form Optimal Control developed by Stein and Scharmack (Reference 6). The FFOC program designs reduced state (practical) feedback optimal controllers. KONPACT-2 also interfaces with FLEXSTAB through the LSA program to evaluate performances of the above designed optimal flight control system.

Table 1 provides a brief description of programs KONPACT-1 and KONPACT-2 and their subprograms. The interface between KONPACT and the LSA program is illustrated in Figure 1. The overlay structure of KONPACT-1 program is illustrated in Figure 2. It consists of a main overlay and five primary overlays (Reference 3). The overlay structure of KONPACT-2 program is illustrated in Figure 3. It consists of a main overlay and three primary overlays.

INFORMATION FLOW

The normal sequence for obtaining an overall state space model of a flight control system using the modeling program (KONPACT-1) is as follows:

- The vehicle model is obtained by using either subprogram STAMK1 for LSA data or subprogram STAMK4 for other types of vehicle data.
Table 1. KONPACT Program Descriptions

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>SUBPROGRAM</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>KONPACT-1</td>
<td>STAMK1</td>
<td>State space modeling program</td>
</tr>
<tr>
<td></td>
<td>STAMK2</td>
<td>Obtains state space model from LSA simulator deck data</td>
</tr>
<tr>
<td></td>
<td>STAMK3</td>
<td>Obtains state space model from transfer function data</td>
</tr>
<tr>
<td></td>
<td>STAMK4</td>
<td>Obtains state space model from quadruple data and interconnection data</td>
</tr>
<tr>
<td></td>
<td>CONDK</td>
<td>Obtains state space model from simulation equations (user written)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modifies the state space model by scaling, shuffling, truncating and residualizing the system variables</td>
</tr>
<tr>
<td>KONPACT-2</td>
<td>DATAK</td>
<td>Optimal design program</td>
</tr>
<tr>
<td></td>
<td>DIAK</td>
<td>Preparers data for DIAK, FFOC and LSA programs</td>
</tr>
<tr>
<td></td>
<td>FFOC</td>
<td>Designs full state feedback optimal controllers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designs reduced state (Practical) feedback optimal controllers</td>
</tr>
</tbody>
</table>
The actuator, sensor, controller, implicit and explicit models are obtained by using either subprogram STAMK2 with transfer function input data or subprogram STAMK3 with quadruple input data.

The subsystems defined above are combined to get an overall system by using subprogram STAMK3 with interconnection input data.

The overall system model is conditioned (modified) by scaling and/or shuffling and/or truncating and/or residualizing the variables using the CONDK program. This program also develops the rate of change of response variables when required.

The normal sequence for designing optimal feedback controllers and evaluating the performance of the resulting system using the design program KONPACT-2 is as follows:

- Full state feedback control gains are obtained by varying the quadratic weights and using the DIAK subprogram.
- The resulting full state feedback control gains are reduced to gains only on specified measurements by using the FFOC subprogram.
- The performance of the resulting closed loop system is evaluated using the LSA program.
- The above steps are repeated until a satisfactory design is obtained.

Table 2 describes all the data tapes used in KONPACT-1 and KONPACT-2 programs. The state space model data (quadruple data) and the Name List data are written on tapes QDATA and NDATA, respectively. The
vehicle data (simulator deck data) are written on tape VDATA. The feedback gain data from DIAK and FFOC are written on tapes DDATA and FDATA, respectively. The overall system data in frequency representation form are written on tape SDSTP for use by the LSA program. The DATAK sub-program is used in preparing data tapes for DIAK, FFOC, and LSA.

Table 2. KONPACT Data Tapes

<table>
<thead>
<tr>
<th>TAPE NAME</th>
<th>DESCRIPTION</th>
<th>GENERATING PROGRAM</th>
<th>BENEFITING PROGRAM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDATA</td>
<td>Simulator interface data in the form of card images</td>
<td>LSA</td>
<td>KONPACT-1</td>
</tr>
<tr>
<td>QDATA</td>
<td>Quadruple ((A, B, C, D)) or state variable representation data</td>
<td>KONPACT-1</td>
<td>KONPACT-1 KONPACT-2</td>
</tr>
<tr>
<td>NDATA</td>
<td>Name list data of the state variable representation</td>
<td>KONPACT-1</td>
<td>KONPACT-1</td>
</tr>
<tr>
<td>DDATA</td>
<td>Full state feedback gain data in the form of card images</td>
<td>KONPACT-2</td>
<td>KONPACT-2</td>
</tr>
<tr>
<td>FDATA</td>
<td>Reduced feedback gain data in the form of card images</td>
<td>KONPACT-2</td>
<td>KONPACT-2</td>
</tr>
<tr>
<td>SDSTP</td>
<td>Frequency domain representation of quadruple data</td>
<td>KONPACT-2</td>
<td>LSA</td>
</tr>
</tbody>
</table>

**VARIABLE DIMENSIONING**

Variable dimensioning (dynamic data storage) techniques (Reference 4) are used for efficient data storage. This technique also facilitates changing the amount of allocated (required) storage space by a data card input. In KONPACT the subprogram arrays, whose size depend on the maximum
system dimension inputs, are stored in scratch storage blocks using variable entry points. In the subprograms the arrays are dimensioned with integer variables. These "variable DIMENSION statements" remain unchanged although the amount of required data storage is altered. The maximum size of the scratch storage blocks is specified, in a "fixed DIMENSION statement," in the main program.

The size of storage actually needed by the arrays varies depending on the maximum system dimension inputs. Thus, if the maximum size a user allows for his program changes, there are only the "fixed DIMENSION statements," in the main program, to be changed. Changing the main program of KONPACT-1 is done by a precompiler, as discussed in Section V. The user provides the new maximum system dimensions by data cards. Updating and running with the updated main program are done with control cards in a single run.

In KONPACT programs, four scratch storage blocks, namely S1, S2, S3, and S4 are used. These are specified in the MAIN program of main overlay in labeled COMMON statements under SCI, SC2, SC3, and SC4, respectively. The maximum sizes of these scratch storage blocks are defined there.

The main programs in the primary overlays perform four specific tasks of variable dimensioning. A primary overlay main program first defines the scratch storage blocks under labeled COMMON statements as follows:

```
COMMON/SC1/S1(1)
COMMON/SC2/S2(1)
COMMON/SC3/S3(1)
COMMON/SC4/S4(1)
```
Second, it calculates the start indexes (N1, N2, ... etc.) of the scratch arrays for the stored data as shown in Table 3. Third, it checks the total length occupied by the arrays against the size of the allocated scratch storage blocks. Fourth, it passes the start indexes of the arrays to the subprograms.

Table 3. Typical Dynamic Storage Map

<table>
<thead>
<tr>
<th>Storage Block</th>
<th>Arrays</th>
<th>Block Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 (N1)</td>
<td>V(MAXN)</td>
<td>N1 = 1</td>
</tr>
<tr>
<td>S1 (N2)</td>
<td>W(MAXM)</td>
<td>N2 = N1 + MAXN</td>
</tr>
<tr>
<td>S1 (N3)</td>
<td>F(MAXN, MAXM)</td>
<td>N3 = N2 + MAXM</td>
</tr>
<tr>
<td>S1 (N4)</td>
<td>U(NUM)</td>
<td>N4 = N3 + MAXN * MAXM</td>
</tr>
</tbody>
</table>
SECTION III

MODELING PROGRAM (KONPACT-1)

KONPACT-1 interfaces with FLEXSTAB through the LSA program to obtain the unaugmented vehicle model. It augments this model with the specified dynamics (actuator, sensor, controller, gust, etc.) to obtain the state space description (quadruple data) of the overall flight control system for design.

In this section, a description of the KONPACT-1 program is presented in terms of overlay structure, flow charts, and program listings.

OVERLAY STRUCTURE

The KONPACT-1 program consists of a main overlay and five primary overlays. The overlay structure and the subroutines in each overlay are given in Figure 4. The subroutine summary consisting of name, description, reference, overlay position, and interrelationship is given in Table 4.

DESCRIPTION OF MAIN PROGRAMS

Program MAIN

This is the main program for overlay (0,0). This program assigns the various file numbers used in KONPACT-1. Maximum system dimensions
Table 4. KONPACT-1 Subroutine Summary

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
<th>Flow Chart Fig. #</th>
<th>Program Listing Fig. #</th>
<th>Overlay</th>
<th>Inter-relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>Sets up system dimensions and scratch array dimensions.</td>
<td>5</td>
<td>0, 0</td>
<td>KORGI</td>
<td></td>
</tr>
<tr>
<td>KORGI</td>
<td>Organizes input data and calls the primary overlays.</td>
<td>11</td>
<td>12</td>
<td>0, 0</td>
<td>MAIN</td>
</tr>
<tr>
<td>NAMEL</td>
<td>Reads, prints and updates name list data for the systems.</td>
<td>13</td>
<td>14</td>
<td>0, 0</td>
<td>FILE</td>
</tr>
<tr>
<td>QDIO</td>
<td>Reads and writes quadruple data.</td>
<td>57</td>
<td>58</td>
<td>0, 0</td>
<td>MPBS</td>
</tr>
<tr>
<td>IDRO</td>
<td>Reorganizes the input data.</td>
<td>59</td>
<td>60</td>
<td>0, 0</td>
<td>KORGI</td>
</tr>
<tr>
<td>FILE</td>
<td>Locates and inserts system labels on disc files and writes end of data mark on the disc files.</td>
<td>65</td>
<td>66</td>
<td>0, 0</td>
<td>KORGI</td>
</tr>
<tr>
<td>TPR</td>
<td>Prints transfer function data.</td>
<td>67</td>
<td>68</td>
<td>0, 0</td>
<td>SIMK</td>
</tr>
<tr>
<td>HPR</td>
<td>Prints heading for the system name.</td>
<td>69</td>
<td>0, 0</td>
<td>NAMEL</td>
<td></td>
</tr>
</tbody>
</table>

Calls | Called by |
FILE   |               |
HPR    | KORGI         |
IDRO   | KORGI         |
QDIO   | SIMK          |
FILE   | NAMEL, QDIO, KORGI |
IDRO   | KORGI         |
FILE   | NAMEL, KORGI, QDIO |
IDRO   | KORGI         |
QDIO   | SIMK          |
FILE   | NAMEL, QDIO, KORGI |
IDRO   | KORGI         |
QDIO   | SIMK          |
FILE   | NAMEL, QDIO, KORGI |
IDRO   | KORGI         |
QDIO   | SIMK          |
FILE   | NAMEL, QDIO, KORGI |
IDRO   | KORGI         |
QDIO   | SIMK          |
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
<th>Flow Chart Fig.</th>
<th>Program Listing Fig.</th>
<th>Overlay</th>
<th>Inter-relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDPR</td>
<td>Prints input data.</td>
<td>70</td>
<td>0, 0</td>
<td></td>
<td>KORG1</td>
</tr>
<tr>
<td>MPRS</td>
<td>Prints matrix data on line printer.</td>
<td>71</td>
<td>0, 0</td>
<td></td>
<td>QDI0 SIMK2 SIMK2 SIM1K3 SIMK4 IMRATE REDUCE</td>
</tr>
<tr>
<td>ZERO</td>
<td>Initializes (or zeros) the elements of matrices.</td>
<td>73</td>
<td>0, 0</td>
<td></td>
<td>QUADK SIMK</td>
</tr>
<tr>
<td>INPT</td>
<td>Reads non zero elements of a matrix.</td>
<td>74</td>
<td>0, 0</td>
<td></td>
<td>SIMK2 QUADK SIMK</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Prints a debugging message.</td>
<td>76</td>
<td>0, 0</td>
<td></td>
<td>STAMK1 SIMK1 MAIN2 STAMK2 SIMK7 DFN PHER RTRANX MAIN5 RESPK MNAMN RSDRD SDRB SHIFET</td>
</tr>
<tr>
<td>ERRM</td>
<td>Prints error message.</td>
<td>77</td>
<td>0, 0</td>
<td></td>
<td>RESPK MNAMN RSDRD SDRB</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Flow Chart Fig.</td>
<td>Program Listing Fig.</td>
<td>Overlay</td>
<td>Inter-relationship Calls</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
<td>----------------------</td>
<td>--------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DERRM</td>
<td>Prints an error message when the dimensions for scratch arrays are not sufficient.</td>
<td>70</td>
<td>0.0</td>
<td></td>
<td>MAIN1, SIMK1, MAIN2, STAMK2, MAIN3, STAMK3, MAIN4, STAMK4</td>
</tr>
<tr>
<td>DERRMS</td>
<td>Prints an error message when the system dimensions are not sufficient.</td>
<td>70</td>
<td>0.0</td>
<td></td>
<td>STAMK1</td>
</tr>
<tr>
<td>TDINVR</td>
<td>Inverts a non-singular matrix or solves a set of linear equations.</td>
<td>82</td>
<td>0.0</td>
<td></td>
<td>STAMK1, STAMK2, STAMK3, STAMK4</td>
</tr>
<tr>
<td>MAINI</td>
<td>Sets up block addresses and checks if scratch array size is sufficient.</td>
<td>6</td>
<td>1.0</td>
<td></td>
<td>DERRM</td>
</tr>
<tr>
<td>STAMK1</td>
<td>Obtains state space model from LSA simulator deck data and load equation data (implemented in SIMK1 subroutine).</td>
<td>15</td>
<td>1.0</td>
<td></td>
<td>SIMK1, DERRM, TDINVR, MPRS, NAMEL, DBUG, QDO, HPR</td>
</tr>
<tr>
<td>SIMK1</td>
<td>Reads simulator deck data and load equation data and implements them into simulation equations.</td>
<td>17</td>
<td>1.0</td>
<td></td>
<td>DBUG, DERRM, INFTI, MPRS</td>
</tr>
<tr>
<td>MPRS</td>
<td>Prints simulator deck data and load equation data.</td>
<td>72</td>
<td>1.0</td>
<td></td>
<td>SIMK1</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Flow Chart Fig.</td>
<td>Program Listing Fig.</td>
<td>Overlay</td>
<td>Inter-relationship</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>----------------------</td>
<td>---------</td>
<td>-------------------</td>
</tr>
<tr>
<td>INPT1</td>
<td>Reads simulator deck data and Load equation data.</td>
<td>75</td>
<td>1.0</td>
<td>SIMK1</td>
<td></td>
</tr>
<tr>
<td>MAIN2</td>
<td>Sets up block addresses and checks if scratch array size is sufficient.</td>
<td>7</td>
<td>2.0</td>
<td>DERRM</td>
<td></td>
</tr>
<tr>
<td>STAMK2</td>
<td>Obtains state space model from Transfer function data and connection data (implemented in SIMKT subroutine).</td>
<td>19</td>
<td>2.0</td>
<td>SIMKT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DERRM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DEBUG</td>
<td></td>
</tr>
<tr>
<td>SIMKT</td>
<td>Reads transfer function data and Connection data and implements them into simulation equations.</td>
<td>21</td>
<td>2.0</td>
<td>STAMK2</td>
<td></td>
</tr>
<tr>
<td>TRANSK</td>
<td>Computes state space model for rational transfer functions of up to 5th order.</td>
<td>23</td>
<td>2.0</td>
<td>DEBUG</td>
<td></td>
</tr>
<tr>
<td>DFN</td>
<td>Selects the specified pade approximation to transport (time) delay from a table of pade approximations.</td>
<td>25</td>
<td>2.0</td>
<td>SIMKT</td>
<td></td>
</tr>
<tr>
<td>PHERR</td>
<td>Computes the phase error of pade approximation to transport (time) delay.</td>
<td>27</td>
<td>2.0</td>
<td>DEBUG</td>
<td></td>
</tr>
<tr>
<td>MAIN3</td>
<td>Sets up block addresses and checks if scratch array size is sufficient.</td>
<td>8</td>
<td>1.0</td>
<td>DERRM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>STAMK3</td>
<td></td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Flow Chart Fig.</td>
<td>Program Listing Fig.</td>
<td>Overlay</td>
<td>Inter-relationship</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>---------------------</td>
<td>---------</td>
<td>-------------------</td>
</tr>
<tr>
<td>STAMK3</td>
<td>Obtains state space model from state space data of subsystems and inter-connection data (implemented in SIMK subroutine).</td>
<td>29</td>
<td>30</td>
<td>3.0</td>
<td>SIMK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TDRVR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DERRM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NAMEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QDO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QUADK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DERRMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPBS</td>
</tr>
<tr>
<td>SIMK</td>
<td>Reads state space data of subsystems and inter-connection data and implements them into simulation equations.</td>
<td>31</td>
<td>32</td>
<td>3.0</td>
<td>ZERO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INPUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPBS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FILE</td>
</tr>
<tr>
<td>QUADK</td>
<td>Reads directly the state space data for the system.</td>
<td>33</td>
<td>34</td>
<td>3.0</td>
<td>NAMEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QDO</td>
</tr>
<tr>
<td>MAIN4</td>
<td>Sets up block addresses and checks if scratch array size is sufficient.</td>
<td>9</td>
<td>4.0</td>
<td></td>
<td>DERRM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>STAMK4</td>
</tr>
<tr>
<td>STAMK4</td>
<td>Obtains state space model for the ALDCS controller (implemented in SIMK2 subroutine).</td>
<td>35</td>
<td>36</td>
<td>4.0</td>
<td>SIMK2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DERRM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DERRMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TDRVR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MPBS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NAMEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QDO</td>
</tr>
<tr>
<td>SIMK2</td>
<td>Reads ALDCS controller gains and switch modes and implements ALDCS controller into simulation equations.</td>
<td>37</td>
<td>38</td>
<td>4.0</td>
<td>DEBUG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DERRM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONDK</td>
</tr>
<tr>
<td>MAIN5</td>
<td>Sets up block addresses and checks if scratch array size is sufficient.</td>
<td>10</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Flow Chart Fig.</td>
<td>Program Listing Fig.</td>
<td>Overlay</td>
<td>Inter-relationship</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------</td>
<td>-----------------</td>
<td>----------------------</td>
<td>---------</td>
<td>--------------------</td>
</tr>
<tr>
<td>CONDK</td>
<td>Modifies state space data and name list data according to the design specifications.</td>
<td>39</td>
<td>40</td>
<td>5, 0</td>
<td>MNAME, QDRO, DEBUG, SDRD, SCAL, ERRM, DIFFK, REDUCE, SHUFF, HPR, FILE</td>
</tr>
<tr>
<td>MNAME</td>
<td>Reads, modifies, and prints the name list data for a system.</td>
<td>41</td>
<td>42</td>
<td>5, 0</td>
<td>ERRM, DEBUG, SHIFT, HPR, FILE</td>
</tr>
<tr>
<td>IMRATE</td>
<td>Obtains the implicit model error rates and truncates the implicit model.</td>
<td>43</td>
<td>44</td>
<td>5, 0</td>
<td>TDINVR, MPRS</td>
</tr>
<tr>
<td>DIFFK</td>
<td>Differentiates either a specified response or state of a system.</td>
<td>45</td>
<td>46</td>
<td>5, 0</td>
<td>CONDK</td>
</tr>
<tr>
<td>REDUCE</td>
<td>Residualizes or truncates the state space data of a system.</td>
<td>47</td>
<td>48</td>
<td>5, 0</td>
<td>TDINVR, MPRS</td>
</tr>
<tr>
<td>SCAL</td>
<td>Computes scaled state space data.</td>
<td>49</td>
<td>50</td>
<td>5, 0</td>
<td>CONDK</td>
</tr>
<tr>
<td>SHUFF</td>
<td>Shuffles the state space data and name list data for a system.</td>
<td>51</td>
<td>52</td>
<td>5, 0</td>
<td>SHUFF1, SHUFF2</td>
</tr>
<tr>
<td>SHUFF1</td>
<td>Shuffles the specified rows and columns of a matrix.</td>
<td>53</td>
<td>54</td>
<td>5, 0</td>
<td>SHUFF</td>
</tr>
<tr>
<td>SHUFF2</td>
<td>Shuffles the name list data arrays.</td>
<td>55</td>
<td>56</td>
<td>5, 0</td>
<td>SHUFF</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Program Fig.</td>
<td>Flow Chart Fig.</td>
<td>Calls</td>
<td>Called by</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>BSRED</td>
<td>Reads fractional-precision, fields 1 and 2, and shuffling data.</td>
<td>61</td>
<td>62</td>
<td>5, 0</td>
<td>DEBUG</td>
</tr>
<tr>
<td>SDRED</td>
<td>Reads scaling data.</td>
<td>63</td>
<td>64</td>
<td>5, 0</td>
<td>DEBUG</td>
</tr>
<tr>
<td>SHIFT</td>
<td>Shifts the contents of old name list arrays into new name list arrays.</td>
<td>80</td>
<td></td>
<td>5, 0</td>
<td>DEBUG</td>
</tr>
</tbody>
</table>
and scratch array dimensions are set in this program. The program calls the organizing subroutine KORG1. The program listing is given in Figure 5.

Program MAIN1

This is the main program for overlay (1,0). This program computes the required scratch array dimensions as explained in Section II, and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK1. The program listing is given in Figure 6. The dynamic storage map is given in Table 5.

Program MAIN2

This is the main program for overlay (2,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK2. The program listing is given in Figure 7. The dynamic storage map is given in Table 6.

Program MAIN3

This is the main program for overlay (3,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK3. The program listing is given in Figure 8. The dynamic storage map is given in Table 7.
Table 5. Dynamic Storage Map for Program MAIN1

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1(N1)</td>
<td>N1 + 1</td>
<td>V(MAXN)</td>
<td>MAXN + NXM + NYM + NRM</td>
<td>Calculated in KORG1</td>
</tr>
<tr>
<td>S1(N2)</td>
<td>N2 + N1 + MAXN</td>
<td>W(MAXM)</td>
<td>MAXM + NXM + NYM + NUM</td>
<td>Calculated in KORG1</td>
</tr>
<tr>
<td>S1(N3)</td>
<td>N3 + N2 + MAXM</td>
<td>F(MAXN, MAXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N4)</td>
<td>N4 + N3 + MAXN + MAXM</td>
<td>U(NUM)</td>
<td>NUM</td>
<td></td>
</tr>
<tr>
<td>S2(M1)</td>
<td>M1 + 1</td>
<td>A(NXM, NXM)</td>
<td>NXM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M2)</td>
<td>M2 + M1 + NXM + NXM</td>
<td>B(NXM, NUM)</td>
<td></td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M3)</td>
<td>M3 + M2 + NXM + NUM</td>
<td>C(NRM, NXM)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M4)</td>
<td>M4 + M3 + NRM + NXM</td>
<td>D(NRM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L1)</td>
<td>L1 + 1</td>
<td>NNS(NXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L2)</td>
<td>L2 + L1 + NXM</td>
<td>VNS(NXM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L3)</td>
<td>L3 + L2 + NXM +2</td>
<td>DES(NXM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L4)</td>
<td>L4 + L3 + NXM +10</td>
<td>UNITS(NXM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L5)</td>
<td>L5 + L4 + NXM +4</td>
<td>NNO(NRM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L6)</td>
<td>L6 + L5 + NRM</td>
<td>VNO(NRM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L7)</td>
<td>L7 + L6 + NRM +2</td>
<td>DESO(NRM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L8)</td>
<td>L8 + L7 + NRM +10</td>
<td>UNITO(NRM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L9)</td>
<td>L9 + L8 + NRM +4</td>
<td>NNI(NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L10)</td>
<td>L10 + L9 + NUM</td>
<td>VNI(NUM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L11)</td>
<td>L11 = L10 + NUM +2</td>
<td>DESI(NUM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L12)</td>
<td>L12 = L11 + NUM +10</td>
<td>UNITTI(NUM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calling Program Array</td>
<td>Array Start Index</td>
<td>Called Program Array</td>
<td>Maximum Recursion</td>
<td>Remarks</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>N1 = 1</td>
<td>S1(N1)</td>
<td>VMAX = VMH + VMH</td>
<td>VMAX</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N2 = N1 + VMH</td>
<td>S1(N2)</td>
<td>WMAX = WMH + WMH</td>
<td>WMAX</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N3 = N2 + VMH</td>
<td>S1(N3)</td>
<td>MXMD = MAX + MAX</td>
<td>MXMD</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N4 = N3 + VMH</td>
<td>S1(N4)</td>
<td>XNNT = XH + XNHT</td>
<td>XNNT</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N5 = N4 + VMH</td>
<td>S1(N5)</td>
<td>HMLT = HT + HMLT</td>
<td>HMLT</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N6 = N5 + VMH</td>
<td>S1(N6)</td>
<td>PNFH = PNFH + PNFH</td>
<td>PNFH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N7 = N6 + VMH</td>
<td>S1(N7)</td>
<td>NAXY = NAXY + NAXY</td>
<td>NAXY</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N8 = N7 + VMH</td>
<td>S1(N8)</td>
<td>NMAX = NMAX + NMAX</td>
<td>NMAX</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N9 = N8 + VMH</td>
<td>S1(N9)</td>
<td>L1N = L1N + L1N</td>
<td>L1N</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>N10 = N9 + VMH</td>
<td>S1(N10)</td>
<td>L2N = L2N + L2N</td>
<td>L2N</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>M1 = 1</td>
<td>S2(M1)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Calculated in KOBGI</td>
</tr>
<tr>
<td>M2 = M1 + VMH</td>
<td>S2(M2)</td>
<td>WMH = WMH + WMH</td>
<td>WMH</td>
<td>Calculated in KOBGI</td>
</tr>
<tr>
<td>M3 = M2 + VMH</td>
<td>S2(M3)</td>
<td>MXH = MAX + MXH</td>
<td>MXH</td>
<td>Calculated in KOBGI</td>
</tr>
<tr>
<td>M4 = M3 + VMH</td>
<td>S2(M4)</td>
<td>XH = XH + XH</td>
<td>XH</td>
<td>Calculated in KOBGI</td>
</tr>
<tr>
<td>L1 = 1</td>
<td>S3(L1)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L2 = L1 + VMH</td>
<td>S3(L2)</td>
<td>WMH = WMH + WMH</td>
<td>WMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L3 = L2 + VMH</td>
<td>S3(L3)</td>
<td>MXH = MAX + MXH</td>
<td>MXH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L4 = L3 + VMH</td>
<td>S3(L4)</td>
<td>XH = XH + XH</td>
<td>XH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L1 = L2 + VMH</td>
<td>S3(L5)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L2 = L3 + VMH</td>
<td>S3(L6)</td>
<td>WMH = WMH + WMH</td>
<td>WMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L3 = L4 + VMH</td>
<td>S3(L7)</td>
<td>MXH = MAX + MXH</td>
<td>MXH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L4 = L5 + VMH</td>
<td>S3(L8)</td>
<td>XH = XH + XH</td>
<td>XH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L5 = L6 + VMH</td>
<td>S3(L9)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L6 = L7 + VMH</td>
<td>S3(L10)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L7 = L8 + VMH</td>
<td>S3(L11)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L8 = L9 + VMH</td>
<td>S3(L12)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L9 = L10 + VMH</td>
<td>S3(L13)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L10 = L11 + VMH</td>
<td>S3(L14)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>L11 = L12 + VMH</td>
<td>S3(L15)</td>
<td>VMH = VMH + VMH</td>
<td>VMH</td>
<td>Defined in MAIN</td>
</tr>
</tbody>
</table>

Table 6: Dynamic Storage Map for Program MAIN2
Table 7. Dynamic Storage Map for Program MAIN3

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1(N1)</td>
<td>N1 = 1</td>
<td>V(MAXN)</td>
<td>MAXN = NXM + NYM + NRM</td>
<td>Calculated in KORGI</td>
</tr>
<tr>
<td>S1(N2)</td>
<td>N2 = N1 + MAXN</td>
<td>W(MAXM)</td>
<td>MAXM = NXM + 2 + NYM + NUM</td>
<td>Calculated in KORGI</td>
</tr>
<tr>
<td>S1(N3)</td>
<td>N3 = N2 + MAXM</td>
<td>F(MAXN, MAXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N4)</td>
<td>N4 = N3 + MAXN + MAXM</td>
<td>NDOT(NXM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N5)</td>
<td>N5 = N4 + NXM + MB</td>
<td>N(NXM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N6)</td>
<td>N6 = N5 + NXM + MB</td>
<td>H(NRM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N7)</td>
<td>N7 = N6 + NRM + MB</td>
<td>UH(NUM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N8)</td>
<td>N8 = N7 + NUM + MB</td>
<td>U(NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N9)</td>
<td>N9 = N8 + NUM</td>
<td>R(X(NRMMB))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N10)</td>
<td>N10 = N9 + NRMMB</td>
<td>N(NXNMB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N11)</td>
<td>N11 = N10 + MB</td>
<td>NNR(MB)</td>
<td></td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(N12)</td>
<td>N12 = N11 + MB</td>
<td>NNU(MH)</td>
<td></td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M1)</td>
<td>M1 = 1</td>
<td>A NXN, NXM)</td>
<td></td>
<td>Calculated in MAIN3</td>
</tr>
<tr>
<td>S2(M2)</td>
<td>M2 = M1 + NXM + NXM</td>
<td>B(NXM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M3)</td>
<td>M3 = M2 + NXM + NUM</td>
<td>C(NRM, NXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M4)</td>
<td>M4 = M3 + NRM + NXN</td>
<td>D(NRM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L1)</td>
<td>L1 = 1</td>
<td>NNS(NXNM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L2)</td>
<td>L2 = L1 + NXM</td>
<td>VNS(NXNM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L3)</td>
<td>L3 = L2 + NXM + 2</td>
<td>DES(NXM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L4)</td>
<td>L4 = L3 + NXM + 10</td>
<td>UNITS(NXM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L5)</td>
<td>L5 = L4 + NXM + 4</td>
<td>N(NRM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L6)</td>
<td>L6 = L5 + NRM</td>
<td>V(NRMMB, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L7)</td>
<td>L7 = L6 + NRM + 2</td>
<td>DES(NRMMB, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L8)</td>
<td>L8 = L7 + NRM + 10</td>
<td>UNIT(NRMMB, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L9)</td>
<td>L9 = L8 + NRM + 4</td>
<td>N(NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L10)</td>
<td>L10 = L9 + NUM</td>
<td>V(NUM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L11)</td>
<td>L11 = L10 + NUM + 2</td>
<td>DES(NUM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L12)</td>
<td>L12 = L11 + NUM + 10</td>
<td>UNIT(NUM, 4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Program MAIN4

This is the main program for overlay (4,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK4. The program listing is given in Figure 9. The dynamic storage map is given in Table 8.

Program MAIN5

This is the main program for overlay (5,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the conditioning subroutine CONDK. The program listing is given in Figure 10. The dynamic storage map is given in Table 9.

DESCRIPTION OF BASIC SUBROUTINES

Subroutine KORG1

This subroutine organizes the execution of KONPACT-1 program. The input data cards for KONPACT-1 program are read and printed by this subroutine. The print specification cards are read in this subroutine and the print control parameter IPRINT is set for the printer output options of KONPACT-1 program. The flow chart is given in Figure 11 and the program listing is given in Figure 12.
<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1(N1)</td>
<td>N1 = 1</td>
<td>V(M+NX)</td>
<td>MAXN = NX + NY + NR</td>
<td>Calculated in KORG1</td>
</tr>
<tr>
<td>S1(N2)</td>
<td>N2 = N1 + MAXN</td>
<td>W(MAXM)</td>
<td>MAXM = NX + 2 + NY + NUM</td>
<td>Calculated in KORG1</td>
</tr>
<tr>
<td>S1(N3)</td>
<td>N3 = N2 + MAXM</td>
<td>F(MAXN, MAXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(N4)</td>
<td>N4 = N3 + MAXN MAXM</td>
<td>U(NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M1)</td>
<td>M1 = 1</td>
<td>A(NXM, NUM)</td>
<td>NXM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M2)</td>
<td>M2 = M1 + NXM-NXM</td>
<td>B(NXM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M3)</td>
<td>M3 = M2 + NXM=NUM</td>
<td>C(NRM, NUM)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M4)</td>
<td>M4 = M3 + NRM-NXM</td>
<td>D(NRM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L1)</td>
<td>L1 = 1</td>
<td>NNS(NXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L2)</td>
<td>L2 = L1 + NXM</td>
<td>VNS(NXM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L3)</td>
<td>L3 = L2 + NXM-2</td>
<td>DES(NXM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L4)</td>
<td>L4 = L3 + NXM+10</td>
<td>UNITS(NXM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L5)</td>
<td>L5 = L4 + NXM+4</td>
<td>NNO(NRM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L6)</td>
<td>L6 = L5 + NRM</td>
<td>VNO(NRM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L7)</td>
<td>L7, L6 + NRM-2</td>
<td>DES(NRM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L8)</td>
<td>L8 = L7 + NRM+10</td>
<td>UNIT(NRM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L9)</td>
<td>L9 = L8 + NRM+4</td>
<td>NNN(NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L10)</td>
<td>L10 = L9 + NUM</td>
<td>VNN(NUM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L11)</td>
<td>L11 = L10 + NUM+2</td>
<td>DES(NUM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(L12)</td>
<td>L12 = L11 + NUM+10</td>
<td>UNITI(NUM, 4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 9. Dynamic Storage Map for Program MAIN5

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1(L1)</td>
<td>L1 = 1</td>
<td>DUMMY1(NDM11, NDM12)</td>
<td>NDM11 = MAX(17, NXM, NRM)</td>
<td>Calculated in MAIN5</td>
</tr>
<tr>
<td>S1(L2)</td>
<td>L2 = L1 + NDM11 + NDM12</td>
<td>DUMMY2(NDM21, NDM22)</td>
<td>NDM12 = MAX(NXM * NUM, NRM)</td>
<td>Calculated in MAIN5</td>
</tr>
<tr>
<td>S1(L3)</td>
<td>L3 = L2 + NDM21 + NDM22</td>
<td>DUMMY3(NUM)</td>
<td>NDM21 = MAX(NRM, NXM)</td>
<td>Calculated in MAIN5</td>
</tr>
<tr>
<td>S1(L4)</td>
<td>L4 = L3 + NUM</td>
<td>ES(NXM, NUM)</td>
<td>NDM22 = MAX(NXM, NRM, NUM)</td>
<td>Calculated in MAIN5</td>
</tr>
<tr>
<td>S1(L5)</td>
<td>L5 = L4 + NXM * NUM</td>
<td>ER(NRM, NUM)</td>
<td>NXM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L6)</td>
<td>L6 = L5 + NRM * NUM</td>
<td>NSHUF8(NXM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L7)</td>
<td>L7 = L6 + NXM</td>
<td>NSHUF0(NRM)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L8)</td>
<td>L8 = L7 + NRM</td>
<td>NSHUF1(NUM)</td>
<td>NXM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L9)</td>
<td>L9 = L8 + NUM</td>
<td>CS(NRM, NXM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L10)</td>
<td>L10 = L9 + NRM * NXM</td>
<td>DS(NRM, NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L11)</td>
<td>L11 = L10 + NRM * NUM</td>
<td>CW(NRM, NXM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L12)</td>
<td>L12 = L11 + NRM * NXM</td>
<td>DW(NRM, NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L13)</td>
<td>L13 = L12 + NRM * NUM</td>
<td>IRS(NRM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L14)</td>
<td>L14 = L13 + NRM</td>
<td>Q(NRM, NRM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M1)</td>
<td>M1 = 1</td>
<td>AN(XM, NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M2)</td>
<td>M2 = M1 + NXM * NXM</td>
<td>B(NXM, NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M3)</td>
<td>M3 = M2 + NXM * NUM</td>
<td>C(NRM, NXM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M4)</td>
<td>M4 = M3 + NRM * NXM</td>
<td>DN(RM, NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M5)</td>
<td>M5 = M4 + NRM * NUM</td>
<td>CM(NRM, NXM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S2(M6)</td>
<td>M6 = M5 + NRM * NXM</td>
<td>DM(NRM, NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S3(N1)</td>
<td>N1 = 1</td>
<td>NNS(NXM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S3(N2)</td>
<td>N2 = N1 + NXM</td>
<td>VNS(NXM, 2)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S3(N3)</td>
<td>N3 = N2 + NXM * 2</td>
<td>DESS(NXM, 10)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S3(N4)</td>
<td>N4 = N3 + NXM * 10</td>
<td>UNITS(NXM, 4)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
</tbody>
</table>
Table 9. Dynamic Storage Map for Program MAIN5  (Concluded)

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3(N5)</td>
<td>N5 = N4 + NXM+4</td>
<td>NNO(NRM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N6)</td>
<td>N6 = N5 + NRM</td>
<td>VNO(NRM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N7)</td>
<td>N7 = N6 + NRM+2</td>
<td>DESO(NRM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N8)</td>
<td>N8 = N7 + NRM+10</td>
<td>UNITO(NRM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N9)</td>
<td>N9 = N8 + NRM+4</td>
<td>NNI(NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N10)</td>
<td>N10 = N9 + NUM</td>
<td>VNI(NUM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N11)</td>
<td>N11 = N10 + NUM+2</td>
<td>DES(NUM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N12)</td>
<td>N12 = N11 + NUM+10</td>
<td>UNIT(NUM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N13)</td>
<td>N13 = N12 + NUM+4</td>
<td>NNS(NXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N14)</td>
<td>N14 = N13 + NXM</td>
<td>VNS(NXM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N15)</td>
<td>N15 = N14 + NXM+2</td>
<td>DESNS(NXM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N16)</td>
<td>N16 = N15 + NXM+10</td>
<td>UNITNS(NXM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N17)</td>
<td>N17 = N16 + NXM+4</td>
<td>NNO(NRM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N18)</td>
<td>N18 = N17 + NRM</td>
<td>VNO(NRM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N19)</td>
<td>N19 = N18 + NRM+2</td>
<td>DES(NRM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N20)</td>
<td>N20 = N19 + NRM+10</td>
<td>UNIT(NRM, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N21)</td>
<td>N21 = N20 + NRM+4</td>
<td>NNI(NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N22)</td>
<td>N22 = N21 + NUM</td>
<td>VNI(NUM, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N23)</td>
<td>N23 = N22 + NUM+2</td>
<td>DES(NUM, 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(N24)</td>
<td>N24 = N23 + NUM+10</td>
<td>UNIT(NUM, 4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Subroutine NAMEL

This subroutine obtains the name list data for the system variables. The subroutine either reads the name list data from input data cards or internally obtains a default name list data. In the case of combining various subsystems into an overall system, the subroutine uses the interconnection data to obtain the appropriate name list data. This subroutine also writes the name list data of each system on NDATA file for use by other programs. The flow chart is given in Figure 13 and the program listing is given in Figure 14.

Subroutine STAMK1

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK1. The flow chart is given in Figure 15 and the program listing is given in Figure 16. The dynamic storage map is given in Table 10.

Subroutine SIMK1

This subroutine reads simulator deck data and load equation data obtained by the Linear System Analysis (LSA) program and implements them into simulation equations. The flow chart is given in Figure 17 and the program listing is given in Figure 18.

Subroutine STAMK2

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMKT. The flow chart is given in
Table 10. Dynamic Storage Map for Program STAMK1

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(N1)</td>
<td>N1 = 1</td>
<td>XDOT(NX)</td>
<td>NX</td>
<td>Calculated in SIMK1</td>
</tr>
<tr>
<td>W(N2)</td>
<td>N2 = N1 + NX</td>
<td>Y(NY)</td>
<td>NY</td>
<td>Calculated in SIMK1</td>
</tr>
<tr>
<td>W(N3)</td>
<td>N3 = N2 + NY</td>
<td>X(NX)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W(N4)</td>
<td>N4 = N3 + NX</td>
<td>U(NU)</td>
<td>NU</td>
<td>Calculated in SIMK1</td>
</tr>
<tr>
<td>V(N1)</td>
<td>N1 = 1</td>
<td>XDOTL(NX)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V(N2)</td>
<td>N2 = N1 + NX</td>
<td>YL(NY)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V(N3)</td>
<td>N3 = N1 + NY</td>
<td>RL(NR)</td>
<td>NR</td>
<td>Calculated in SIMK1</td>
</tr>
<tr>
<td>S1(L1)</td>
<td>L1 = 1</td>
<td>DESSS(NXM, 10, MB)</td>
<td>NXM, MB</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L2)</td>
<td>L2 = L1 + NXM<em>10</em>MB</td>
<td>UNITSS(NXM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L3)</td>
<td>L3 = L2 + NXM<em>4</em>MB</td>
<td>DESOO(NRM, 10, MB)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L4)</td>
<td>L4 = L3 + NRM<em>10</em>MB</td>
<td>UNITOO(NRM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L5)</td>
<td>L5 = L4 + NRM<em>4</em>MB</td>
<td>DESII(NUM, 10, MB)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L6)</td>
<td>L6 = L5 + NUM<em>10</em>MB</td>
<td>UNITII(NUM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L7)</td>
<td>L7 = L6 + NUM<em>4</em>MB</td>
<td>NXX(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L8)</td>
<td>L8 = L7 + MB</td>
<td>NRR(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L9)</td>
<td>L9 = L8 + MB</td>
<td>NUU(MB)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 19 and the program listing is given in Figure 20. The dynamic storage map is given in Table 11.

Subroutine SIMKT

This subroutine reads transfer function data and connection data and implements them into simulation equations. The flow chart is given in Figure 21 and the program listing is given in Figure 22.

Subroutine TRANSK

This subroutine computes the state space model for rational transfer functions using the input Frobenius form of realization. The flow chart is given in Figure 23 and the program listing is given in Figure 24.

Subroutine DFN

This subroutine selects the specified Pade approximation to transport (time) delay from a table of Pade approximations. The flow chart is given in Figure 25 and the program listing is given in Figure 26.

Subroutine PHERR

This subroutine computes the phase error of the Pade approximation to transport (time) delay. The flow chart is given in Figure 27 and the program listing is given in Figure 28.
<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>V(N1)</td>
<td>N1 = 1</td>
<td>XDOL(NX)</td>
<td>NX</td>
<td>Calculated in SIMKT</td>
</tr>
<tr>
<td>V(N2)</td>
<td>N2 = N1 + NX</td>
<td>YL(NY)</td>
<td>NY</td>
<td>Calculated in SIMKT</td>
</tr>
<tr>
<td>V(N3)</td>
<td>N3 = N2 + NY</td>
<td>RL(NR)</td>
<td>NR</td>
<td>Calculated in SIMKT</td>
</tr>
<tr>
<td>S1(L1)</td>
<td>L1 = 1</td>
<td>DESSS(NXM, 10, MB)</td>
<td>NXM, MB</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L2)</td>
<td>L2 = L1 + NXM^<em>4</em>MB</td>
<td>UNITSO(NRM, 4, MB)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L3)</td>
<td>L3 = L2 + NXM^<em>4</em>MB</td>
<td>DESOO(NRM, 10, MB)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L4)</td>
<td>L4 = L3 + NRM^<em>4</em>MB</td>
<td>UNITTOO(NRM, 4, MB)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L5)</td>
<td>L5 = L4 + NRM^<em>4</em>MB</td>
<td>DESII(NUM, 10, MB)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L6)</td>
<td>L6 = L5 + NUM^<em>4</em>MB</td>
<td>UNITII(NUM, 4, MB)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L7)</td>
<td>L7 = L6 + NUM^<em>4</em>MB</td>
<td>NXX(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L8)</td>
<td>L8 = L7 + MB</td>
<td>NRR(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L7)</td>
<td>L9 = L8 + MB</td>
<td>NUU(MB)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Subroutine STAMK3

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK. The flow chart is given in Figure 29 and the program listing is given in Figure 30. The dynamic storage map is given in Table 12.

Subroutine SIMK

This subroutine reads interconnection data and state space data for sub-systems and implements the simulation equations for the overall system. SIMK also writes the interconnection data on the scratch file for use by subroutine NAMEL. The flow chart is given in Figure 31 and the program listing is given in Figure 32.

Subroutine QUADK

This subroutine reads directly the state space data for the system. The flow chart is given in Figure 33 and the program listing is given in Figure 34.

Subroutine STAMK4

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK2. The flow chart is given in Figure 35 and the program listing is given in Figure 36. The dynamic storage map is given in Table 13.
Table 12. Dynamic Storage Map for Program STAMK3

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>V(N1)</td>
<td>N1 = 1</td>
<td>XDOTL(NX)</td>
<td>NX</td>
<td>Calculated in SIMK</td>
</tr>
<tr>
<td>V(N2)</td>
<td>N2 = N1 + NX</td>
<td>YL(NY)</td>
<td>NY</td>
<td>Calculated in SIMK</td>
</tr>
<tr>
<td>V(N3)</td>
<td>N3 = N2 + NY</td>
<td>RL(NR)</td>
<td>NR</td>
<td>Calculated in SIMK</td>
</tr>
<tr>
<td>S1(L1)</td>
<td>L1 = 1</td>
<td>DESSS(NXM, 10, MB)</td>
<td>NXM, MB</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L2)</td>
<td>L2 = L1 + NXM<em>10</em>MB</td>
<td>UNITSS(NXM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L3)</td>
<td>L3 = L2 + NXM<em>4</em>MB</td>
<td>DESOO(NRM, 10, MB)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L4)</td>
<td>L4 = L3 + NRM<em>10</em>MB</td>
<td>UNITI(NRM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L5)</td>
<td>L5 = L4 + NRM<em>4</em>MB</td>
<td>DESII(NUM, 10, MB)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L6)</td>
<td>L6 = L5 + NUM<em>10</em>MB</td>
<td>UNITII(NUM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L7)</td>
<td>L7 = L6 + NUM<em>4</em>MB</td>
<td>NXX(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L8)</td>
<td>L8 = L7 + MB</td>
<td>NRR(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L9)</td>
<td>L9 = L8 + MB</td>
<td>NUU(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M1)</td>
<td>M1 = 1</td>
<td>AT(NXM, NXM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M2)</td>
<td>M2 = M1 + NXM<em>NXM</em>MB</td>
<td>BT(NXM, NUM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M3)</td>
<td>M3 = M2 + NXM<em>NUM</em>MB</td>
<td>CT(NRM, NXM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M4)</td>
<td>M4 = M3 + NRM<em>NXM</em>MB</td>
<td>DT(NRM, NUM, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M5)</td>
<td>M5 = M4 + NRM<em>NUM</em>MB</td>
<td>P(MN, MN)</td>
<td>MN=MM*MB</td>
<td>Calculated in KORG1</td>
</tr>
</tbody>
</table>
### Table 12. Dynamic Storage Map for Program STAMK3 (Concluded)

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2(M6)</td>
<td>M6 = M5 + MN*MN</td>
<td>Q(MN, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M7)</td>
<td>M7 = M6 + MN*NUM</td>
<td>R(NRM, MN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(M8)</td>
<td>M8 = M7 + NRM*MN</td>
<td>S(NRM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(K1)</td>
<td>K1 = 1</td>
<td>PP(MP, MM, MM)</td>
<td>MN = MAX</td>
<td>Calculated in KORG1</td>
</tr>
<tr>
<td>S3(K2)</td>
<td>K2 = K1 + MP<em>MM</em>MM</td>
<td>QQ(MQ, MM, NUM)</td>
<td>MQ = MB</td>
<td>Calculated in KORG1</td>
</tr>
<tr>
<td>S3(K3)</td>
<td>K3 = K2 + MQ<em>MM</em>NUM</td>
<td>RR(MR, NRM, MM)</td>
<td>MR = MB</td>
<td>Calculated in KORG1</td>
</tr>
<tr>
<td>S3(K4)</td>
<td>K4 = K3 + MR<em>NRM</em>MM</td>
<td>NSP(MP)</td>
<td>MP = MB*2</td>
<td></td>
</tr>
<tr>
<td>S3(K5)</td>
<td>K5 = K4 + MP</td>
<td>NSQ(MQ)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3(K6)</td>
<td>K6 = K5 + MQ</td>
<td>NSR(MR)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 13. Dynamic Storage Map for Program STAMK4

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(N1)</td>
<td>N1 = 1</td>
<td>XDOT(NX)</td>
<td>NX</td>
<td>Calculated in SIMK2</td>
</tr>
<tr>
<td>W(N2)</td>
<td>N2 = N1 + NX</td>
<td>Y(NY)</td>
<td>NY</td>
<td>Calculated in SIMK2</td>
</tr>
<tr>
<td>W(N3)</td>
<td>N3 = N2 + NY</td>
<td>X(NX)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W(N4)</td>
<td>N4 = N3 + NX</td>
<td>U(NU)</td>
<td>U(U)</td>
<td>Calculated in SIMK2</td>
</tr>
<tr>
<td>V(N1)</td>
<td>N1 = 1</td>
<td>XDOTL(NX)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V(N2)</td>
<td>N2 = N1 + NX</td>
<td>YL(NY)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V(N3)</td>
<td>N3 = N2 + NY</td>
<td>RL(NR)</td>
<td>NR</td>
<td>Calculated in SIMK2</td>
</tr>
<tr>
<td>S1(L1)</td>
<td>L1 = 1</td>
<td>DESSS(NXM, 10, MB)</td>
<td>NXM, MB</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L2)</td>
<td>L2 = L1 + NXM<em>10</em>MB</td>
<td>UNITSS(NXM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L3)</td>
<td>L3 = L2 + NXM<em>4</em>MB</td>
<td>DESOO(NRM, 10, MB)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L4)</td>
<td>L4 = L3 + NRM<em>10</em>MB</td>
<td>UNITOO(NRM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L5)</td>
<td>L5 = L4 + NRM<em>4</em>MB</td>
<td>DESII(NUM, 10, MB)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(L6)</td>
<td>L6 = L5 + NUM<em>10</em>MB</td>
<td>UNITII(NUM, 4, MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L7)</td>
<td>L7 = L6 + NUM<em>4</em>MB</td>
<td>NXX(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L8)</td>
<td>L8 = L7 + MB</td>
<td>NRR(MB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1(L9)</td>
<td>L9 = L8 + MB</td>
<td>NUU(MB)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Subroutine SIMK2

This is a users written subroutine. Here it is written for the ALDCS controller. It reads ALDCS controller gains and switch modes (positions) and implements the controller into simulation equations. The flow chart is given in Figure 37 and the program listing is given in Figure 38.

Subroutine CONDK

This subroutine organizes the modification (conditioning) of the state space data and name list data according to specification. The flow chart is given in Figure 39 and the program listing is given in Figure 40.

Subroutine MNAME

This subroutine modifies the name list data of the system according to the conditioning data. The flow chart is given in Figure 41 and the program listing is given in Figure 42.

Subroutine IMRATE

This subroutine obtains the implicit model error rates and truncates the implicit model. The flow chart is given in Figure 43 and the program listing is given in Figure 44.
Subroutine DIFFK

This subroutine obtains the rate of change of either a specified response or state of the system by differentiation. If the differentiation requires external rate inputs in the model, a message is printed by the subroutine. The flow chart is given in Figure 45 and the program listing is given in Figure 46.

Subroutine REDUCE

This subroutine residualizes or truncates the state space data of the system. In addition it computes the error of residualization. The flow chart is given in Figure 47 and the program listing is given in Figure 48.

Subroutine SCAL

This subroutine computes the scaled state space data. The flow chart is given in Figure 49 and the program listing is given in Figure 50.

Subroutine SHUFF

This subroutine shuffles the state space data and the name list data by calling subroutines SHUF1 and SHUF2. The flow chart is given in Figure 51 and the program listing is given in Figure 52.

Subroutine SHUF1

This subroutine shuffles the rows and columns of a matrix. The flow chart is given in Figure 53 and the program listing is given in Figure 54.
Subroutine SHUF2

This subroutine shuffles the name list data arrays. The flow chart is given in Figure 55 and the program listing is given in Figure 56.

DESCRIPTION OF AUXILIARY SUBROUTINES

Subroutine QDIO

This subroutine reads the state space data from file QDATA and prints it. It also writes the state space data on file QDATA. The flow chart is given in Figure 57 and the program listing is given in Figure 58.

Subroutine IDRO

This subroutine reorganizes the input data. The reorganized input data is written on file BINPUT. The flow chart is given in Figure 59 and the program listing is given in Figure 60.

Subroutine RSDRD

This subroutine reads residualization, truncation, and shuffling data for the variables of the system. The flow chart is given in Figure 61 and the program listing is given in Figure 62.
Subroutine SDRD

This subroutine reads the scaling factor and the new units for the system variables. The flow chart is given in Figure 63 and the program listing is given in Figure 64.

Subroutine FILE

This subroutine positions the data file for reading or writing data. There are three modes of calling this subroutine. INSERT mode inserts the label name and positions the data file for writing. LOCATE mode locates the label name and positions the data file for reading. NULL mode removes the label name from the data file. The flow chart is given in Figure 65 and the program listing is given in Figure 66.

Subroutine TPR

This subroutine prints transfer function data. The flow chart is given in Figure 67 and the program listing is given in Figure 68.

Subroutine HPR

This subroutine prints the headings for the system label name. The program listing is given in Figure 69.

Subroutine IDPR

This subroutine prints the input data. The program listing is given in Figure 70.
Subroutine MPRS

This subroutine prints a matrix, identifying the rows and columns. The program listing is given in Figure 71.

Subroutine MPRS1

This subroutine prints the simulator interface matrix data from the Linear System Analysis (LSA) program. The program listing is given in Figure 72.

Subroutine ZERO

This subroutine initializes (or zeros) the elements of a matrix. The program listing is given in Figure 73.

Subroutine INPT

This subroutine reads the nonzero elements of a matrix. The program listing is given in Figure 74.

Subroutine INPT1

This subroutine reads the simulator interface matrix data from Linear System Analysis (LSA) program. The program listing is given in Figure 75.
Subroutine DEBUG

This subroutine prints a debugging message. The program listing is given in Figure 76.

Subroutine ERRM

This subroutine prints an error message indicating the program and overlay at which the error was detected. The program listing is given in Figure 77.

Subroutine DERRM

This subroutine prints a message when the maximum dimensions for scratch arrays are not sufficient. The program listing is given in Figure 78.

Subroutine DERRMS

This subroutine prints a message when the Maximum System dimensions are not sufficient. The program listing is given in Figure 79.

Subroutine SHIFT

This subroutine shifts the contents of old name list arrays into new name list arrays. The program listing is given in Figure 80.

Subroutine TDINVR

This subroutine inverts a non-singular matrix or solves a set of linear equations. The program listing is given in Figure 81.
SECTION IV

DESIGN PROGRAM (KONPACT-2)

The data produced by KONPACT-1 are utilized by KONPACT-2. KONPACT-2 contains two Air Force-owned synthesis programs, DIAK and FFOC. The DIAK (Doubly Iterative Algorithm developed by Konar) program computes optimal controller gains for full state feedback. FFOC (Fixed Form Optimal Control) simplifies these gains to specified measurements. KONPACT-2 interfaces with FLEXSTAB through the LSA program to evaluate performances of the closed loop system.

In this section, a description of KONPACT-2 program is presented in terms of overlay structure, flow charts, and program listings. The DIAK and FFOC programs are fully documented in Reference 2 and only the program listings are given here for completeness. Modularization and variable dimensioning of DIAK and FFOC programs are beyond the scope of this contract.

OVERLAY STRUCTURE

The KONPACT-2 program consists of a main overlay and three primary overlays. The overlay structure and the subroutines in each overlay is given in Figure 82. The subroutine summary consisting of name, description, reference, overlay position, and interrelationship is given in Table 14.
### Table 14. KONPACT-2 Subroutine Summary

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
<th>Flow Chart Fig. #</th>
<th>Program Listing Fig. #</th>
<th>Overlay</th>
<th>Inter-relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>Sets up system dimensions and scratch array dimensions.</td>
<td>83</td>
<td>0, 0</td>
<td>KORG2</td>
<td></td>
</tr>
</tbody>
</table>
| KORG2      | Organizes input data and calls the primary overlays. | 87 | 88 | IDRO  
|            |             |                  |                        |         | ERMM               |
| IDRO       | Reorganizes the input data. | | 0, 0 | KORG2 |
| IDPR       | Prints input data. | | 0, 0 | KORG2 |
| ERRM       | Prints error message. | | 0, 0 | KORG2  
|            |             |                  |                        |         | DATAK              |
| TDINVR     | Inverts a nonsingular matrix or solves a set of linear equations. | | 0, 0 | DIAK  
|            |             |                  |                        |         | CALL1  
|            |             |                  |                        |         | FFOC  
|            |             |                  |                        |         | GCAL  
|            |             |                  |                        |         | CAL |
| MP         | Prints matrix data. | 110 | 0, 0 | DIAK  
|            |             |                  |                        |         | STRIC  
|            |             |                  |                        |         | RESP |
|            |             |                  |                        |         | FFOC |
| OUTP       | Writes nonzero elements of a matrix on a data file. | 111 | 0, 0 | DIAK  
|            |             |                  |                        |         | FFOC |
| INPT       | Reads nonzero elements of a matrix. | | 0, 0 | DIAK  
|            |             |                  |                        |         | FFOC |
| ZERO       | Initializes (or zeros) the elements of a matrix. | | 0, 0 | FFOC  
|            |             |                  |                        |         | DIAK  
<p>|            |             |                  |                        |         | DFILOC |
|            |             |                  |                        |         | DLSA |</p>
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
<th>Flow Chart Fig.</th>
<th>Program Listing Fig.</th>
<th>Overlay</th>
<th>Inter-relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLES</td>
<td>Computes the eigenvalues of a matrix.</td>
<td>112</td>
<td>0,0</td>
<td>HESSEN QRCALL DIAK</td>
<td></td>
</tr>
<tr>
<td>HESSEN</td>
<td>Reduces a matrix to upper Hessenberg form by Gaussian elimination.</td>
<td>113</td>
<td>0,0</td>
<td>POLES</td>
<td></td>
</tr>
<tr>
<td>QRCALL</td>
<td>Computes eigenvalues of an upper Hessenberg matrix.</td>
<td>114</td>
<td>0,0</td>
<td>QR POLES</td>
<td></td>
</tr>
<tr>
<td>QR</td>
<td>Performs a double QR iteration on a real matrix.</td>
<td>115</td>
<td>0,0</td>
<td>QRCALL</td>
<td></td>
</tr>
<tr>
<td>DIAK</td>
<td>Computes optimal state feedback gains for a linear time-invariant system with a quadratic cost function.</td>
<td>84</td>
<td>1,0</td>
<td>INPT SHUFL MP STRIC TDINVR CALI OUTP TIMER POLES</td>
<td></td>
</tr>
<tr>
<td>TIMER</td>
<td>Computes time response.</td>
<td>89</td>
<td>1,0</td>
<td>SGUST DIAK</td>
<td></td>
</tr>
<tr>
<td>SGUST</td>
<td>Computes step gust input.</td>
<td>90</td>
<td>1,0</td>
<td>TIMER</td>
<td></td>
</tr>
<tr>
<td>CALI</td>
<td>Solves square Lyapunov equation.</td>
<td>91</td>
<td>1,0</td>
<td>TDINVR COVR CONST DIAK</td>
<td></td>
</tr>
<tr>
<td>STRIC</td>
<td>Computes stable set of starting gains for DIAK.</td>
<td>92</td>
<td>1,0</td>
<td>MP DIAK</td>
<td></td>
</tr>
<tr>
<td>SHUFL</td>
<td>Reorders columns and rows of a matrix.</td>
<td>93</td>
<td>1,0</td>
<td>DIAK</td>
<td></td>
</tr>
<tr>
<td>GRAN</td>
<td>Generates random numbers.</td>
<td>118</td>
<td>1,0</td>
<td>TIMER</td>
<td></td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Flow Chart Fig.</td>
<td>Program Listing Fig.</td>
<td>Overlay</td>
<td>Inter-relationship</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
<td>---------------------</td>
<td>---------</td>
<td>-------------------</td>
</tr>
<tr>
<td>FFOC</td>
<td>Computes simplified controller gains for a linear time-invariant system with a quadratic cost function.</td>
<td>85</td>
<td>2,0</td>
<td></td>
<td>POLES OUTP COVAR TRANS COSTAT RESP UNSCR MP ZERO INPT SHUF TRNVR</td>
</tr>
<tr>
<td>SHUF</td>
<td>Reorders rows and columns of matrices.</td>
<td>94</td>
<td>2,0</td>
<td></td>
<td>FFOC</td>
</tr>
<tr>
<td>RESP</td>
<td>Computes covariances for disturbance inputs.</td>
<td>95</td>
<td>2,0</td>
<td></td>
<td>FFOC</td>
</tr>
<tr>
<td>COVAR</td>
<td>Computes covariance matrix.</td>
<td>96</td>
<td>2,0</td>
<td>CAL GCAL</td>
<td>RESP</td>
</tr>
<tr>
<td>COSTAT</td>
<td>Computes costate matrix.</td>
<td>97</td>
<td>2,0</td>
<td>CAL GCAL</td>
<td>FFOC</td>
</tr>
<tr>
<td>TRANS</td>
<td>Computes gradient transformation matrix.</td>
<td>98</td>
<td>2,0</td>
<td></td>
<td>FFOC</td>
</tr>
<tr>
<td>UNSCR</td>
<td>Transforms the gradient transformation matrix.</td>
<td>99</td>
<td>2,0</td>
<td></td>
<td>FFOC</td>
</tr>
<tr>
<td>GCAL</td>
<td>Solves rectangular Lyapunov equation.</td>
<td>100</td>
<td>2,0</td>
<td>TBINVR</td>
<td>COVAR COSTAT</td>
</tr>
<tr>
<td>CAL</td>
<td>Solves square Lyapunov equation.</td>
<td>101</td>
<td>2,0</td>
<td>TBINVR</td>
<td>COVAR COSTAT</td>
</tr>
<tr>
<td>DATAK</td>
<td>Sets up array start indices and checks if scratch array size is sufficient.</td>
<td>102</td>
<td>3,0</td>
<td></td>
<td>BMASK DIFOC DEXL FMP DURM EURM</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Flow Chart Fig.#</td>
<td>Program Listing Fig.#</td>
<td>Overlay</td>
<td>Inter-relationship Calls</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
<td>---------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DDIAK</td>
<td>Prepare data file for DIAK program.</td>
<td>102</td>
<td>103</td>
<td>3,0</td>
<td>ZERO FILE MPRS WTP</td>
</tr>
<tr>
<td>DFFOC</td>
<td>Prepares data file for FFOC program.</td>
<td>104</td>
<td>105</td>
<td>3,0</td>
<td>ZERO FILE MPRS WTP</td>
</tr>
<tr>
<td>DLSA</td>
<td>Prepares data for FINK program.</td>
<td>106</td>
<td>107</td>
<td>3,0</td>
<td>ZERO FILE MPRS INPTM</td>
</tr>
<tr>
<td>FINK</td>
<td>Converts state space data into frequency domain data for LSA program.</td>
<td>108</td>
<td>109</td>
<td>3,0</td>
<td>MPRS</td>
</tr>
<tr>
<td>MPRS</td>
<td>Prints matrix data.</td>
<td></td>
<td></td>
<td>3,0</td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>Locates and inserts system labels on disc files and writes end of data mark.</td>
<td></td>
<td></td>
<td>3,0</td>
<td></td>
</tr>
<tr>
<td>INPTM</td>
<td>Reads nonzero elements of a matrix.</td>
<td></td>
<td></td>
<td>3,0</td>
<td></td>
</tr>
<tr>
<td>WTP</td>
<td>Writes nonzero elements of a matrix on a data file.</td>
<td></td>
<td></td>
<td>3,0</td>
<td></td>
</tr>
<tr>
<td>DERRM</td>
<td>Prints an error message when the dimensions for scratch arrays are not sufficient.</td>
<td></td>
<td></td>
<td>3,0</td>
<td></td>
</tr>
</tbody>
</table>
DESCRIPTION OF MAIN PROGRAMS

Program MAIN

This is the main program for overlay (0, 0). This program assigns the various file numbers used in KONPACT-2. Maximum system dimensions and scratch array dimensions are set in this program. (Note that scratch arrays should be defined in DATAK program.) The program calls the organizing subroutine KORG2. The program listing is given in Figure 83.

Program DIAK

This is the main program for overlay (1, 0). This program computes optimal state feedback gains for a linear time-invariant system with quadratic cost function. The program listing is given in Figure 84.

Program FFOC

This is the main program for overlay (2, 0). This program computes simplified controller gains for a linear time-invariant system with a quadratic cost function. The program listing is given in Figure 85.

Program DATAK

This is the main program for overlay (3, 0). The scratch arrays are defined here. The program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the data preparation subroutines DDIAK, DFFOC, DLSA and FINK. The program listing is given in Figure 86. The dynamic storage map is given in Table 15.
Table 15. Dynamic Storage Map for Program DATAK

<table>
<thead>
<tr>
<th>Calling Program Array</th>
<th>Array Start Index</th>
<th>Called Program Array</th>
<th>Maximum Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1(M1)</td>
<td>M1 = 1</td>
<td>A(NXM, NXM)</td>
<td>NXM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(M2)</td>
<td>M2 = M1 + NXM*NXM</td>
<td>B(NXM, NUM)</td>
<td>NUM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(M3)</td>
<td>M3 = M2 + NXM*NUM</td>
<td>C(NRM, NXM)</td>
<td>NRM</td>
<td>Defined in MAIN</td>
</tr>
<tr>
<td>S1(M4)</td>
<td>M4 = M3 + NRM*NXM</td>
<td>D(NRM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(N1)</td>
<td>N1 = 1</td>
<td>B1(NXM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(N2)</td>
<td>N2 = N1 + NXM*NUM</td>
<td>B2(NXM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(N3)</td>
<td>N3 = N2 + NXM*NUM</td>
<td>C1(NRM, NXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(N4)</td>
<td>N4 = N3 + NRM*NUM</td>
<td>C3(NRM, NXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(N5)</td>
<td>N5 = N4 + NRM*NXM</td>
<td>D11(NRM, NUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(N6)</td>
<td>N6 = N5 + NRM*NUM</td>
<td>BK(NUM, NRM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(N7)</td>
<td>N7 = N6 + NUM*NRM</td>
<td>BKC3(NUM, NXM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(K1)</td>
<td>K1 = 1</td>
<td>CC(NXRM, NXRUM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2(K2)</td>
<td>K2 = K1 + NXRM*NXRUM</td>
<td>NAME(NXRUM)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calculations:

- NXRM = NXM + NRM
- NXRUM = NXRM + NUM

**Remarks**

- Defined in MAIN
- Calculated in DATAK
DESCRIPTION OF BASIC SUBROUTINES

Subroutine KORG2

This subroutine organizes the execution of KONPACT-2 program. The input data cards for KONPACT-2 program are read and printed by the subroutine. The print specification cards are read in this subroutine, and the print control parameter IPRINT is set for the printed output options of the KONPACT-2 program. Under the control of the input data this subroutine calls the overlay loader to load the required primary overlay into central memory for execution. The flow chart is given in Figure 87 and the program listing is given in Figure 88.

Subroutine TIMER

This subroutine computes the time response for step command inputs and step gust inputs. The program listing is given in Figure 89.

Subroutine SGUST

This subroutine computes step gust input. The program listing is given in Figure 90.

Subroutine CAL1

This subroutine solves square Lyapunov equation. The program listing is given in Figure 91.
Subroutine STRIC

This subroutine computes a stable set of starting gains for DIAK. The program listing is given in Figure 92.

Subroutine SHUFL

This subroutine reorders the columns and rows of a matrix. The program listing is given in Figure 93.

Subroutine SHUF

This subroutine records rows and columns of matrices. The program listing is given in Figure 94.

Subroutine RESP

This subroutine computes covariances for disturbance inputs. The program listing is given in Figure 95.

Subroutine COVAR

This subroutine computes the covariance matrix. The program listing is given in Figure 96.

Subroutine COSTAT

This subroutine computes the costate matrix. The program listing is given in Figure 97.
Subroutine TRANS

This subroutine computes the gradient transformation matrix. The program listing is given in Figure 98.

Subroutine UNSCR

This subroutine transforms the gradient transformation matrix. The program listing is given in Figure 99.

Subroutine GCAL

This subroutine solves the rectangular Lyapunov equation. The program listing is given in Figure 100.

Subroutine CAL

This subroutine solves the square Lyapunov equation. The program listing is given in Figure 101.

Subroutine DDIAK

This subroutine reads data from cards or from file QDATA and prepares data file SCRTCH for DIAK subprogram. The flow chart is given in Figure 102 and the program listing is given in Figure 103.
Subroutine DFFOC

This subroutine reads data from cards or from file QDATA and prepares data file SCHTCH for FFOC subprogram. The flow chart is given in Figure 104 and the program listing is given in Figure 105.

Subroutine DLSA

This subroutine reads data from files QDATA, DDATA, and FDATA and prepares open loop or closed loop state space data. The flow chart is given in Figure 106, and the program listing is given in Figure 107.

Subroutine FINK

This subroutine uses the state space data obtained by the DLSA subroutine, computes the frequency domain data, and writes it on file SDSTP for the LSA program. The flow chart is given in Figure 108, and the program listing is given in Figure 109.

DESCRIPTION OF AUXILIARY SUBROUTINES

Subroutine MP

This subroutine prints matrix data. The program listing is given in Figure 110.
Subroutine OUTP

This subroutine writes the nonzero elements of a matrix on a data file. The program listing is given in Figure 111.

Subroutine POLES

This subroutine computes the eigenvalues of a matrix. The program listing is given in Figure 112.

Subroutine HESSEN

This subroutine computes the upper Hessenberg form of a matrix by Gaussian elimination. The program listing is given in Figure 113.

Subroutine QRCALL

This subroutine computes the eigenvalues of an upper Hessenberg form matrix. The program listing is given in Figure 114.

Subroutine QR

This subroutine performs a double QR iteration on a real matrix. The program listing is given in Figure 115.

Subroutine INPTM

This subroutine reads nonzero elements of a matrix. The program listing is given in Figure 116.
Subroutine WTP

This subroutine writes the nonzero elements of a matrix on a data file. The program listing is given in Figure 117.

Function GRAN

This function subroutine generates random numbers. The program listing is given in Figure 118.

For documentation of subroutines IDRO, IDPR, ERRM, TDINVR, INPT, ZERO, MPRS, FILE, and DERRM the reader is referred to Section III.
SECTION V

CONCLUSIONS AND RECOMMENDATIONS

The scope of this program was to develop programs to interface the level 2.01.00 FLEXSTAB with DIAK/FFOC optimal control programs. The theory and algorithms for the interface are presented in Volume I. Two demonstration examples are given in Volume III to show the data mechanics of the interface. A brief documentation of the interface program KONPACT is provided in this volume.

RECOMMENDATIONS FOR FUTURE SOFTWARE DEVELOPMENT WORK

- Full documentation of KONPACT should be made
- DIAK/FFOC programs should be modularized and variable dimensioned
- Faster algorithms should be used to reduce design time
- Reduced Controller Software (FFOC) should be augmented with the minimal order observer design capability

CONCLUSIONS

A large-scale software - KONPACT - for the design and analysis of active control systems is briefly documented in this volume. The work reported in Volumes I, II and III established the total dynamic system approach for the design and analysis.
Figure 1. Interface Between LSA and KONPACT Programs
Figure 2. Overlay Structure of KONPACT-1
Figure 3. Overlay Structure of KONPACT-2
Figure 4. Overlay Structure and Subroutines in KONPACT-1
Figure 5. Program MAIN Program Listing
Figure 6. Program MAIN1 Program Listing

59
Figure 7. Program MAIN2 Program Listing
Figure 7. Program MAIN2 Program Listing (Concluded)
Figure 8. Program MAIN3 Program Listing
IF (NS, GT, MS1) GOS TO (MS, GT, MS2) GOS TO (LS, GT, MS3)

CALL SUBROUTINE STANK

CALL STANK (S1 (W1), S1 (V2), S1 (W1), S1 (V2), S1 (43), S1 (H6), S1 (W7),
151 (W1), S1 (V2), S1 (W1), S1 (V2), S1 (43), S1 (H6), S1 (W7),
253 (L1), S1 (L2), S1 (L3), S1 (L4), S1 (L5), S1 (L6), S1 (L7), S1 (L8),
353 (L9), S1 (L10), S1 (L11), S1 (L12), S1 (L13), S1 (L14), S1 (L15), S1 (L16), S1 (L17),
CALL SUBROUTINE VTM

RETURN TO MAIN OVERLAY

END

Figure 8. Program MAIN3 Program Listing (Concluded)
Figure 9. Program MAIN4 Program Listing
Figure 10. Program MAIN5 Program Listing
35(I(N1)+S(N4)+S(N5)+S(N6)+S(N7)+S(N8)+
45(I(N2)+S(N2)+S(N3)+S(N4)+S(N5)+S(N6)+
55(I(L1)+S(L2)+S(L1)+S(L4)+S(5)+S(L6)+
65(I(L7)+S(L8)+S(L9)+S(L10)+S(L11)+S(L12)+
75(I(L14)+S(L15)+NAM+NUM+NOM+NOH+NOH+NOH+NOH+NOH+NOH+NOH+NOH+
IF(INT.EQ.6)CALL DFRUG1,4=M4N,4HE+5=0+1W)
RETURN TO MAIN OVERLAY
END

Figure 10. Program MAIN5 Program Listing (Concluded)
Figure 11. Subroutine KORGI Flow Chart
Figure 11. Subroutine KORG1 Flow Chart (Continued)
Figure 11. Subroutine KORGI Flow Chart (Concluded)
Figure 12. Subroutine KORGI Program Listing
Figure 12. Subroutine KORGI Program Listing (Continued)
WRITE(1W=150)
150 FORMAT(1H1//;X,3HPRINT CARD SPECIFICATION END//;X+)
143 INPUT AND FINAL OUTPUT DATA WILL BE PRINTED.
1PRINT=4
C
PRINT INPUT DATA
C
152 CONTINUE
REPRINT IR
IF(I(1PRINTERNE.1).AND.(1PRINTER.1T.4))GO TO 15A
WRITE(1W=156)
156 FORMAT(1H1//;X,2A4H*** INPUT DATA CARDS ***//)
CALL IOPR(1R+1W)
REPRINT IR
C
158 CONTINUE
C
REORGANIZE INPUT DATA
C
CALL IOPR(1R+1W,JS)
C
PRINT REORGANIZED INPUT DATA
C
IF(1PRINTER.LT.6)GO TO 166
WRITE(1W=166)
166 FORMAT(1H1//;X,3H*** REORGANIZED INPUT DATA ***//)
CALL IOPR(1R+1W)
C
READ INITIALIZING INSTRUCTIONS
C
164 CONTINUE
ISYS=
DO 164 I=1,9
DO 164 J=1,20
166 SHEAD(I,J)=HRRA
168 CONTINUE
READ(1R,170)CARD
170 FORMAT(2044)
IF(CARD(1).EQ.HPRIN111)GO TO 1AB
IF(CARD(1).NE.HCONTO)GO TO 175
CALL FILE(1JN=LOCATE-0AS)
READ( )((SHEAD(I,J),J=1,20),I=1,9)
CALL FILE(JNN=NULL+LABEL)
WRITE(1W=630)
WRITE(1W=640)((SHEAD(I,J),J=1,20),I=1,9)
17A CONTINUE
GO TO 180
C
165 CONTINUE
CALL FILE(JN1+INSERT+MARK)
CALL FILE(IJO1+INSERT+MARK)
GO TO 190
C
READ SYSTEM REFERENCE DATA
C
180 CONTINUE
READ(1R,170)CARD
IF(CARD(1).NE.HREFE1G0 TO 1AB
183 CONTINUE
READ(1R,170)CARD
IF(CARD(1).EQ.HEND)GO TO 18A
DECODE(4+22+E20+CARD(1)/11..HASYNOD2)
*OR1G180
DO 18A I=1,20
185 SHEAD(NSYNO(I)=CARD(1)
GO TO 163
18B CONTINUE
WRITE(1W=430)
WRITE(1W=440)SHEAD(I,J),J=1,20),I=1,9)
C
Figure 12. Subroutine KORG1 Program Listing (Continued)
Figure 12. Subroutine KORG1 Program Listing (Continued)
CALL \textit{OVERLAY}(\&\&KON) + 400
GO TO 190

380 CONTINUE
CALL \textit{OVERLAY}(\&\&KON) + 500
GO TO 190

WRITE SYSTEM LABELS ON NFILE FOR CONTINUATION RUNS

400 CONTINUE
CALL FILE(JN+INSERT+LABEL)
WRITE(JN)((SHEAD(I,J),J=1,21),I=1,9)
CALL FILE(JN+INSERT+MARK)
WRITE(1M+30)

430 FORMAT((1X,30)), **REFERENCE OF SYSTEM LABELS **
WRITE(1M+40)((SHEAD(I,J),J=1,20),I=1,9)

440 FORMAT(1X,20A4+/
WRITE(1M+50)

450 FORMAT(1X,30), **FILE OF SYSTEM LABELS CREATED IN THIS **
WRITE(1M+60)((SHEAD(I,J),J=1,20),I=1,9)
STOP
END

Figure 12. Subroutine KORGI Program Listing (Concluded)
Figure 13. Subroutine NAMEL Flow Chart
SUBROUTINE NAMEL(NNS,NVS,DESS,UNIT,NN,O,DESO,UNIT,O,NNI,NN,1*, NAM1,2
IDESI,INITI,DES55,UNIT55,DESO,UNIT11,DESII,UNITII,NXX,NRX,NUU, NAM3
2NR,NOM,NUM,XN,NR,NNI,FLAG1111,MB,KB,N1*N)
C
C Purpose - To read, print and update namelist data for systems
C Analists - A F Konar / J K Maresh - The Honeywell Inc
C Date Written - 1975
C
C SUBPROGRAMS CALLED
C DEUG
C HPG
C FILE
C
C ARGUMENTS LIST
C NNS IN/OUT NUMBER ARRAY FOR STATE
C NVS IN/OUT VARIABLE NAME ARRAY FOR STATE
C UNIT IN/OUT DESCRIPTION ARRAY FOR STATE
C NNO IN/OUT NUMBER ARRAY FOR OUTPUT
C VNO IN/OUT VARIABLE NAME ARRAY FOR OUTPUT
C DESO IN/OUT DESCRIPTION ARRAY FOR OUTPUT
C UNITO IN/OUT UNIT ARRAY FOR OUTPUT
C N1 IN/OUT NUMBER ARRAY FOR INPUT
C VNI IN/OUT VARIABLE NAME ARRAY FOR INPUT
C DESI IN/OUT DESCRIPTION ARRAY FOR INPUT
C UNITI IN/OUT UNIT ARRAY FOR INPUT
C DESII IN/OUT STATE DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C UNITSS IN/OUT STATE UNIT ARRAY FOR ALL SUBSYSTEMS
C DECOO IN/OUT OUTPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C UNITOO IN/OUT OUTPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C DESII IN/OUT INPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C UNITII IN/OUT INPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C NXL NO OF STATE ARRAY FOR ALL SUBSYSTEMS
C NR NO OF OUTPUT ARRAY FOR ALL SUBSYSTEMS
C NUM NO OF INPUT ARRAY FOR ALL SUBSYSTEMS
C NAX INPUT MAXIMUM NUMBER OF STATES
C NR NO INPUT MAXIMUM NUMBER OF OUTPUTS
C NUM NO INPUT MAXIMUM NUMBER OF INPUTS
C N1 INPUT NUMBER OF STATES
C N1 INPUT NUMBER OF OUTPUTS
C N1 INPUT NUMBER OF INPUTS
C NFLAG INPUT CONTROLS ENTRY POINT IN THE SUBROUTINE
C NMB MAXIMUM NO OF SYSTEMS FOR COMBINING
C NMB MAXIMUM SYSTEM NO - IMPPLICIT MODEL
C
C DIMENSION NNS(NMX),NVS(NMX,2),DESS(NMX,10),UNIT(NMX,1)
C DIMENSION NNO(NRM),VNO(NRM,2),DEO(NMX,10),UNITO(NRM,1)
C DIMENSION NNI(NUN),UNI(NUN,2),DES(NUN,10),UNITII(NUN,1)
C DIMENSION DESS55(NMX,10,MB),UNITSS(NMX,4,MB)
C DIMENSION DESO(NMX,10,MB),UNITOO(NRM,4,MB)
C DIMENSION DESII(NUN,10,MB),UNITII(NUN,4,MB)
C DIMENSION NMAX(NRX),NUM(NRX)
C DIMENSION CARD(20)
C COMMON /INOUT/ IR,11,IPRINT,INSERT,LOCATE,NULL,MARK(28),JN,O,JG,JS
C COMMON /SYS/ SCODE,SOES(5),NSYS,HEAD(28),NSYS(9),SHEAD(9,28)
C
C Figure 14. Subroutine NAMEL Program Listing
Figure 14. Subroutine NAMEL Program Listing (Continued)
440 CONTINUE
   IF(JF .GE. 5) GO TO 460
   IF(JF(JF, EQ, 1)) GO TO 440
   JF(JF) = 1
   IF(SDFS(JJ), EQ, HRLANK) JFLAG = JFLAG + 1
   DES(JJ) = HRLANK
   GO TO 440
460 CONTINUE
   JF(JF) = JF(JF) + 1
   DES(JJ) = HRLANK
   ENC(4, HRLANK, HRLANK) = ENC(4, HRLANK, HRLANK) + 1
   GO TO 460
470 FORMAT (A1, I7, A1)
480 CONTINUE
   JD(JD) = JD(JD) + 1
   IF(JD(JD), EQ, JFLAG) GO TO 550
   DES(JJ) = HRLANK
   GO TO 480
500 CONTINUE
   IF(JF(JF), EQ, 6) CALL DPHUG(4, NAMEL, NAMEL + 6*10)

C FORM NAME LIST FOR OUTPUTS

NRJ(JRJ) = NRJ(JRJ) + 1
GO TO 111
NRJ(JRJ) = NRJ(JRJ) + 1
ENCOD(4, 670, VAS(JJ), JF(JF)) = ENCOD(4, 670, VAS(JJ), JF(JF)) + 1
VNO(JJ) = VNO(JJ) + 1
UNIT(JO(JO), 1) = HRLANK
UNIT(JO(JO), 2) = HRLANK
UNIT(JO(JO), 3) = HRLANK
UNIT(JO(JO), 4) = HRLANK
JD(JD) = JD(JD) + 1
JF(JF) = JF(JF) + 1
CONTINUE

C FORM NAME LIST FOR THE IMPLICIT MODEL ERROR RESPONSES

IF(JF(JF), EQ, JF(JF)) GO TO 640
NJ(JJ) = NJ(JJ) + 1
ENCOD(4, 610, DSO(JJ), HRLANK) = ENCOD(4, 610, DSO(JJ), HRLANK) + 1
IF(JF(JF), EQ, 62) JD(JD) = JD(JD) + 1
UNIT(JO(JO), 1) = UNIT(JO(JO), 1) + 1
GO TO 620
610 FORMAT (A1, A1)
615 FORMAT (A1, 6S, 5O(JJ), JF(JF))
620 UNIT(JO(JO), 1) = UNIT(JO(JO), 1) + 1
GO TO 620
640 CONTINUE
   IF(JF(JF), EQ, 6A) GO TO 660
   IF(JF(JF), EQ, 6A) GO TO 640
   JD(JD) = JD(JD) + 1
   IF(SDFS(JJ), EQ, HRLANK) JFLAG = JFLAG + 1
   DES(JJ) = SDFS(JJ)
   GO TO 640
660 CONTINUE
   JD(JD) = JD(JD) + 1
   DES(JJ) = HDR
   JD(JD) = JD(JD) + 1

Figure 14. Subroutine NAMEL Program Listing (Continued)
ENCODF(4*420+DESI(I+J)HUT,11)
END
680 CONTINUE
JW=J+1
IF(JJGT.10)GO TO 780
DESI(I+J)=HBLANK
GO TO 680
700 CONTINUE
IF(IDPNT.EQ.6)CALL DFRUIGS(6,4,NAMEL+ML+0,0,1,W)
C FORM NAME LIST FOR INPUTS
DO 906 I=1,NV
NN(I)=I
ENCODF(4*420+VNI(I+1)+HUP,11)
VNI(I+1)=NP
UNIT(I+1)=HBLANK
UNIT(I+2)=HBLANK
UNIT(I+3)=HBLANK
UNIT(I+4)=HBLANK
JF=J
JFLAG=0
840 CONTINUE
IF(JJGT.5)GO TO 366
IF(JFJLAG,EQ.11)GO TO 940
JW=J+1
IF(SDES(J),EQ.HBLANK)JFLAG=JFLAG+1
DESI(I+1)=SDES(J)
GO TO 940
860 CONTINUE
JW=J+1
DESI(I+1)=MINPU
JW=J+1
ENCODF(4*470+DESI(I+1)+HMT,11,MB)
880 CONTINUE
JW=J+1
IF(JJGT.10)GO TO 990
DESI(I+1)=HBLANK
GO TO 980
900 CONTINUE
IF(IPUNIT.EQ.6)CALL DFRUGS(6,4,NAMEL+ML+0,0,1,W)
IF(INFLAG.EQ.01)GO TO 1220
C COMBINE THE NAME LIST DATA OF SUBSYSTEMS AND OBTAIN THE NAME LIST DATA FOR THE COMBINED SYSTEM
1000 CONTINUE
II=0
DO 1040 K=1,KB
NXXK=XX(K)
DO 1040 I=1,NXXK
II=II+1
NNS(I)=II
ENCODF(4*420+VNS(I+1)+HUP,11)
VNS(I+1)=NP
DO 1020 J=1,10
1020 DESS(I+1,J)=DESSS(I,J+K)
DO 1040 J=1,14
1040 UNITS(I+1,J)=UNITSS(I,J+K)
C READ NAME LIST DATA FOR OUTPUTS OBTAINABLE FROM INTERCONNECTION EQUATIONS WRITTEN ON SCRATCH FILE JS
C BY SUBROUTINE SIMK
C READ 1600 CARD IF(CARD(1).NE.HOUTP)GO TO 1720
1050 CONTINUE

Figure 14. Subroutine NAMEL Program Listing (Continued)
Figure 14. Subroutine NAMEL Program Listing (Continued)
IF (N Nin.EU. -1) GO TO 1270  
N Nin. (N Nin.) = Nin. 
DO 1245 J=1,2 
1245 V N J (N Nin. + J ) = V N ( J ) 
DO 1270 J = 1, 10 
1270 DESI (J) = DESI 
DO 1275 J = 1.4 
1275 UNIT (J) = UNIT ( J ) 
GO TO 1260  
C  
READ NAME LIST DATA FOR INPUTS  
C  
1300 CONTINUE  
C  
(DESI (J) + J = 1, 10) , (UNIT (J) + J = 1, 4 ) 
IF (N Nin. EOU. -1) GO TO 1270  
N Nin. (N Nin.) = Nin. 
DO 1245 J=1,2 
1245 V N J (N Nin. + J ) = V N ( J ) 
DO 1270 J = 1, 10 
1270 DESI (J) = DESI 
DO 1275 J = 1.4 
1275 UNIT (J) = UNIT ( J ) 
GO TO 1300  
C  
1340 CONTINUE  
C  
IF (I N P T . EQ. 0.1) CALL DEBUG (9.4 + NAME + 4HML + 0.0 + 1W )  
C  
PRINT HEADING AND NAME LIST DATA  
C  
IF (I N P T . EQ. 0.1) GO TO 1540  
CALL -PR (HEAD.1W) 
WRITE (9, 1360) N x. N R . N U  
1360 FORMAT (//, 1X, 1R18, 66 NUMBER OF STATES =+, 12++, 1X , 186 NUMBER OF INPUTS =+, 12++, ) 
1360 FORMAT (//, 120X, 23H** NAME LIST TABLE **/) 
WRITE (1W, 1400)  
1400 FORMAT (//, 1X, 6H NAME , 6X, 13H DESCRIPTION ,  
1X, 6X, UNIT ++/) 
IF (I N P T . EQ. 6.1) CALL DEBUG (144H NAME + 4HML + 0.0 + 1W )  
C  
PRINT NAME LIST DATA FOR STATFS  
C  
WRITE (1W, 1460)  
1460 FORMAT (//, 1X, 6H STAT F S, / ) 
WRITE (1W, 1480) (N N S (I) + (V N S (I) + J ) + J = 1, 2) + (D E S I ( I ) + J = 1, 10) ,  
1 (U N I T S (I) + J ) + J = 1. 6) , 1 = 1 N X ) 
1480 FORMAT (I X, 126X, 6A4, 6X, 10A4, 4X, 4A)  
C  
PRINT NAME LIST DATA FOR OUTPUTS  
C  
WRITE (1W, 1500)  
1500 FORMAT (//, 1X, 6H OUTPUT S, / ) 
WRITE (1W, 1480) (N N S (I) + (V N S (I) + J ) + J = 1, 2) + (D E S I ( I ) + J = 1, 10) ,  
1 (U N I T S (I) + J ) + J = 1. 6) , 1 = 1 N X ) 
C  
PRINT NAME LIST DATA FOR INPUTS  
C  
WRITE (1W, 1520)  
1520 FORMAT (//, 1X, 6H INPUT S, / ) 
WRITE (1W, 1480) (N N S (I) + (V N S (I) + J ) + J = 1, 2) + (D E S I ( I ) + J = 1, 10) ,  
1 (U N I T S (I) + J ) + J = 1. 6) , 1 = 1 N X ) 
C  
CONTINUE  
C  
IF (I N P T . EQ. 6.1) CALL DEBUG (144H NAME + 4HML + 0.0 + 1W )  
C  
WRITE NAME LIST DATA ON DISK FILE  
C  
Figure 14. Subroutine NAMEL Program Listing (Continued)
CALL FILE(JN, INSERT, HEAD)
WRITE(JN, INTEX, NR, NU).
1 (NN(S(i), VN(S(i, J), J=1, 2, 10)
2 (OES(S(i, J), J=1, 2), UNIT(S(i), J, J=1, 4, 1 N, X),
3 (NNO(S(i), VN(0(S(i, J), J=1, 2, 10)
4 (OES(O(S(i, J), J=1, 2), UNIT(O(S(i), J, J=1, 4, 1 N, R),
5 (NNI(S(i), VN(I(S(i, J), J=1, 2, 10)
6 (OES(I(S(i, J), J=1, 10, UNIT(I(S(i, J), J=1, 4, 1 N, U).
CALL FILE(JN, INSERT, MARK)
IF(IPRINT.EQ.6) CALL DEBUG(I7, 4MNAME, 44L, 0, 0, I7)
RETURN

C
PRINT ERROR MESSAGE
C
200 CONTINUE
WRITE(I7, 220)
220 FORMAT(I7, //, 1X, 37D DATA CONTROL CARD SPECIFICATION ERROR)
STOP 111
1320 CONTINUE
WRITE(I7, 1330)
1330 FORMAT(I7, //, 1X, 37D ERROR IN DATA PROVIDED BY SIMK)
STOP 111
END

Figure 14. Subroutine NAMEL Program Listing (Concluded)
Start

Initialize the Dimensions and Check for Dimension Error (Call SIMK1)

Compute Simulation Matrix F (Call SIMK1)

Compute State Space Data (A, B, C, D) from Simulation Matrix F

Form the Name List Data and Write on NDATA File (Call NAMEL)

Write the State Space Data on QDATA File (Call QDIO)

Return

Figure 15. Subroutine STAMK1 Flow Chart
SUBROUTINE STAMK1(INV.X.X.F.X.X.A.X.C.D.

1.VNS.S.DESS.UNITNS.NNO.VNO.DESO.UNITJO+1.N1+1.VNI+1.DESI+1.UNITI+1.
2.MAXN+MAXXM+MAXNM+NUMN.YM.RA+51])*S2+MS3+MS4+MS5)

C C

PURPOSE - TO OBTAIN STATE SPACE MODEL OF THE VEHICLE

DESCRIBED BY SIMULATOR DECK DATA FROM LSA PROGRAM

ANALYSIS - A F KONAR / J K MAHESHWARS THE MONEYWELL INC

DATE WRITTEN - 1975

SUBPROGRAMS CALLED

DEPS

DEPNS

NOW

GDI0

DIFYNR

DEJRM

MPU

NAME

SIMK1

ARGUMENTS LIST

V A U R R E F O R C O M P U T I N G S I M U L A T I O N M A T R I X


F S I M U L A T I O N M A T R I X

A A R R A Y F O R E X T E R N A L I N P U T S

H A R R A Y F O R T R A N S I T I O N M A T R I X

C A R R A Y F O R S T A T E O U T P U T M A T R I X

VNS I N / O U T N U M B E R A R R A Y F O R S T A T E

VNO I N / O U T V A R I A B L E N A M E A R R A Y F O R S T A T E

DESS I N / O U T D E S C R I P T I O N A R R A Y F O R S T A T E

UNITNS I N / O U T U N I T A R R A Y F O R S T A T E

NUMN I N / O U T N U M B E R A R R A Y F O R O U T P U T

UNITS I N / O U T U N I T A R R A Y F O R O U T P U T

UNITO I N / O U T U N I T A R R A Y F O R O U T P U T

NNT I N / O U T N U M B E R A R R A Y F O R I N P U T

VNT I N / O U T V A R I A B L E N A M E A R R A Y F O R I N P U T

DESI I N / O U T D E S C R I P T I O N A R R A Y F O R I N P U T

UNITI I N / O U T U N I T A R R A Y F O R I N P U T


NUMN I N P U T M A X I M U M N U M B E R O F S T A T E S

NUMI N P U T M A X I M U M N U M B E R O F I N P U T S

NYO I N P U T M A X I M U M D I M E N S I O N F O R I N T E R C O N N E Q U A T I O N S


MSI I N P U T M A X I M U M D I M E N S I O N F O R S C R A T C H A R R A Y 51

MS5 I N P U T M A X I M U M D I M E N S I O N F O R S C R A T C H A R R A Y 53

MS5 I N P U T M A X I M U M D I M E N S I O N F O R S C R A T C H A R R A Y 54

NB I N P U T M A X I M U M S Y S T E M N O - I M P L I C I T M O D E L

COMMON /INOUT/ IR,IO,IPRT,INSERT,LOCATE,NULL,MARK(20)+JN+JQ+JS
COMMON /SYS/ SCODE,SESS(5)+WSYS+HEAD(20)+WSYS(0)+SHEAD(0)+20
I*PHEAN(20)
DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
DIMENSION U(XNUM)
DIMENSION A(MAXN,MAXM,RMAX,NUM),C(NMAX,NAX),D(NRM,NUM)
DIMENSION NNS(NXM),VNS(INM),DESS(2*,101+UNITNS),NNI(2*+101)
DIMENSION NNO(NRM),VNO(INM),DESO(2*,101+UNITO),NNO(2*+101)
DIMENSION NNT(NUM),VNT(INM),DESI(2*,101+UNITI),NNT(2*+101)

Figure 16. Subroutine STAMK1 Program Listing
Figure 16. Subroutine STAMK1 Program Listing (Continued)
Figure 16. Subroutine STAMK1 Program Listing (Concluded)
Figure 17. Subroutine SIMK1 Flow Chart
SUBROUTINE SIMK1(XDOT, Y, XDOTL, YL, NL, NX, NY, NU, INIT, IMS1, IMS2, IMS3, IMS4)

C PURPOSE - TO READ SIMULATOR MATRIX DATA FROM LSA AND
TO IMPLEMENT STANDARD LSA EQUATIONS
C ANALYSIS - A.F. KOVAR / J.K. VANNHOL - THE HONEYWELL INC
C DATE WRITTEN - MAY 1975
C
C SUBPROGRAMS CALLED
C DFMUG
C INPUT
C OUTPUT
C ARGUMENTS LIST
C XDOT  ARRAY FOR STATE DERIVATIVES
C Y  ARRAY FOR Y EQUATIONS
C U  ARRAY FOR STATES
C XDOTL OUTPUT ARRAY FOR DERIVATIVE OF STATE
C YL OUTPUT ARRAY FOR Y EQUATION VARIABLES
C RL OUTPUT ARRAY FOR EXTERNAL RESPONSE VARIABLES
C NX OUTPUT NUMBER OF STATES
C NY OUTPUT NUMBER OF Y EQUATIONS
C NU OUTPUT NUMBER OF INPUTS
C INIT INPUT INITIAL MODE FLAG
C T OUTPUT SAMPLE TIME
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C DIMENSION XDOT(NX), Y(NY), X(NX), U(NU), XDOTL(NX), YL(NY), RL(NR)
C
C DIMENSION STATEMENT FOR THE MATRIX DATA FROM LSA
C
COMMON /INPUT/ IR, IN, PRINT, INSERT, LOCATE, NULL, MARK(NM), JW, JO, JS
REAL LVPO(1, LVPI), RLO(LR), LUF(LU), LUE2, LDEL50, LDELS1, LDELS2
REAL LVGO(LVG), LVGI(LVGI), LVGW(LGW)
COMMON /SC/ VPVO(1, VPVI), VPD1, VPD2, VPPO, VPPO1, VPPO2, VPPO3, VPPO4
1. VDFU(OF, VDFU), VFPI(OF, VFPI), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3)
2. VPPO5(OF, VPPO5), VPPO6(OF, VPPO6), VPPO7(OF, VPPO7), VPPO8(OF, VPPO8)
3. VPPO9(OF, VPPO9), VPPO10(OF, VPPO10), VPPO11(OF, VPPO11), VPPO12(OF, VPPO12)
4. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
5. VFPI1(OF, VFPI1), VFPI2(OF, VFPI2), VFPI3(OF, VFPI3), VFPI4(OF, VFPI4)
6. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
7. VDFU5(OF, VDFU5), VDFU6(OF, VDFU6), VDFU7(OF, VDFU7), VDFU8(OF, VDFU8)
8. VDFU9(OF, VDFU9), VDFU10(OF, VDFU10), VDFU11(OF, VDFU11), VDFU12(OF, VDFU12)
9. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
A. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
B. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
C. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
D. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
E. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
F. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
G. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
H. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
I. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
J. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
K. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
L. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
M. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
N. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
O. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
P. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
Q. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
R. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
S. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
T. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
U. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
V. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
W. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
X. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
Y. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)
Z. VDFU1(OF, VDFU1), VDFU2(OF, VDFU2), VDFU3(OF, VDFU3), VDFU4(OF, VDFU4)

Figure 18. Subroutine SIMK1 Program Listing
Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Continued)
4. AND. (NWGo.LE.NUGM) AND. (NWGl.LE.NUGM)
5. AND. (NVGo.LE.NUGM) AND. (NVGl.LE.NUGM)
6. AND. (NVGo.LE.NUGM) AND. (NVG1.LE.NUGM)
7. AND. (NWGo.LE.NUGM) AND. (NWGl.LE.NUGM)
8. AND. (NRt.LE.NRTM) AND. (NFLF.NRM) AND. (NB.LE.NBM) AND.
9. (NL.LE.NLM) GO TO 13A
WRITE (14*13A)
136 FORMAT (1H1, //, 1X, 43H DIMENSION OF LSA DATA EXCEEDS THAT USFD IN
1.16H SUBROUTINE SIMK1)
STOP 111
138 CONTINUE
C
C COMPUTE SYSTEM DIMENSIONS
C
NX=NXVP+NXU+NXR*K
NU=NUC1+NUC2+NUC3+NUC4+NUC5+NUC6+NUC7+NUC8+NUC9
NO=NRT+NRW
IF (IRF.1) GO TO 17
10 = JR
IF (IPRINT.EQ.6) CALL DFBUG5.4H SIMK1,4H: 110:11W)
RETURN
C
C PRINT ERROR MESSAGE
C
140 CONTINUE
WRITE (11.165)
145 FORMAT (1H1, //, 1X, 35H DIMENSION ERROR IN SUBROUTINE SIMK1)
STOP 111
150 CONTINUE
C
C DIFFERENTIAL EQUATIONS FOR RIGID BODY VELOCITIES
C
IF (NXVP.LE.0) GO TO 264
DO 260 I=1,NXVP
XDOTL(I)=0.0
C
C FROM RIGID BODY VELOCITIES
C
DO 262 K=1,NXVP
262 XDOTL(I)=XDOTL(I)+VPVP(I,K)*X(K)+VPV2(I,K)*XDOT(K)
IF (NXR.LE.0) GO TO 160
C
C FROM RIGID BODY ATTITUDES
C
DO 156 K=1,NXR
KK=NXVP*K
156 XDOTL(I)=XDOTL(I)+VPBJ(I,K)*X(KK)+VPBJ2(I,K)*XDOT(KK)
160 CONTINUE
C
C FROM ENDING MODES
C
DO 164 K=1,NXUF
164 XDOTL(I)=XDOTL(I)+VPUE(I,K)*X(K)+VPUE2(I,K)*XDOT(KK)
168 CONTINUE
C
C FROM CONTROL SURFACE INPUTS
C
IF (NUC1.LE.0) GO TO 184
DO 172 K=1,NUC1
172 XDOTL(I)=XDOTL(I)+VPDFLS0(I,K)*U(K)
IF (NUC1.LE.0) GO TO 184
Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Continued)
FROM RIGID BODY ATTITUDES

C DO 29A K=1,NX0
   KK=NX0+K
29A CONTINUE

C FROM ENDING WINDS

C DO 30A K=1,NXUF
   KK=NXUF+K
30A CONTINUE

C FROM CONTROL SURFACE INPUTS

C IF(NUC1.LE.0) GO TO 31A
   DO 30A K=1,NUC1
30A CONTINUE

C FROM U-GUST INPUTS

C IF(NUG0.LE.0) GO TO 32A
   DO 32A K=1,NUG0
32A CONTINUE

C FROM V-GUST INPUTS

C IF(NVG0.LE.0) GO TO 35A
   DO 35A K=1,NVG0
35A CONTINUE

C FROM W-GUST INPUTS

C IF(NWG0.LE.0) GO TO 35B
   DO 35B K=1,NWG0
35B CONTINUE

C

Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Continued)
IF (NXUF.LE.0) GO TO 420

C FROM HEADING MODES

DO 416 K=1,NXUF
KK=NXUF+NXR*K
KKK=NXVF+NXR*NK+F*K
416 RL(1)=RL(1)+TOF0(KK)+TIF(1,K)+X(KK)+TOF2(KK)+XDK(MK)
420 CONTINUE

C FROM CONTROL SURFACE INPUTS

IF (NUC1.LE.0) GO TO 43A
DO 424 K=1,NUC1
424 RL(I)=RL(I)+TOFLS0(I,K)*U(K)
IF (NUC2.LE.0) GO TO 43A
DO 424 K=1,NUC2
KK=NUC1+K
428 RL(I)=RL(I)+TOFLS1(I,K)*U(K)
IF (NUC3.LE.0) GO TO 43A
DO 424 K=1,NUC3
KK=NUC1+NUC2+K
432 RL(I)=RL(I)+TOFLS2(I,K)*U(K)
436 CONTINUE
MU=NUC1+NUC2+NUC3

C FROM U-GUST INPUTS

IF (NUGO.LE.0) GO TO 44A
DO 440 K=1,NUGO
KK=MU+K
440 RL(I)=RL(I)+TOG0(I,K)*U(K)
IF (NUGI.LE.0) GO TO 44A
DO 440 K=1,NUGI
KK=MU+NUGO+K
444 RL(I)=RL(I)+TOGI(I,K)*U(K)
448 CONTINUE
MU=MU+NUGO+NUGI

C FROM V-GUST INPUTS

IF (NVGO.LE.0) GO TO 46A
DO 452 K=1,NVG0
KK=MU+K
452 RL(I)=RL(I)+TVG0(I,K)*U(K)
IF (NVGI.LE.0) GO TO 46A
DO 452 K=1,NVGI
KK=MU+NVGO+K
456 RL(I)=RL(I)+TVGI(I,K)*U(K)
460 CONTINUE
MU=MU+NVGO+NVGI

C FROM W-GUST INPUTS

IF (NWGO.LE.0) GO TO 47A
DO 464 K=1,NWGO
KK=MU+K
464 RL(I)=RL(I)+TWO(I,K)*U(K)
IF (NWGI.LE.0) GO TO 47A
DO 464 K=1,NWGI
KK=MU+NWGO+K
468 RL(I)=RL(I)+TWO(I,K)*U(K)
472 CONTINUE
MU=MU+NWGO+NWGI

C FROM STEADY U-GUST INPUTS

Figure 18. Subroutine SIMK1 Program Listing (Continued)
IF (NUG50.LE.0) GO TO 446
DO 476 K = 1,NUG50
476 RL(I) = RL(I) + TUG0(I,K)*U(KK)
IF (NUGS1.LE.0) GO TO 446
DO 492 K = 1,NUGS1
492 RL(I) = RL(I) + TVG1(I,K)*U(KK)
496 CONTINUE
MU = MU + NUGS0 + NUGS1
C FROM STEADY V-GUST INPUTS
IF (NVGS0.LE.0) GO TO 496
DO 488 K = 1,NVGS0
488 RL(I) = RL(I) + TVG0(I,K)*U(KK)
IF (NVGS1.LE.0) GO TO 496
DO 492 K = 1,NVGS1
492 RL(I) = RL(I) + TVG1(I,K)*U(KK)
496 CONTINUE
MU = MU + NVGS0 + NVGS1
C FROM STEADY W-GUST INPUTS
IF (NWGS0.LE.0) GO TO 508
DO 500 K = 1,NWGS0
500 RL(I) = RL(I) + TVGO(I,K)*U(KK)
IF (NWGS1.LE.0) GO TO 508
DO 500 K = 1,NWGS1
508 CONTINUE
MU = MU + NWGS0 + NWGS1
512 CONTINUE
516 CONTINUE
C LOAD EQUATIONS
IF (NRL.LE.0) GO TO 716
DO 712 I = 1,NRL
712 J = JRT + 1
IF (NXVP.LE.0) GO TO 604
C FROM RIGID BODY VELOCITIES
DO 600 K = 1,NXVP
600 RL(I) = RL(I) + LVPO(I,K)*X(K) + LVPI(I,K)*XDT(K)
604 CONTINUE
IF (NXH.LE.0) GO TO 612
C FROM RIGID BODY ATTITUDES
DO 608 K = 1,NXVP
608 RL(I) = RL(I) + LR0(I,K)*X(KK) + LR1(I,K)*XDT(KK)
612 CONTINUE
IF (NXUE.LE.0) GO TO 620
C FROM BENDING MODES
DO 616 K = 1,NXUF
616
Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Continued)
Figure 18. Subroutine SIMK1 Program Listing (Concluded)
Start

Initialize the Dimensions and Check for Dimension Error (Call SIMKT)

Compute Simulation Matrix F (Call SIMKT)

Compute State Space Data (A,B,C,D) from Simulation Matrix F

Form the Name List Data and Write on NDATA File (Call NAMEL)

Write the State Space Data on QDATA File (Call QDIO)

Return

Figure 19. Subroutine STAMK2 Flow Chart

102
Figure 20. Subroutine STAMK2 Program Listing
Figure 20. Subroutine STAMK2 Program Listing (Continued)
Figure 20. Subroutine STAMK2 Program Listing (Continued)
Figure 20. Subroutine STAMK2 Program Listing (Continued)
Figure 20. Subroutine STAMK2 Program Listing (Concluded)
Figure 21. Subroutine SIMKT Flow Chart
SUBROUTINE SIMKT(XNDTL,XLRL,XDOTXRIRUIU,NNX,NRF,NNU,
  IAT,BT,CDT,PRINT,H5,ROD,SNX,NY,NR,NUM,NMAX,NTFB,MST,
  NNUM,NNRM,INIT,T)

PURPOSE - TO COMBINE SUBSYSTEM BLOCKS WHICH ARE DESCRIBED BY
RATIONAL TRANSFER FUNCTIONS
ANALYSIS - A F KONAR / J K NAHES - THE HONEYWELL INC
DATE WRITTEN - 1974

SUBPROGRAMS CALLED
  DEAUG
  INIT
  PMAR
  TRANSK
  DFN
  MPAR
  TPO
  ZER

ARGUMENTS LIST
  XDOTL  OUTPUT  ARRAY FOR DERIVATIVE OF STATE
  XYL    OUTPUT  ARRAY FOR Y EQUATION VARIABLES
  XRL    OUTPUT  ARRAY FOR EXTERNAL RESPONSE VARIABLES
  NX     OUTPUT  NUMBER OF STATES
  NY     OUTPUT  NUMBER OF Y EQUATIONS
  NR     OUTPUT  NUMBER OF OUTPUTS
  NMAX   OUTPUT  BLOCK NO OF TRANSFER FUNCTION
  INIT   INPUT   INITIAL MODE FLAG
  T      OUTPUT  SAMPLE TIME

OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM

COMMON /INPUT/ IP,IM,IPRINT,PRINT,INSERT,LOCATE,NULL,MARK(20),JN,J0,JS
DIMENSION XDOTL(NX),XYL(NY),XRL(NR)
DIMENSION XDOTL(MST,NTFB),X(MST,NTFB),RI(1,NTFB),UI(1,NTFB)
DIMENSION U(NUM),NNX(NTFB),NRF(NTFB),NNU(NTFB)
DIMENSION AT(MST,NTFB),RT(MST,1,NTFB)
DIMENSION CT(1,NTFB),OT(1,1,NTFB)
DIMENSION P(NTFB,NTFB),Q(NTFB,NUM),R(NTFB,NUM),S(NTFB,NUM)
DIMENSION PRINT(2,NTFB),HS(2,4,NTFB)
DIMENSION CARD(20),IMHEAD(3)
REAL IMHEAD
DATA HC,HNC,HBLQ,MC,HDELA,MY,1/MC,HNC,HBLQ,1/HY/
DATA HUI,HNJ,HRUI,HRRI,HRU,U,HRRI,HRU/
IF(IPRINT.EQ.6)CALL DEBUG(1,4,HSIMK,4HT 1,2,1,IN)
IF(INIT,NE,0)GO TO 100

C INITIALIZE

NBK=0
T=0.0
DO 1020 I=1,2
  DO 1020 J=1,AT
  DO 1020 K=1,NTFB
1020   MS(I,J,K)=0.0
  CALL ZERO(P,NTFB,NTFB)
  CALL ZERO(Q,NTFB,NUM)
  CALL ZERO(R,NNR,NTFB)
  CALL ZERO(S,NNR,NUM)
  IF(IPRINT.EQ.3).OR.(IPRINT.GT.6)WRITE(IW,1030)
1030 FORMAT(//'20*4.1H0** TRANSFER FUNCTION DATA FOR BLOCKS ****/**)
1040 CONTINUE
READ(1R,1060)CARD

Figure 22. Subroutine SIMKT Program Listing
1060 FORMAT (20AH)
1100 FORMAT (A1,A3)
1120 FORMAT (A1,A12,A1)
    IF(CC.EQ.MC)GO TO 1040
    IF(CARD(1).EQ.MEND)GO TO 1400
    IF(CARD(1).NE.MBLOC)GO TO 1440
    BLK1=CARD(1)
    IF(CC.EQ.MC)GO TO 1040
    IF(CARD(2).EQ.MDAY)GO TO 1440
    IF(BLK2.NE.MK)GO TO 1440
    NBK=NBK0
    NBK=NBK+1
    IF(NBK.GT.MTB)GO TO 1190
    DO 1125 I=1,3
    IF((BKDI.EQ.MDELA).AND.(BKDP.EQ.MY))GO TO 1300
    IF(IPRINT.EQ.6)CALL DEBUG(2,4,HSIMK,4HT+2,0,1W)
C READ RATIONAL TRANSFER FUNCTION DATA
C CALL ZER0(PRINTZ2+MT)
    CALL INPT(PRINTZ2+MT)
    DO 1130 I=1,2
    DO 1130 J=1,MT
    MS(I,1:N)=PRINT(I,J)
    DO 1160 I=1,MT
    IF(MS(I,1:N).NE.0.0)NNX(N)=I-1
    IF(MS(I,1:N).EQ.0.0)NNX(N)=I
    1160 CONTINUE
    IF(MS(2,1:N).NE.0.0)GO TO 1200
C PRINT ERROR MESSAGE
C WRITE(LW+1180)
1180 FORMAT (H1,F9.0)WRITE(LW+1180)
1190 CONTINUE
C WRITE(LW+1195)
1195 FORMAT (H1,F9.0)WRITE(LW+1195)
C PRINT THE TRANSFER FUNCTION
C 1200 CONTINUE
    IF((IPRINT.NE.3).AND.(IPRINT.LT.5))GO TO 1040
    NM1=NNX(N)+1
    DO 1240 I=1,2
    DO 1240 J=1,NM1
    1240 PRINT(I,J)=HS(I,J)
    CALL TPM(PRINTZ1,5,MT+1,HEAD,T,WW)
    GO TO 1040
1290 CONTINUE
    IF(IPRINT.EQ.6)CALL DEBUG(3,4,HSIMK,4HT+2,0,1W)
C READ TIME DELAY SPECIFICATION
C READ ARR +1300)TO,XX,XR,UU,OMEGH,DELPHM,ND+NN
1300 FORMAT (6E12.6,2I2)
    IF(TD<=0.0)GO TO 1380
    IF(UU<=0.0)GO TO 1340
    T=UU/MM/LU
    IF(TD<=0.0)GO TO 1340
    DP TO 1380
1380 CONTINUE

Figure 22. Subroutine SIMKT Program Listing (Continued)
C PRINT ERROR MESSAGE
C
WRITE(*,1360)
1360 FORMAT(1H1,/*,1X,30TIME DELAY SPECIFICATION ERROR)
STOP 111
1380 CONTINUE
IF(OMFGM.NE.0.0)GO TO 1400
IF(NN.EQ.0.0).OR.(NN.EQ.8)GO TO 1340
CALL DFN(MS,TMFTB,PDD,N,N,N,T,IPRINT,W)
NNX(N)=ND=1
GO TO 1200
1400 CONTINUE
IF(DELPHM.LE.0.0)GO TO 1340
DO 1470 ND=2,5
NNX(N)=ND=1
NDM=ND
IF(ND.EQ.5)NDM=4
DO 1470 ND=1,NDM
CALL DFN(MS,TMFTB,PDD,N,N,N,T,IPRINT,W)
CALL PHERR(MS,TMFTB,PDD,O,F,DELPHM,IPRINT,W)
IF(DELPHM.LE.0.0)GO TO 1200
1420 CONTINUE
WRITE(*,1430)DELPHM,DELPHN
1430 FORMAT(1H1,/*,1X,3TIME DELAY SPECIFICATION CANNOT BE MET,/*
11X,20ALLWED PHASE ERROR=1E17.6,/*
21X,20ACTUAL PHASE ERROR =11E12.6,/*
GO TO 1200
1440 CONTINUE
C C PRINT ERROR MESSAGE
C WRITE(*,1460)
1460 FORMAT(1H1,/*,1X,37DATA CONTROL CARD SPECIFICATION ERROR)
STOP 111
1480 CONTINUE
IF(POINT.EQ.0.0)CALL DEBUG(4,4HSIMK,4HTIMES)
CALL PRINT(NDP)
C COMPUTE QUADRUPLES FOR ALL RLOCKS
C DO 1540 N=1,NMAX
CALL TRANSK(NNX,NNR,NNR,AT,RT,CT,DT,PRINT,MS,
SST,MF,PF,CN,NR,IPRINT,W)
1540 CONTINUE
IF(NR.EQ.0)N=1,NMAX
DO 1560 N=1,NMAX
NX=N
NNX(N)=NNX(N)
1560 NNR(N)=1
1580 CONTINUE
READ(1R,1060)CARD
IF(CARD(1).EQ.0)GO TO 1580
CALL HNAD(1.TMFTB,PDD,10)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))CALL INPT(P,MT,MTFTB)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
READ(1R,1060)CARD
IF(CARD(1).EQ.0)GO TO 1580
CALL HNAD(1.TMFTB,PDD,10)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))CALL INPT(P,MT,MTFTB)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
READ(1R,1060)CARD
IF(CARD(1).EQ.0)GO TO 1580
CALL HNAD(1.TMFTB,PDD,10)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))CALL INPT(P,MT,MTFTB)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
READ(1R,1060)CARD
C READ INTERCONNECTION QUADRUPLES AND PRINT THEM
C C PRINT ERROR MESSAGE
C WRITE(*,1460)
1460 FORMAT(1H1,/*,1X,37DATA CONTROL CARD SPECIFICATION ERROR)
STOP 111
1480 CONTINUE
IF(POINT.EQ.0.0)CALL DEBUG(4,4HSIMK,4HTIMES)
CALL PRINT(NDP)
C COMPUTE QUADRUPLES FOR ALL RLOCKS
C DO 1540 N=1,NMAX
CALL TRANSK(NNX,NNR,NNR,AT,RT,CT,DT,PRINT,MS,
SST,MF,PF,CN,NR,IPRINT,W)
1540 CONTINUE
IF(NR.EQ.0)N=1,NMAX
DO 1560 N=1,NMAX
NX=N
NNX(N)=NNX(N)
1560 NNR(N)=1
1580 CONTINUE
READ(1R,1060)CARD
IF(CARD(1).EQ.0)GO TO 1580
CALL HNAD(1.TMFTB,PDD,10)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))CALL INPT(P,MT,MTFTB)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
READ(1R,1060)CARD
IF(CARD(1).EQ.0)GO TO 1580
CALL HNAD(1.TMFTB,PDD,10)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))CALL INPT(P,MT,MTFTB)
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
IF((CARD(1).EQ.0).AND.(CARD(2).EQ.0))GO TO 1580
READ(1R,1060)CARD
C READ INTERCONNECTION QUADRUPLES AND PRINT THEM
C
Figure 22. Subroutine SIMKT Program Listing (Continued)
IF(CAD0(D1).EQ.NR) GO TO 158
GO TO 1440
1600 CONTINUE
IF(I*PRINT.EQ.61) CALL DFRUG5,4HSIMK4H, 1200*M)
C C CALCULATE NR AND NU
C DO 1640 J=1,NUM
DO 1670 I=1,NMAX
IF(0(I,J).NE.0.0) GO TO 1660
1620 CONTINUE
DO 1640 I=1,NRNM
IF(S(I,J).NE.0.0) GO TO 1660
1640 CONTINUE
NU=J-1
GO TO 1680
1660 CONTINUE
NU=NUM
1680 CONTINUE
IF(I*PRINT.EQ.0) GO TO 1740
DO 1740 I=1,NRM
DO 1700 J=1,NMAX
IF(R(I,J).NE.0.0) GO TO 1740
1700 CONTINUE
DO 1740 J=1,NU
IF(S(I,J).NE.0.0) GO TO 1760
1720 CONTINUE
NR=1-1
GO TO 1760
1740 CONTINUE
NR=NR*
1760 CONTINUE
IF(NR.GT.0) GO TO 1820
1780 CONTINUE
C C PRINT ERROR MESSAGE
C WRITE(W,1800)
1800 FORMAT(15x,3X,I3) ; INTERCONNECTION SPECIFICATION ERROR)
STOP 111
1820 CONTINUE
IF(I*PRINT.NE.3).AND.(I*PRINT.EQ.5) GO TO 1860
WRITE(W,1840)
1840 FORMAT(//28X,3X, CONNECTION DATA FOR BLOCKS */*/)
CALL MPRS(R,NRM,NRM,NAX,NMAX,T4NP)
CALL MPRS(Q,KFR,NRM,NMAX,NNU,T4AND)
CALL MPRS(S,NRM,NRM,NR,NM,NR,T4NS)
1860 CONTINUE
RETURN
100 CONTINUE
C C COMPUTE SUBSYSTEM STATES XDOT(N)=AN*X+BN*UN
C II=0
DO 251 N=1,NMAX
MX=NAX(N)
DO 200 I=1,MX
II=I+1
XDOTL(I)=.0
NBUY 100(N)
DO 201 J=1,NUX
XDOTL(I)=XDOTL(I)+BT(I,J+N)*UI(J,N)
DO 200 J=1,MX
200 XDOTL(I)=XDOTL(I)+AT(I,J+N)*UI(J,N)
251 CONTINUE

Figure 22. Subroutine SIMKT Program Listing (Continued)
**Figure 22. Subroutine SIMKT Program Listing (Concluded)**

```c
C COMPUTE INTERNAL OUTPUTS RIN=CN*AN+DN*IN
C
I=0
DO 35 N=1,NMAX
MX=NXP(N)
DO 30 I=1,MX
I=I+1
YL(I)=0.0
MXI=NX(I,N)
DO 30 J=1,MX
30 YL(I)+YL(I)+CT(I,J,N)*X(J,N)
NXI=NXU(N)
DO 30 J=1,NXI
30 YL(I)+YL(I)+DT(I,J,N)*UI(J,N)
CONTINUE
C INTERCONNECTION EQUATIONS
C
DO 230 I=1,NMAX
YL(I)=0.0
DO 230 J=1,NMAX
230 YL(I)=YL(I)+P(I,J)*RI(I,J)
DO 240 J=1,NU
240 YL(I)=YL(I)+Q(I,J)*U(I,J)
C EXTERNAL RESPONSE EQUATIONS
C
I=0
DO 280 I=1,MR
II=I+1
RL(I)=0.0
DO 270 J=1,NMAX
270 RL(I)=RL(I)+R(I,J)*RI(I,J)
DO 280 J=1,NU
280 RL(I)=RL(I)+S(I,J)*U(I,J)
RETURN
END
```
Figure 23. Subroutine TRANSK Flow Chart
SUBROUTINE TRANSK (NX, NY, NXW, NYW, AT, CT, NT, PRINT, HS*, TRANSK 2
INAM = AT, CT, NT, PRINT, HS*, TRANSK 3
TRANSK 4
TRANSK 5
TRANSK 6
TRANSK 7
TRANSK 8
TRANSK 9
TRANSK 10
TRANSK 11
TRANSK 12
TRANSK 13
TRANSK 14
TRANSK 15
TRANSK 16
TRANSK 17
TRANSK 18
TRANSK 19
TRANSK 20
TRANSK 21
TRANSK 22
TRANSK 23
TRANSK 24
TRANSK 25
TRANSK 26
TRANSK 27
TRANSK 28
TRANSK 29
TRANSK 30
TRANSK 31
TRANSK 32
TRANSK 33
TRANSK 34
TRANSK 35
TRANSK 36
TRANSK 37
TRANSK 38
TRANSK 39
TRANSK 40
TRANSK 41
TRANSK 42
TRANSK 43
TRANSK 44
TRANSK 45
TRANSK 46
TRANSK 47
TRANSK 48
TRANSK 49
TRANSK 50
TRANSK 51
TRANSK 52
TRANSK 53
TRANSK 54
TRANSK 55
TRANSK 56
TRANSK 57
TRANSK 58
TRANSK 59
TRANSK 60
TRANSK 61
TRANSK 62
TRANSK 63
TRANSK 64
TRANSK 65

Figure 24. Subroutine TRANSK Program Listing
Figure 25. Subroutine DFN Flow Chart
SUBROUTINE DFN(HT,TFN,NN,TD,IP,IN)

PURPOSE - TO PICK A PDE APPROXIMATION TO TIME DELAY

ANALYSIS - A F KONAR / J K WAMESH - THE MONEYFELL INC

DATE WRITTEN - 1975

ARGUMENTS LIST

NO  INPUT  NO OF DENOMINATOR TERMS IN THE TR FN
NN  INPUT  NO OF NUMERATOR TERMS IN THE TF FN
TD  INPUT  TIME OF TRANSPORT DELAY
IP  INPUT  PRINT CONTROL FLAG
IN  INPUT  FILE NO FOR LINE PRINTER

OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM

DIMENSION HS(2,MT+TFN)

IF(IP.EQ.6) CALL DEBUG(1,4,DFN,NH+2*0+1)
IF(NO.EQ.2) NR=NN
IF(NO.EQ.3) NB=NN+2
IF(NO.EQ.4) NR=IN+5
IF(NO.EQ.5) NR=NN-9
IF(NO.GT.13) OR (NR.LT.1) GO TO 660
IF(IP.EQ.6) CALL DEBUG(2,4,DFN,NH+2*0+1)
GO TO (51+520+530+540+553+560+570+580+590+600+610+620+630)NB

FIRST ORDER PDE APPROXIMATIONS

510 CONTINUE
HS(1,N)=1.0
HS(2,N)=TD
GO TO 650

520 CONTINUE
HS(1,N)=TD/2.0
HS(2,N)=1.0
HS(2,N)=TD/2.0
GO TO 650

SECOND ORDER PDE APPROXIMATIONS

530 CONTINUE
HS(1,N)=1.0
HS(2,N)=TD/2.0
HS(2,N)=1.0
GO TO 650

540 CONTINUE
HS(1,N)=TD/3.0
HS(2,N)=1.0
HS(2,N)=TD/6.0
HS(2,N)=2.0*TD/3.0
HS(2,N)=1.0
GO TO 650

550 CONTINUE
HS(1,N)=TD*TD/12.0
HS(1,N)=TD/3.0
HS(1,N)=1.0
HS(2,N)=TD/12.0
HS(2,N)=2.0
HS(2,N)=1.0
GO TO 650

Figure 26. Subroutine DFN Program Listing
Figure 26. Subroutine DFN Program Listing (Continued)
630 CONTINUE
HS(1:N) = T0*3/2*1.0
HS(1:N) = T0*T0/1.0
HS(1:N) = T0*T0/7.0
HS(1:N) = 1.0
HS(2:N) = T0*4/R4.0
HS(2:N) = T0*T0*7/16.0
HS(2:N) = T0**2/7.0
HS(2:N) = 1.0
650 CONTINUE
IF (PRINT.EQ.0) CALL DECHUG(3,0) DFN 141
RETURN
C C PRINT ERROR MESSAGE
C 660 CONTINUE
WRITE(18,E73)
670 FORMAT(2H1,*1X,*4D10,F10.1,F10.1,F10.1)
STOP 111
END

Figure 26. Subroutine DFN Program Listing (Concluded)
The Values of ND OMEGM, TD, The Number of Denominator Terms, Max Omega and Time Delay Passes Through the Argument List

Compute A, A2, A3, A4, C1, C2, C3, C4, C5

Compute CE and CO Corresponding to the Value of ND

Compute the Phase Error DELPH = \sqrt{\text{CE}}^2 + \text{CO}^2

Return

Figure 27. Subroutine PHERR Flow Chart
SUBROUTINE PHERR(HS,MT,MTF,IN,NM,NOMEN,T,U,DELPH,PRINT,W) 

PHERR 2

PURPOSE - TO COMPUTE PHASE ERROR OF FRACTIONAL APPROXIMATION TO TIME DELAY PHERR 3

ANALYSIS - A F KONAR & J K MANNAN - MR HONEYWELL INC PHERR 4

DATE WRITTEN - 1976 PHERR 5

SUBPROGRAMS CALLED PHERR 6

DE HUG PHERR 7

ARGUMENTS LIST PHERR 8

* NO INPUT NO OF DENOMINATOR TERMS IN THE TR FN PHERR 9

* N INPUT TRANFER FN BLOCK NO PHERR 10

* ON-GM INPUT MAXIMUM FREQUENCY FOR COMPUTING PHASE ERROR PHERR 11

* TD INPUT TIME OR TRANSPORT DELAY PHERR 12

* DE-PM OUTPUT PHASE ERROR PHERR 13

* IPINT INPUT PRINT CONTROL FLAG PHERR 14

* IW INPUT FILE NO FOR LINE PRINTER PHERR 15

* OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM PHERR 16

DIMENSION HS(3*MT,MTF)

IF (IPrint,Eq,6) CALL DEBUG(1,4+PHER+4HR*2+Q+1NW)

A=ONE&MT0 PHERR 17

A2=A*2 PHERR 18

A3=A*3 PHERR 19

C1=HS(7*1+N)*A=HS(1*1+N) PHERR 20

C2=HS(7*2+N)*A=HS(1*2+N) PHERR 21

C3=HS(7*3+N)*A=HS(1*3+N) PHERR 22

C4=HS(7*4+N)*A=HS(1*4+N) PHERR 23

C5=HS(7*5+N)*A=HS(1*5+N) PHERR 24

NO=NND-1 PHERR 25

IF (IPRINT,Eq,6) CALL DEBUG(1,3+PHER+4HR*2+Q+1N)

GO TO 110+120+130+140+150 PHERR 26

110 CONTINUE PHERR 27

CE=C2 PHERR 28

CO=C1=A PHERR 29

GO TO 150 PHERR 30

120 CONTINUE PHERR 31

CE=C3-C1*A2 PHERR 32

CO=C2=A PHERR 33

GO TO 150 PHERR 34

130 CONTINUE PHERR 35

CE=C4-C2*A2 PHERR 36

CO=C3=A-C1*A3 PHERR 37

GO TO 150 PHERR 38

140 CONTINUE PHERR 39

CE=C5-C3*A2+1*C1*A4 PHERR 40

CO=C4=A-C2*A3 PHERR 41

150 CONTINUE PHERR 42

CE=C6-C4+C2*2 PHERR 43

CO=C5+C2*2 PHERR 44

DELPH=SQRT(C6+C2*2) PHERR 45

IF (IPRINT,Eq,6) CALL DEBUG(3,4+PHER+4HR*2+Q+1N)

RETURN PHERR 46

END PHERR 47

Figure 28. Subroutine PHERR Program Listing
Start

Initialize the Dimensions and Check for Dimension Error (Call SIMK)

Compute Simulation Matrix F (Call SIMK)

Compute State Space Data \((A,B,C,D)\) from Simulation Matrix F

Form the Name List Data and Write On NDATA File (Call NAMEL)

Write the State Space Data On QDATA File (Call QDIO)

Return

Figure 29. Subroutine STAMK3 Flow Chart
Purpose - To obtain state space model from interconnection
DATA FOR SUBSYSTEMS OR TO READ DIRECTLY THE STATE SPACE DATA
ANALYSIS - A F KONAR / J K NAMSH - THE HONEYWELL INC
DATE WRITTEN - 1976

SUBPROGRAMS CALLED
DEJRN
HP
NAMEL
QIJK
IDINVR
DEJNS
MP-S
OD10
SUM

ARGUMENTS LIST
V ARRAY FOR COMPUTING SIMULATION MATRIX
W ARRAY FOR COMPUTING SIMULATION MATRIX
F SIMULATION MATRIX
ADT ARRAY FOR STATE DERIVATIVES
X ARRAY FOR STATES
RI ARRAY FOR INTERNAL OUTPUTS
UI ARRAY FOR INTERNAL INPUTS
U ARRAY FOR EXTERNAL INPUTS
RJ ARRAY FOR INTERNAL OUTPUTS FOR ALL SYSTEMS
NNK ARRAY FOR STORING SYSTEM DIMENSION NX
NNR ARRAY FOR STORING SYSTEM DIMENSION NR
NN1 ARRAY FOR STORING SYSTEM DIMENSION NU
A IN/OUT STATE TRANSITION MATRIX
B IN/OUT CONTROL INPUT MATRIX
C IN/OUT STATE OUTPUT MATRIX
D IN/OUT CONTROL OUTPUT MATRIX
NN5 IN/OUT NUMBER ARRAY FOR STATE
VNS IN/OUT VARIABLE NAME ARRAY FOR STATE
DESS IN/OUT DESCRIPTION ARRAY FOR STATE
UNITS IN/OUT UNIT ARRAY FOR STATE
NNO IN/OUT NUMBER ARRAY FOR OUTPUT
VNO IN/OUT VARIABLE NAME ARRAY FOR OUTPUT
DESO IN/OUT DESCRIPTION ARRAY FOR OUTPUT
UNITO IN/OUT UNIT ARRAY FOR OUTPUT
NNG IN/OUT NUMBER ARRAY FOR INPUT
VNI IN/OUT VARIABLE NAME ARRAY FOR INPUT
DES1 IN/OUT DESCRIPTION ARRAY FOR INPUT
UNIT1 IN/OUT UNIT ARRAY FOR INPUT
MAXN INPUT MAXIMUM ROW DIMENSION FOR SIMULATION MATRIX F
MAXM INPUT MAXIMUM COLUMN DIMENSION FOR SIMULATION MATRIX F
NXM INPUT MAXIMUM NUMBER OF STATES
NR6 INPUT MAXIMUM NUMBER OF OUTPUTS
NU6 INPUT MAXIMUM NUMBER OF INPUTS
MR6 INPUT MAXIMUM DIMENSION FOR INTERCONNECT EQUATIONS
MNX INPUT MAXIMUM OF (NR6+NU6)
MP INPUT MAXIMUM DIMENSION FOR P ARRAY
NQ INPUT MAXIMUM DIMENSION FOR Q ARRAY
NR INPUT MAXIMUM DIMENSION FOR R ARRAY
MSK1 MAXIMUM DIMENSION FOR SCRATCH ARRAY S1

Figure 30. Subroutine STAMK3 Program Listing

123
Figure 30. Subroutine STAMK3 Program Listing (Continued)

124
Figure 30. Subroutine STAMK3 Program Listing (Continued)
J=J+1
CALL ::MK(V(N1),V(P),V(N3),XDOTX+P1,II+1]U+RIN,NNX,NRR+NNU.
152(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
253(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),S3(K7),S3(K8),NX,NY,NN,NNR,NNMAX,
NNM,NNP,NNR,NNX,NUM,NNR,NNU,
J=J+N.
DO 15 J=1,N
F(I,J)=V(I)

150 C
C COMPUTE PARTIALS WRT INTERNAL INPUTS
C
DO 16 N=1,NMAX
X=NNX(NN)
J=J+1
U(I,J+N)=1.
CALL ::MK(V(N1)+V(P),V(N3),XDOTX+R1,II+1]U+RIN,NNX,NRR+NNU.
152(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
253(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),S3(K7),S3(K8),NX,NY,NN,NNR,NNMAX,
NNM,NNP,NNR,NNX,NUM,NNR,NNU,
U(I,J+N)=0.
DO 15 J=1,N
F(I,J)=V(I)

201 C
C COMPUTE PARTIALS WRT STATES
C
DO 21 J=1,NMAX
X=NNX(NN)
J=J+1
U(I,J+1)=1.
CALL ::MK(V(N1),V(P),V(N3),XDOTX+R1,II+1]U+RIN,NNX,NRR+NNU.
152(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
253(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),S3(K7),S3(K8),NX,NY,NN,NNR,NNMAX,
NNM,NNP,NNR,NNX,NUM,NNR,NNU,
F(I,J+1)=V(I)

251 F(I,J+1)=V(I)

250 C
C COMPUTE PARTIALS WRT EXTERNAL INPUTS
C
DO 25 J=1,NU
J=J+1
U(I,J)=0.
CALL ::MK(V(N1)+V(P),V(N3),XDOTX+R1,II+1]U+RIN,NNX,NRR+NNU.
152(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
253(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),S3(K7),S3(K8),NX,NY,NN,NNR,NNMAX,
NNM,NNP,NNR,NNX,NUM,NNR,NNU,
F(I,J+1)=F(I,1)

500 CONTINUE
C
C COMPUTE THE SIMULATION MATRIX
C
NNX,NNY
IF [IP(UINT,EO,A)] CALL KDPS(F,NNX,MAX,Y,V,M,T,HSTIM)
DO 51 I=1,NU
DO 52 J=1,NU
F(I,J)=F(I,J)+1.
51 F(I,J)=F(I,J)+1.
C
C XDOT SPRAY IS BEING USED AS A SCRATCH ARRAY IN TDINVR
C
CALL TDINVR(ISOL,INSLV,NNX,MAX,Y,V,M,T,HSTIM)

Figure 30. Subroutine STAMK3 Program Listing (Continued)
Figure 30. Subroutine STAMK3 Program Listing (Concluded)
Enter

INIT = 0? Yes

No

Set Up Differential Equations for the System

Set Up Internal Output Equations and Interconnection Equations

Set Up External Response Equations

Return

Initialize and Read Interconnection Data

Read Quadruple Data from QDATA File for all Systems to be Interconnected

Compute Interconnection Quadruples and write the Interconnection Data on File JS for Use by Subroutine NAMEL

Compute System Dimensions

Return

Figure 31. Subroutine SIMK Flow Chart
Figure 32. Subroutine SIMK Program Listing
Figure 32. Subroutine SIMK Program Listing (Continued)
C READMATRIX (R/P/N) INTO PROPER AREA OF RR MATRIX

1220 CONTINUE
DECODF(4+1240*CARD(2))NSY,PUMU
1240 FORMAT(1X'A3')
KR=K+1
IF(KR,GT,MRI)GO TO 1470
NSR(KR)=NSY
CALL ZEDO(R,NRM,MN)
CALL INPT(R,NRM,MN)
DO 1240 I=1,NRM
1240 RR(KR,I,J)=R(I,J)
GO TO 1040
C READ < MATRIX (R/U)

1290 CONTINUE
CALL ZEDO(S+NP+NUM)
CALL INPT(S+NR+NUM)
GO TO 1040
C READ * MATRIX (UIN/RIM) INTO PROPER AREA OF PP MATRIX

1320 CONTINUE
NSY=NSY*(NSY-1)+NSY
KP(KP+1)
IF(KP,GT,MP)GO TO 1470
NSP(KP)=NSY
CALL ZEDO(P,NN+MN)
CALL INPT(P,NN+MN)
DO 1340 I=1,NM
DO 1340 J=1,NM
1340 PP(KP,I,J)=P(I,J)
IF(IPRINT,LT,6)GO TO 1040
WRITE(IW+900)KP,NSY,NSP
CALL PREP(P,NN+MN,MM+MM,0,0,4,HP,1)
GO TO 1040
C READ ? MATRIX (UIN/U) INTO PROPER AREA OF QQ MATRIX

1360 CONTINUE
KQ=KQ+1
IF(KO,GT,MQ)GO TO 1470
NSQ(KQ)=NSY
CALL ZEDO(Q,NN+NUM)
CALL INPT(Q,NN+NUM)
DO 1380 I=1,NM
DO 1380 J=1,NM
1380 QQ(KQ,I,J)=Q(I,J)
GO TO 1040
1400 CONTINUE
C PRINT ERROR MESSAGE
C WRITE(IW+1420)
1420 FORMAT(1M1/+/1X+37HDATA CONTROL CARD SPECIFICATION ERROR)
C STOP 111
1440 CONTINUE
WRITE(IW+1440)KR,MN
1460 FORMAT(1M1/+/1X+30HTOO MANY SYSTEMS FOR COMBINING+)
C STOP 111
1470 CONTINUE
WRITE(IW+1475)

Figure 32. Subroutine SIMK Program Listing (Continued)
1475 FORMAT(1H1,'//**IX,30/MANY INTERCONNECTIONS FOR COMBINING)  SIMK 197
STOP 111  SIMK 198
C  SIMK 199
C OBTAIN QUADRUPLE DATA FOR SUBSYSTEMS FROM Q DATA FILE  SIMK 200
C  SIMK 201
C  SIMK 202
1480 CONTINUE  SIMK 203
DO 1490 I=1,20  SIMK 204
1490 CARD(I)=HEAD(I)  SIMK 205
DO 1500 N=1,KB  SIMK 206
NSY=NSYS(N)  SIMK 207
DO 1500 I=1,20  SIMK 208
HEAD(I)=SHEAD(NSY+I)  SIMK 209
1500 CONTINUE  SIMK 210
CALL FILE(0,LOCATF,HEAD)  SIMK 211
READ(10)AT,NNXN,NNR,N,NUN+  SIMK 212
1   ((AT(I,J)+I=1,NNXN),J=1,NNXN),  SIMK 213
2   ((BT(I,J)+I=1,NNXN),J=1,NNUJ,,)  SIMK 214
3   ((CT(I,J)+I=1,NNR),J=1,NAXN),  SIMK 215
NNX(N)=NNXN  SIMK 216
NNR(N)=NNRN  SIMK 217
NUN(N)=NNUJ  SIMK 218
C STORE THE IMPLICIT MODEL SYSTEM DIMENSIONS SEPARATELY  SIMK 219
C IF(NSY,NE,NSY)GO TO 1510  SIMK 220
NXI=NNXN  SIMK 221
NRI=NNRN  SIMK 222
NUI=NNUJ  SIMK 223
1510 CONTINUE  SIMK 224
C FORM INTERCONNECTION QUADRUPLES  SIMK 225
C CALL ZERO(P,MN,NN)  SIMK 226
CALL ZERO(Q,NN,NUM)  SIMK 227
CALL ZERO(R,NRM,NN)  SIMK 228
C FORM P MATRIX (RI/RJ)  SIMK 229
C KYOUT=0  SIMK 230
NM1=1  SIMK 231
NM2=0  SIMK 232
DO 1565 N=1,KB  SIMK 233
KYOUT=KYOUT+NMR(M)  SIMK 234
IF(NM1=NMR(M)+1)  SIMK 235
NMR2=NMR2+NMR(M)  SIMK 236
NSY2=NSYS(N)  SIMK 237
DO 1533 KR=1,MR  SIMK 238
IF(NSY2(EQ),NSY2)GO TO 1536  SIMK 239
1533 CONTINUE  SIMK 240
GO TO 1545  SIMK 241
1536 CONTINUE  SIMK 242
DO 1540 I=1,MR  SIMK 243
DO 1540 J=NM1,NMZ  SIMK 244
JR=J+NM1  SIMK 245
1540 R(I,J)=R(KR,I,JJ)  SIMK 246
1545 CONTINUE  SIMK 247
C FORM P MATRIX (UI/RI)  SIMK 248
C NM1=1  SIMK 249
NM2=0  SIMK 250
DO 1562 N=1,KB  SIMK 251

Figure 32. Subroutine SIMK Program Listing (Continued)
IF(N,GT,1)NN1=NN1+NN(N-1)
NN2=NN2+NNU(N)
NSY1=SYS(N)
NSY=NSY*(NSY1-1)*NSY
DO 1550 KP=1,4P
IF(NSP(KP).EQ,NSY)GO TO 1555
1550 CONTINUE
GO TO 156P
1555 CONTINUE
DO 1560 I=NN1,NN2
II=NN1+I
JJ=NN1+I
1560 P(I,J)=PP(KP,II,JJ)
1562 CONTINUE
IF(IP, INT, LT, 6) GO TO 1565
WRITE(IW,980)KP,NSP,NSP
NMP=NNP-NN1+I
NMP=NNP-NN1+I
CALL HPRS(KP,NN,NMP,NMP,0,0,4HP)
1565 CONTINUE
C
C FORM D MATRIX (UI/UI)
C
KYIN=;
NN1=1
NN2=0
DO 1570 N=1,3K
KYIN=KYIN+NNU(N)
1570 CONTINUE
IF(N,GT,1)NN1=NN1+NNU(N-1)
NN2=NN2+NNU(N)
NSY1=SYS(N)
DO 1580 KQ=1,KQ
IF(NSQ(KQ),EQ,NSY)GO TO 1575
1570 CONTINUE
GO TO 1600
1575 CONTINUE
DO 1580 J=NN1,NM2
II=NN1+J
DO 1580 J=1,NM2
1580 Q(I,J)=Q(KI,II,J)
1600 CONTINUE
IF(IP,INT,NE,6) GO TO 1610
CALL HPRS(KP,NN,NMP,KYIN+KOUT+T,4HP)
CALL HPRS(KP,NN,NMP,KYIN+NUM+KOUT+T,4HP)
CALL HPRS(KP,NN,NMP,NMP,KYOUT+T,4MR)
CALL HPRS(KP,NN,NMP,NMP,NMP,KYOUT+T,4MR)
1610 CONTINUE
C
C CALCULATE NR AND NH BY USING Q, R AND S MATRICES
C
DO 1620 J=1,NM2
DO 1620 J=1,KYIN
IF(O(J,J),NE,0.0)GO TO 1660
1620 CONTINUE
GO TO 1660
1660 CONTINUE
IF(NU, EQ, 0) GO TO 1780
DO 1690 J=1,NRM
DO 1690 J=1,KYOUT
1690 CONTINUE
Figure 32. Subroutine SIMK Program Listing (Continued)
IF (R(I,J),NE,0.0) GO TO 1740
1700 CONTINUE
  DO 1720 J=1,NU
  IF (R(I,J),NE,0.0) GO TO 1740
1720 CONTINUE
  NR=I-1
  GO TO 1760
1740 CONTINUE
  NR=NR
1760 CONTINUE
  IF (NR,GT,0.0) GO TO 1820
C     PRINT ERROR MESSAGE
C
1780 CONTINUE
  WRITE (1M*1800)
1800 FORMAT (1HI+//+1X+3HINTERCONNECTION SPECIFICATION ERROR)
  STOP 111
C     CALCULATE NX AND NY
1820 CONTINUE
  NX=0
  DO 1840 N=1,KR
    NX=NX+NX(N)
1840 CONTINUE
  NY=KYIN+KYOUT
  IF (NX.EQ.0.0) AND ((PRINT,LT,5)) GO TO 1860
  WRITE (1M*1800)
1860 FORMAT (1HI+//+1X+26HINTERCONNECTION DATA ***/**
  CALL •PRSlP(NR,N+NR,NR+N+KYIN+KYOUT+1,4,H+)
  CALL •PRSlP(NR,N+NR,N+NR+KYOUT+1,4,H)
  CALL •PRSlP(NR,N+NR,N+NR+KYOUT+1,4,H)
  CALL •PRSlP(NR,N+NR,N+NR+KYOUT+1,4,H)
C     CALCULATE NSA-NRA AND NUA
1980 CONTINUE
  NSA=NX-NA
  NRA=N+NR-1
  NUA=NU-NU1
C     WRITE INTERCONNECTION DATA ON SCRATCH FILE FOR NAMEL
C     TO FORM NAME LIST DATA
 REWIND JS
  IF (PRINT.EQ.6) WRITE (1W,1890)
1890 FORMAT (1X+1X+1X+33HINTERCONNECTION DATA ON SCRATCH FILE FOR NAMEL+)
C     CALCULATE AND WRITE DATA TO FORM NAME LIST FOR OUTPUTS
C     CARD(I)=HOUTP
  WRITE (JS*1960) CARD
  IF (PRINT.EQ.6) WRITE (1W,2003) CARD
2000 FORMAT (1X+20A4)

NRRK=1
  DO 2110 K=1,KR
    NNRK=NNRK+NRR(K)
    IF (K,GT,1) NNRKP=NNRK+K-1
  DO 2110 I=1,NU
    DO 2100 J=NNRK,NRRK
    IF (I,J),EQ,0.0) GO TO 2100
C     DO 2070 J=1,NU
    IF (I,J),EQ,0.0) GO TO 2070
C     IF (R(I,J),NE,0.0) GO TO 2100
C
Figure 32. Subroutine SIMK Program Listing (Continued)
C2020 CONTINUE
   DO 2040 JJ=NRRKP+NRRK
   IF(IJJ.EQ.1) GO TO 2400
   IF(I(JI),NE,0,0) GO TO 210^  
2040 CONTINUE
   NNRKK=0
   NNRKK=1
   DO 2070 KK=1,AKA
   NNRKK=NRKK+1
   IF(KK,GT,1) NNRKKP=NRRKPK+NNRKPK+NNRKPK+NNRKPK+NNRKPK
   IF(I(JJ),NE,0,0) GO TO 210^  
2070 CONTINUE
   JJ=J=NRRKP+1
   WRITE(JS+2800) J+KJJ
   2080 FORMAT(312)
   IF(IPINT.EQ.6) WRITE(IW+269) J+KJJ
   2090 FORMAT(15*(' ',1X))
   2100 CONTINUE
   J=1
   WRITE(JS+2800) J
   IF/IPINT.EQ.6) WRITE(IW+269) J
   C C CALCULATE AND WRITE DATA TO FORM NAME LIST FOR INPUTS
   CARD(I)=INPU
   WRITE(JS+1060) CARD
   IF/IPINT.EQ.6) WRITE(IW+269) CARD
   NNUK=1
   NNUKP=1
   DO 2200 K=1,KB
   NNUK=NNUK+1
   IF(KK,GT,1) NNUKP=NNUKP+NNUKPK+NNUKPK+NNUKPK+NNUKPK
   IF(KK,GT,1) NNUKP=NNUKP+NNUKPK+NNUKPK+NNUKPK+NNUKPK
   DO 2200 J=1,NNUK
   IF(I(JJ),EQ,0,0) GO TO 2200
   DO 2120 J=1,NNUKP+NNUK
   IF(I(JJ),EQ,0,0) GO TO 2200
   IF(I(JJ),NE,0,0) GO TO 2200
   2120 CONTINUE
   DO 2140 J=1,NNUKP+NNUK
   IF(I(JJ),EQ,0,0) GO TO 2140
   IF(I(JJ),NE,0,0) GO TO 2200
   2140 CONTINUE
   NNUKPK=0
   NNUKP=1
   DO 2170 KK=1,AKA
   NNUKPK=NNUKPK+NNUKPK
   IF(KK,GT,1) NNUKP=NNUKP+NNUKPK+NNUKPK+NNUKPK+NNUKPK
   IF(KK,GT,1) NNUKP=NNUKP+NNUKPK+NNUKPK+NNUKPK+NNUKPK
   IF(KK,EQ,1) GO TO 2170
   DO 2140 J=1,NNUKP+NNUKPK+NNUKP
   IF(I(JJ),EQ,0,0) GO TO 2200
   2160 CONTINUE
   NNUKPK=0
   NNUKP=1
   WRITE(JS+2800) J+KJJ
   IF/IPINT.EQ.6) WRITE(IW+269) J+KJJ
   2170 CONTINUE
   III=1-NNUKP+1
   WRITE(JS+2800) J+KJJ
   IF/IPINT.EQ.6) WRITE(IW+269) J+KJJ
   2200 CONTINUE
   J=1
   WRITE(JS+2800) J
   IF/IPINT.EQ.6) WRITE(IW+269) J
   CARD(I)=HEND
   WRITE(JS+1060) CARD
   IF/IPINT.EQ.6) WRITE(IW+269) CARD
Figure 32. Subroutine SIMK Program Listing (Continued)
Figure 32. Subroutine SIMK Program Listing (Concluded)
Enter

Read State Space Quadruple Data Directly from Cards

Form the Name List Data and Write on NDATA File (Call NAMEL)

Write the State Space Quadruple Data on QDATA File (Call QDIO)

Return

Figure 33. Subroutine QUADK Flow Chart
Figure 34. Subroutine QUADK Program Listing
IF ((CARD(1), NE, XDOT), OR (CARD(2), NE, XSU)) GO TO 270
C READ MATRIX (XDOT/X)
C DECODE (4+2)0+CARD(4) X, DUMMY
210 FORMAT (13x A1)
CALL INPT(A, NXM, NAV)
GO TO 190
C CONTINUE
IF ((CARD(1), NE, XDOT), OR (CARD(2), NE, XSU)) GO TO 240
C READ MATRIX (XDOT/U)
C DECODE (4+2)0+CARD(4) X, DUMMY
DECODE (4+2)30+CARD(5) DUMMY, NU
230 FORMAT (13x A1, I3)
CALL INPT(B, NXM, NUM)
GO TO 190
C CONTINUE
IF ((CARD(1), NE, XRSX)) GO TO 260
C READ C MATRIX (R/X)
C DECODE (4+2)0+CARD(4) NP, DUMMY
DECODE (4+2)30+CARD(5) DUMMY, NX
CALL INPT(C, NRM, NXM)
GO TO 190
C CONTINUE
IF ((CARD(1), NE, XHSU)) GO TO 280
C READ D MATRIX (R/U)
C DECODE (4+2)0+CARD(4) NP, DUMMY
DECODE (4+2)30+CARD(5) DUMMY, NU
CALL INPT(D, NRM, NUM)
GO TO 190
C CONTINUE
IF ((CARD(1), NE, XMND)) GO TO 320
C READ AND UPDATE NAME LIST DATA
C NFLAG=0
CALL NAMEL(NNS, NSS, DESS, UNITNS, NNO, VNO, DESO, UNITSO, VNI, VNI,
1DES1, UNIT1, DES1, UNIT10, DES10, UNIT11, NAX, NRR, NU, NU,
2NRM, NRM, NUM, NR, &NU, NFLAG, QB, KR, NB)
320 CONTINUE
IF ((CARD(1), NE, MFND)) GO TO 320
C WRITE QUADRUPLE DATA ON FILE QDATA
C ID=0
C MFLAG=0
NA=NAX 5 NRM = NR 5 NUA = NU
CALL 2DID1(A, R, C, D, DES1, NX, VR, NUA, NRM, VRH, NUA, NRM, VRH, NUA,
1NR1, NR2, NR3, NU, NU, NU, NU, T, IO, IPRINT, IN, JO, HEAD, MARK)
2LOCATE(NULL, INSERT, NFLAG)
RETURN
C PRINT ERROR MESSAGE
C 320 CONTINUE
WRITE(1, '840')
340 FORMAT (1H1, '/* 1X 37H DATA CONTROL CARD SPECIFICATION ERROR*/')
STOP 111
END

Figure 34. Subroutine QUADK Program Listing (Concluded)
Start

Initialize the Dimensions and Check for Dimension Error (Call SIMK2)

Compute Simulation Matrix F (Call SIMK2)

Compute State Space Data (A,B,C,D) from Simulation Matrix F

Form the Name List Data and Write On NDATA File (Call NAMEL)

Write the State Space Data On QDATA File (Call QDIO)

Return

Figure 35. Subroutine STAMK4 Flow Chart
SUBROUTINE STAMK4(V,W,F,U,A,R,C,D,NN,S,VS,DESS,UNITS)

C PURPOSE - TO OBTAIN STATE SPACE MODEL FROM USER WRITTEN
C SIMULATION EQUATION SUBROUTINE SIM2
C ANALYSIS - A F KONAR / J K MANNES - THE HONEYWELL INC
C DATE WRITTEN - 1974
C
C SUBPROGRAMS CALLED
C DERM
C MPS
C ODIO
C TOINV
C DERMNS
C NAMEL
C SIM2
C
C ARGUMENTS LIST
C V V ARRAY FOR COMPUTING SIMULATION MATRIX
C W W ARRAY FOR COMPUTING SIMULATION MATRIX
C F SIMULATION MATRIX
C U ARRAY FOR EXTERNAL INPUTS
C A IN/OUT STATE TRANSITION MATRIX
C R IN/OUT CONTROL INPUT MATRIX
C C IN/OUT STATE OUTPUT MATRIX
C D IN/OUT CONTROL OUTPUT MATRIX
C NN IN/OUT NUMBER ARRAY FOR STATE
C NV IN/OUT NUMBER ARRAY FOR OUTPUT
C VN IN/OUT VARIABLE NAME ARRAY FOR STATE
C DESS IN/OUT DESCRIPTION ARRAY FOR STATE
C UNITS IN/OUT UNIT ARRAY FOR STATE
C NNS IN/OUT NUMBER ARRAY FOR OUTPUT
C VNS IN/OUT VARIABLE NAME ARRAY FOR OUTPUT
C DESS IN/OUT DESCRIPTION ARRAY FOR OUTPUT
C UNITO IN/OUT UNIT ARRAY FOR OUTPUT
C NN0 IN/OUT NUMBER ARRAY FOR INPUT
C VNO IN/OUT VARIABLE NAME ARRAY FOR INPUT
C DESI IN/OUT DESCRIPTION ARRAY FOR INPUT
C UNITI IN/OUT UNIT ARRAY FOR INPUT
C MAXN INPUT MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F
C MAXM INPUT MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F
C NX INPUT MAXIMUM NUMBER OF STATES
C N+ INPUT MAXIMUM NUMBER OF OUTPUTS
C N+ INPUT MAXIMUM NUMBER OF INPUTS
C N+ INPUT MAXIMUM DIMENSION FOR INTERCONN EQUATIONS
C Mb INPUT MAXIMUM NO OF SUBSYSTEMS FOR COMBINING
C MS1 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C MS2 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C MS3 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C MS4 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C NB INPUT MAXIMUM SYSTEM NO - IMPLICIT MODEL

COMMON /SCI/ S(1)
COMMON /SC/ INOUT/ IP*1W*1PRINT*INSERT*LOCATE*NULL*MARK(20),JN*JO*JS

1*PHEAN(20)
DIMENSION V(MAXN*W(MAXN)),F(MAXN*MAXM)
DIMENSION U(NUM)
DIMENSION A(NX,NX),R(NX,NX),C(NR,NX),D(NR,NM)
DIMENSION NNS(NX),VNS(NX),DESS(NX,10),UNITS(NM),VNS(NM)
DIMENSION NNO(NR),VNO(NR),DESS(NR,10),UNITO(NRM),VNO(NRM)
COMMON /SCI/ S(1)

C DIMENSION DESS(NX,10,MR),UNITSS(NM,MR)

Figure 36. Subroutine STAMK4 Program Listing
Figure 36. Subroutine STAMK4 Program Listing (Continued)
STAMK131
STAMK132
STAMK133
STAMK134
STAMK135
STAMK136
STAMK137
STAMK138
STAMK139
STAMK140
STAMK141
STAMK142
STAMK143
STAMK144
STAMK145
STAMK146
STAMK147
STAMK148
STAMK149
STAMK150
STAMK151
STAMK152
STAMK153
STAMK154
STAMK155
STAMK156
STAMK157
STAMK158
STAMK159
STAMK160
STAMK161
STAMK162
STAMK163
STAMK164
STAMK165
STAMK166
STAMK167
STAMK168
STAMK169

Figure 36. Subroutine STAMK4 Program Listing (Concluded)
Figure 37. Subroutine SIMK2 Flow Chart
SUBROUTINE SIMK2(XDOT,Y,X,PDOT,YL,NX,NU,INIT,T)

PURPOSE - TO IMPLEMENT SIMULATION EQUATIONS FOR C5A CONTROLLER

ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC

DATE WRITTEN - 1975

ARGUMENTS LIST

- XDOT: ARRAY FOR STATE DERIVATIVES
- Y: ARRAY FOR Y EQUATIONS
- X: ARRAY FOR STATES
- U: ARRAY FOR EXTERNAL INPUTS
- PDOT: OUTPUT ARRAY FOR DERIVATIVE OF STATE
- YL: OUTPUT ARRAY FOR Y EQUATION VARIABLES
- RL: OUTPUT ARRAY FOR EXTERNAL RESPONSE VARIABLES
- NX: OUTPUT NUMBER OF STATES
- NY: OUTPUT NUMBER OF Y EQUATIONS
- NR: OUTPUT NUMBER OF OUTPUTS
- NU: OUTPUT NUMBER OF INPUTS
- INIT: INPUT INITIAL MODE FLAG
- T: OUTPUT SAMPLE TIME

DIMENSION XDOT(NX),Y(NY),X(NX),U(NU),PDOT(NX),YL(NY),RL(NR)
COMMON /INOUT/ IR,xw.inx.printf,insert locate null mark(2),jn,jq,j5
DIMENSION CARD(20)
REAL km1,km2,kaf,k0,k0,kn0,mlc1,mlc2
DATA men08,hs1,hs2,men08,hs1,hs2,men08,hs1,hs2/0
DATA mlc1,mlc2,msas,msas,msas,msas,msas,msas/0
DATA km1,km2,kaf,k0,k0,kn0/km1,km2,kaf,k0,k0,kn0/0

CHECK IF INITIALIZATION MODE
IF(INIT.NE.0) GO TO 100

SET FILTER GAINS
AP=-1, $ AR=.22361E+03
ANF=-4,0
AF=.22
AM1=-.01
AM2=-.01
ANF1=-.1, $ BHF=-1.0
ATF=4.0, $ ATF=4.0

SET CONTROLLER SWITCHES
SAS=.0, $ ALDC=.0, $ MLCl=.0, $ MLc2=.0

SET CONTROLLER GAINS
KM1=1.0/0.26
KM2=1.0/0.5791
KAF=34.0/0.26
K0=0.5
K0=0.5
KNF=0.09

READ CONTROLLER SWITCHES ON AND CONTROLLER GAIN VALUES
10 CONTINUE
READ(120)CARD
20 FORMAT(20A4)
IF(CARD(1).EQ.MEND) GO TO 80

Figure 38. Subroutine SIMK2 Program Listing
IF(CARD(4),NE.,SWITCH) GO TO 40
C
C READ CONTROLLER SWITCHES ON
C
30 CONTINUE
READ(I*R+20)CARD
IF(CARD(1),EQ.,MENDE) GO TO 10
IF(CARD(1),EQ.,HMLC1),MLC1*1.5
IF(CARD(1),EQ.,HMLC1),GO TO 30
IF(CARD(1),EQ.,HMLC1),MLC2*1.5
IF(CARD(1),EQ.,HMLC1),GO TO 30
IF(CARD(1),EQ.,HSASRA),SAS1=1.0
IF(CARD(1),EQ.,HSASRA),GO TO 30
IF(CARD(1),EQ.,MALDC1),ALDC5=1.0
IF(CARD(1),EQ.,MALDC1),GO TO 30
STOP 111
C
C READ CONTROLLER GAIN VALUES
C
40 CONTINUE
IF(CARD(4),NE., MAIN1) STOP 111
50 CONTINUE
READ(I*R+20)CARD
IF(CARD(1),EQ.,MENDE) GO TO 16
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KM1
FORMAT(E12.6)
60 IF(CARD(1),EQ.,HMLC1),GO TO 57
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KM2
IF(CARD(1),EQ.,HMLC1),GO TO 59
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KAF
IF(CARD(1),EQ.,HMLC1),GO TO 55
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KAF
IF(CARD(1),EQ.,HMLC1),GO TO 55
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KAF
IF(CARD(1),EQ.,HMLC1),GO TO 55
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KAF
IF(CARD(1),EQ.,HMLC1),GO TO 55
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KAF
IF(CARD(1),EQ.,HMLC1),GO TO 55
IF(CARD(1),EQ.,HMLC1),READ(I*R+60)KAF
IF(CARD(1),EQ.,HMLC1),GO TO 55
STOP 111
80 CONTINUE
C
C SET DIMENSIONS OF SYSTEM
C
NX=7 \ NR=3 \ NU=9 \ NY=5
C
RETURN
C
RETURN
C
C SIMULATION EQUATIONS
C
100 CONTINUE
C
C DIFFERENTIAL EQUATIONS
C
XX0TL(1)=AP*X(1)+B*X(3)
XX0TL(2)=ANF*X(2)+ALDC5*U(6)
XX0TL(3)=ANF*X(3)+MLC1*Y(2)
XX0TL(4)=AF*X(4)+ALDC5*Y(1)
XX0TL(5)=ATF*X(5)+RTF*Y(3)
XX0TL(6)=AHF*X(6)+RHF*U(2)
XX0TL(7)=AHF*X(7)+MLC2*Y(4)
C
C SUMMING POINT EQUATIONS
C
YL(1)=KAF*X(2)+ANF*U(4)
YL(2)=KNF*U(4)-U(9)

Figure 38. Subroutine SIMK2 Program Listing (Continued)
Figure 38. Subroutine SIMK2 Program Listing (Concluded)
Enter

Read State Space Data from QDATA File (Call M010)

Read Name List Data from NDATA File (Call MNAMN)

Read Scale Data (Call SORD)

Compute Scaled Quadruples (Call SCAL)

Read Response Specification Data and Write On Scratch File JS for Use by MNAMN

Response is a State

Form CS, DS by Construction

End of Response Set

Response is a Derivative of State

Form CS, DS from A1, B1

A

Response is a Response of the Original System

Form CS, DS from C1, D1

A

Response is a Derivative of the Response of the Original System

Response is a Response of the Original System

Set IMF=1 if the Response is Implicit Model Error

Form CS, DS (Call DIFFK)

If IMF=1
Compute the Implicit Model Error Rate and Truncate Implicit Model States (Call IMRATE)

A

Figure 39. Subroutine CONDK Flow Chart
Modify the Name List Data (Call MNAME)

Read Reduction and Shuffling Data (Call RSDRD)

Shuffle State Space Data and Name List Data (Call SHUFF)

Reduce the State Space Data by Truncation or Residualization (Call REDUCE)

Print Name List Data and Write on NDATA File (Call MNAME)

Print State Space Data and Write on ODATA File (Call Odio)

Return

Figure 39. Subroutine CONDK Flow Chart (Concluded)
SUBROUTINE CONDK(A,B,C,D,M,N+NS+VNS+DESS+UNITS+NNO+VNO+DESO)  CONOK 2
   LUNITO+NNI+VNI+DESI+UNITI+NNNS+VNNNS+DESN+UNITNS+NNO+VNO+DESN)  CONOK 3
   ZUNITN+NNNI+VNNI+DESNI+UNITNI+DUMMY1+DUMMY2+DUMMY3+ES+ER+NSHUF+  CONOK 4
   3NSHUF+NSHUF1+C5+DS+CW+13S+G+NN+NNN+S+NN+NMN+S+NN+NM+11+S+NM+12+S+NM+21;  CONOK 5
   (DUMMY)  CONOK 6

C PURPOSE - TO CONDITION THE STATE SPACE QUADRUPLATE DATA  CONOK 7
C RESPONSE SPECIFICATIONS, TRUNCATION AND RESIDUALIZATION  CONOK 8
C AND SHUFFLING  CONOK 9
C ANALISIS - A K KONAR / J K MAHESH - THE HONEYWELL INC  CONOK 10
C DATE WRITTEN - 1973  CONOK 11
C NOTE - PREVIOUS NAME OF THE SUBROUTINE IS RESPK  CONOK 12
C
C SUBPROGRAMS CALLED
C ODIO  CONOK 13
C DEHUG  CONOK 14
C MNNAME  CONOK 15
C ERJN  CONOK 16
C SDOO  CONOK 17
C SCAL  CONOK 18
C DIFFK  CONOK 19
C IMJATE  CONOK 20
C RSNRO  CONOK 21
C SHIFF  CONOK 22
C RENUCE  CONOK 23
C HPP  CONOK 24

C ARGUMENTS LIST
C A IN/OUT STATE TRANSITION MATRIX  CONOK 25
C B IN/OUT CONTROL INPUT MATRIX  CONOK 26
C C IN/OUT STATE OUTPUT MATRIX  CONOK 27
C D IN/OUT CONTROL OUTPUT MATRIX  CONOK 28
C CM IN/OUT MODIFIED STATE OUTPUT MATRIX  CONOK 29
C DM IN/OUT MODIFIED CONTROL OUTPUT MATRIX  CONOK 30
C NNS IN/OUT OLD NUMBER ARRAY FOR STATE  CONOK 31
C VNS IN/OUT OLD VARIABLE NAME ARRAY FOR STATE  CONOK 32
C DESS IN/OUT OLD DESCRIPTION ARRAY FOR STATE  CONOK 33
C UNIT5 IN/OUT OLD UNIT ARRAY FOR STATE  CONOK 34
C NNO IN/OUT OLD NUMBER ARRAY FOR OUTPUT  CONOK 35
C VNO IN/OUT OLD VARIABLE NAME ARRAY FOR OUTPUT  CONOK 36
C DESO IN/OUT OLD DESCRIPTION ARRAY FOR OUTPUT  CONOK 37
C UNIT0 IN/OUT OLD UNIT ARRAY FOR OUTPUT  CONOK 38
C NNN IN/OUT OLD NUMBER ARRAY FOR INPUT  CONOK 39
C VNI IN/OUT OLD VARIABLE NAME ARRAY FOR INPUT  CONOK 40
C DESI IN/OUT OLD DESCRIPTION ARRAY FOR INPUT  CONOK 41
C UNITI IN/OUT OLD UNIT ARRAY FOR INPUT  CONOK 42
C NHH IN/OUT NEW NUMBER ARRAY FOR STATE  CONOK 43
C VNN IN/OUT NEW VARIABLE NAME ARRAY FOR STATE  CONOK 44
C DESN IN/OUT NEW DESCRIPTION ARRAY FOR STATE  CONOK 45
C UNITNS IN/OUT NEW UNIT ARRAY FOR STATE  CONOK 46
C NNN0 IN/OUT NEW NUMBER ARRAY FOR OUTPUT  CONOK 47
C VNHO IN/OUT NEW VARIABLE NAME ARRAY FOR OUTPUT  CONOK 48
C DESNO IN/OUT NEW DESCRIPTION ARRAY FOR OUTPUT  CONOK 49
C UNITNO IN/OUT NEW UNIT ARRAY FOR OUTPUT  CONOK 50
C NNH1 IN/OUT NEW NUMBER ARRAY FOR INPUT  CONOK 51
C VN1 IN/OUT NEW VARIABLE NAME ARRAY FOR INPUT  CONOK 52
C DESN1 IN/OUT NEW DESCRIPTION ARRAY FOR INPUT  CONOK 53
C UNITNI IN/OUT NEW UNIT ARRAY FOR INPUT  CONOK 54
C DUMMY1 SCRATCH ARRAY  CONOK 55
C DUMMY2 SCRATCH ARRAY  CONOK 56
C DUMMY3 SCRATCH ARRAY  CONOK 57
C ES IN/OUT STATE RESIDUALIZATION ERROR MATRIX  CONOK 58
C ER IN/OUT OUTPUT RESIDUALIZATION ERROR MATRIX  CONOK 59

Figure 40. Subroutine CONDK Program Listing

150
Figure 40. Subroutine CONDK Program Listing (Continued)
CALL SORD(DUMMY1*UNITNS*UNITS*DUMMY2*UNITNO*UNITO*DUMMY3*UNITN1)
COMPUTE SCALED QUANUPLES
CALL SCAL(A*B*C*D*UNITS*DUMMY2*DUMMY3*NX*NR*NU+NX+NR+NUM+NUM+1)
READ RESPONSE SPECIFICATION DATA AND WRITE IT
ON A SCRATCH FILE IS FOR USE BY SUBROUTINE 'NAME'

110 CONTINUE
NXO=N X NRO=NR
IFLAG=0
IRR=0
JU=0
IRES=0
NCL=0
NGT=0
NCD=0

120 CONTINUE
READ(I140)CARD
WRITE(JS140)CARD
IF(CAPD(1).EQ.HENDR)GO TO 570
IRES=1
CALL FRRM2(NRESP,HRESP,CONDK)
IF(CARD(3).NE.MHMONTR).AND.(CARD(3).NE.HMUSTB).AND.(CARD(3).NE.
IHMOMA11)GO TO 240
READ INPUT SPECIFICATION AND MODIFY R AND D MATRICES
IF(CARD(3).EQ.HMONTR)IUU=1
IF(CARD(3).EQ.HMUSTB)IUU=2
IF(CARD(3).EQ.HOMMA)IUU=3
160 CONTINUE
READ(I140)CARD
WRITE(JS140)CARD
IF(CAPD(1).EQ.HENDR)GO TO 2P0
JU=JU-1
DECODE(A=340:CARD(11))NZ,K,D0
DO 180 I=1:NX
180 DUMMY1(I,JU)=B(I,K)
DO 200 I=1:NR
200 DUMMY2(I,JU)=D(I,K)
GO TO 160
220 CONTINUE
IF(IUU.EQ.1)NCL=JU
IF(IUU.EQ.2)NGT=JU-NCL
IF(IUU.EQ.3)NCD=JU-NCL-NGT
GO TO 120
240 CONTINUE
IF(IFLAG.EQ.1)GO TO 300
NUN=NCL+NGT+NCD
IF(NUN.EQ.0)GO TO 300
NUN=NUN
DO 260 I=1:NX
DO 260 J=1:NU
260 B(I,J)=DUMMY1(I,J)
DO 280 I=1:NR
DO 280 J=1:NU
280 B(I,J)=DUMMY2(I,J)
IFLAG=1
READ OUTPUT SPECIFICATION AND COMPUTE C AND D MATRICES
300 CONTINUE
IF(CARD(4).EQ.HSIGH)IRR=1
IF(CARD(4).EQ.HRFOR)IRR=2

Figure 40. Subroutine CONDK Program Listing (Continued)
IF (CA(1), EO, MBSO) IRR = 3
IF (RE, NM = 5) J = 0

320 CONTINUE

READ (I, 140) CARD
WRITE (JS, 140) CARD
IF (CA(0) (1), EO, MBSO) GO TO 550
J = J + 1
DECKID (4, 340, CARD (2)) IX, K, 0

340 FORMAT (1, [P, A1])
IX = 0 * IX = 0
IF (CA(1), EO, MBSO) GO TO 360
IX = 1 * IX = 1
IF (CA(0) (1), EO, MBSO) GO TO 360
IX = 1 * IX = 1
IF (CA(0) (1), EO, MBSO) GO TO 340
IX = 1 * IX = 1
IF (CA(0) (1), EO, MBSO) GO TO 340
CALL PRM (3, 4HRESP, 4HK * 5, 0, 1W)

360 CONTINUE
IF (IP = INT, EO, 4) CALL DEBUG (3, 4HRESP, 4HK * 5, 0, 1W)
IF (J, EO, 1100 TO 440)
IF (X, EO, 1100 TO 430)
DO 38 L = 1, NX

380 CS(J, L) = C(K, L)
DO 40 L = 1, NU

400 DS(J, L) = D(K, L)
GO TO 320

420 CONTINUE
DO 424 L = 1, NX

424 CS(J, L) = R, 0
CS(J, L) = X, 1, 0
DO 43 L = 1, NU

430 DS(J, L) = R, 0
GO TO 320

440 CONTINUE
IF (IX, EO, 1100 TO 560)
NRS = I
RS(I) = K
IF (K, EO, 4HRESP) GO TO 450

450 CONTINUE
DEFIN flags and responses for implicit model operation

C IM = 1
C IM = IM - 1

443 CM(L) = C(K, L)
DO 444 L = 1, NU

444 DM(L) = D(K, L)

450 CONTINUE
IF (IP = INT, EO, 4) CALL DEBUG (4, 4HRESP, 4HK * 5, 0, 1W)

C COMPUTE DERIVATIVES OF RESPONSES

C CALL DIFF (A, R, C, D, DUMMY1, DUMMY2, NX, NR, NU, XN, NUM, NR, NS, R, IN, 1W, 1PRINT, 1NDM1, 1NDM2, 1NDM3, 1NDM4)

C 460 CS(J, L) = C(NR, L)
DO 48 L = 1, NX

480 DS(J, L) = D(NR, L)
NR = NR - 1
GO TO 320

500 CONTINUE
DO 524 L = 1, NX

520 CS(J, L) = A(K, L)
DO 544 L = 1, NU

540 DS(J, L) = B(K, L)

Figure 40. Subroutine CONDK Program Listing (Continued)
550 CONTINUE

C COMPUTE IMPLICIT MODEL ERROR RATES AND TRUNCATE THE
C IMPLICIT MODEL

NKR=NPARA
IF (IFUM.EQ.0)GO TO 560

C COMPUTE NEW C AND D MATRICES (OUTPUTS)

NKN=NPARA

C MODIFY NAME LIST DATA

C READ SHUFFLING AND REDUCTION DATA

Figure 40. Subroutine CONDK Program Listing (Continued)
Figure 40. Subroutine CONDK Program Listing (Concluded)
Figure 41. Subroutine MNAME Flow Chart
Figure 42. Subroutine MNAME Program Listing
Figure 42. Subroutine MNAME Program Listing (Continued)
460 CONTINUE
READ IS, 40, ICARD
480 FORMAT (28A4)
IF (CARD(1), EQ., 'END') GO TO 460
JU = 3
DECODE (4*530, CARD(1:1), J, D)
500 FORMAT (A1, I2, A1)
ID = 0, J = 0
IF (CARD(1), EQ., 'DATA') GO TO 520
ID = 0, J = 14
IF (CARD(1), EQ., 'DATA') GO TO 520
ID = 1, J = 0
IF (CARD(1), EQ., 'DATA') GO TO 520
ID = 1, J = 1
IF (CARD(1), EQ., 'DATA') GO TO 520
CALL FORM (3, 4, MNAME, 4HE , A, 0, 12)
520 CONTINUE
VNO(1, 1) = J
ENCOD (4*255, VNO (1, 1) HP)
VNO (1, 2) = HP
VNO (1, 1) = VNO (1, 1)
VNO (1, 2) = VNO (1, 2)
IF (ID, EQ., 1) GO TO 500
IF (ID, EQ., 1) GO TO 500
DO 56 L = 1, 10
540 DESNO (J, L) = DESO (K, L)
DO 60 L = 1, 4
560 UNITNO (J, L) = UNITO (K, L)
GO TO 462
580 CONTINUE
DO 60 L = 1, 4
600 DESNO (J, L) = DESO (K, L)
DO 62 L = 1, 4
620 UNITNO (J, L) = UNITO (K, L)
GO TO 460
C FORM NAME LIST DATA FOR DERIVATIVES OF SPECIFIED STATES
C OR OUTPUTS
C
640 CONTINUE
DESNO (J, 1) = DESO (K, 1)
DESNO (J, 2) = DESO (K, 2)
DESNO (J, 3) = DESO (K, 3)
DESNO (J, 10) = DESO (K, 10)
UNITNO (J, 4) = UNITO (K, 4)
IF (ID, EQ., 1) GO TO 760
DO 66 L = 1, 6
LL = 3
660 DESNO (J, L) = DESO (K, L)
DO 68 L = 1, 3
680 UNITNO (J, L) = UNITO (K, L)
GO TO 463
700 CONTINUE
DO 72 L = 1, 6
LL = 3
720 DESNO (J, L) = DESO (K, L)
DO 76 L = 1, 3
760 UNITNO (J, L) = UNITO (K, L)
GO TO 463
760 CONTINUE
IF (ID, EQ., 1) CALL ORIG (5, 4, MNAME, 4HE, 5, 0, 1, 2)
C WRITE THE NEW NAME LIST TABLE ON THE DATA FILE
C
Figure 42. Subroutine MNAME Program Listing (Continued)
IF (NFILE.GE.3) CALL ERRH (1.4, MNAME + MHE. +6 +0 +1) 
CALL FILE (IN., INSPEX, HEAD) 
WRITE (JNX, NIN, NUT, NUN)
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  

Figure 42. Subroutine MNAME Program Listing (Concluded)
Assuming SS Value of Implicit Model Following Error is Zero, Compute the SS Value of the Implicit Model States

Substitute SS Value of Implicit Model States into the Implicit Model Following Error Rate

Truncate the Implicit Model States

Return

Figure 43. Subroutine IMRATE Flow Chart
SUBROUTINE IMRATE(CH,DM,CW,DW,DUMMY1,DUMMY2,NX,NR,NU)
INXM,NRM,NUM,NXA,NRA,NUA,INV,IPRINT,NDM11,NDM12,NDM21,NDM22)
PURPOSE - TO OBTAIN IMPLICIT MODEL ERROR RATES AND TRUNCATE THE IMPLICIT MODEL
ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
DATE WRITTEN - 1976
SUBPROGRAMS CALLED
MPRS
TDINVR
ARGUMENTS LIST
NX INPUT NUMBER OF STATES
NR INPUT NUMBER OF OUTPUTS
NU INPUT NUMBER OF INPUTS
NXA INPUT NO OF STATES WITHOUT IMPLICIT MODEL
NRA INPUT NO OF OUTPUTS WITHOUT IMPLICIT MODEL
NUM INPUT NO OF INPUTS WITHOUT IMPLICIT MODEL
IW INPUT FILE NO FOR LINE PRINTER
IPRINT INPUT PRINT CONTROL FLAG
OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
DIMENSION CM(NRM+NXM+1,DM(NRM+NUM,CM(NRM+NUM) + DM(NRM+NUM)
DIMENSION DUMMY1(NDM11+NDM12),DUMMY2(NDM21+NDM22)
COMPUTE STEADY STATE VALUE OF IMPLICIT MODEL STATES
FOR ZERO MODEL ERROR
NXR = NX - NXA
DO 160 I = 1, NXR
DO 123 J = 1, NXR
JJ = NXR + J
120 DUMMY1(I,J) = CW(I,J)
DO 160 J = 1, NU
JJ = NXR + J
140 DUMMY1(I, JJ) = CW(I, J)
DO 160 J = 1, NU
JJ = NXR + J
160 DUMMY1(I, JJ) = DW(I, J)
NDM = NXR
ND1 = NXR + NU
170 CONTINUE
CALL TDINVR(ISOL,ISOSOL,NDM,DUMMY1,NDM11,DUMMY2,DET)
IF ((ISOL.EQ.0).AND. (ISOSOL.EQ.1)) GO TO 170
WRITE(*,180) ISOL, ISOSOL
180 FORMAT(1X,'1X,1X,TDINVR FAILURE, 6H ISOL==',12,7H ISOSOL==',12)
WRITE(*,200)
200 FORMAT(1X,'1X,3HSTEADY STATE VALUE OF MODEL STATE CANNOT BE Computes For ZERO ERROR)
WRITE(*,220)
220 FORMAT(1X,'1X,26HMODEL STATES ARE TRUNCATED')
COMPUTE IMPLICIT MODEL ERROR RATES
240 CONTINUE
DO 260 I = 1, NR
DO 200 J = 1, NU
?60 DUMMY2(I, J) = CM(I, J)
Figure 44. Subroutine IMRATE Program Listing
DO 280 I=1,NR
DO 280 J=1,NXA
JJ=NXA+J
DO 280 K=1,NXR
KK=NXA+K
CM(I,J)=CM(I,J)+DUMMY2(I,KK)*DUMMY1(K+J)
DO 300 I=1,NR
DO 300 J=1,NU
JJ=NXA+J
DO 300 K=1,NXR
KK=NXA+K
DN(I,J)=DN(M(I,J)+DUMMY2(I,KK)*DUMMY1(K+J))
NX=NXA
IF(IPPNT.LT.6)GO TO 320
CALL UPRS(CM,NRM,NX,NR,NXA,0,0,4,HCN)
CALL UPRS(DM,NRM,NUM,NR,NU,0,4,HDN)
CALL UPRS(DUMMY1,NUM1,NDM2,0,0,4,HDN1)
320 CONTINUE
RETURN
END

Figure 44. Subroutine IMRATE Program Listing (Concluded)
Figure 45. Subroutine DIFFK Flow Chart
SUBROUTINE DIFFK(A,B,C,D,N,DUMMY1,DUMMY2,NX,NR,NUM)
INXM,NX=N,XMRS1,RS,IN=M1=ORNVM1,NOM1,NOM21,NOM22)

PURPOSE - TO OBTAIN DERIVATIVES OF RESPONSES

ANALYSIS - A F KONAP / J K MAHESW - THE HONEYWELL INC

DATE WRITTEN - 197

ARGUMENTS LIST

NX INPUT NUMBER OF STATES
NR INPUT NUMBER OF OUTPUTS
NU INPUT NUMBER OF INPUTS
NR2 INPUT NO OF RESPONSES TO BE DIFFERENTIATED
IO INPUT CONTROLS ENTRY POINT IN THE SUBROUTINE
IN INPUT FILE NO FOR LINE PRINTER

OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM

DIMENSION A(NXM,NXM),R(NXM,NUM),C(NXM,NXM),D(NXM,NUM)
DIMENSION IRS(NXM),DUMMY1(NUM11,NUM2),DUMMY2(NOM21,NOM22)
NUM=NR
IF(ID GT 1) GO TO 140

OBTAIN FIRST DERIVATIVES ONLY

FORM A MATRIX

DO 10 I=1,NRS
I=I+1
DO 10 J=1,NX
10 DUMMY(I,J)=C(I,J)+1

COMPUTE CS*A MATRIX

DO 30 I=1,NRS
30 DUMMY(I,J)=DUMMY(I,J)+DUMMY(I,J)*A(K,J)

COMPUTE NEW C MATRIX

DO 50 I=1,NRS
50 C(I,J)=DUMMY(I,J)

FORM C*B MATRIX

DO 60 I=1,NRS
60 DUMMY(I,J)=DUMMY(I,J)+DUMMY(I,J)*B(K,J)

FORM D MATRIX

DO 70 I=1,NRS
70 DUMMY(I,J)=DUMMY(I,J)

COMPUTE NEW D MATRIX

Figure 46. Subroutine DIFFK Program Listing
DO 80 I=1,NRS
DO 80 J=1,NU
I=NR+1
80 D(I:I+1)=DUMMY?(-1,J)
C CHECK IF US MATRIX IS NULL
C
DO 90 I=1,NRS
DO 90 J=1,NRS
IF(DUMMY(I,J).NE.0.0) GO TO 150
90 CONTINUE
NR=NR+NRS
RETURN
C PRINT A MESSAGE THAT THE INPUT RATES ARE NECESSARY FOR
C CORRECTLY OBTAINING THE DERIVATIVES OF THE RESPONSES
C
100 CONTINUE
WRITE(16,120)
120 FORMAT(1X,'X',/1X,'X',XSH*** THE INPUT RATES SHOULD BE INCLUDED IN TAKING THE DERIVATIVES OF THE RESPONSES ***///)
RETURN
140 CONTINUE
C OBTAIN FIRST AND SECOND DERIVATIVES
C
WRITE(16,160)
160 FORMAT(1X,'X',XSH*** THE SECOND DERIVATIVE OPTION IS NOT IMPLEMENTED///)
RETURN
END
Print Message "Cannot be Residualized"

Truncate the States
\[ AN = A_{11} - A_{12} A_{22} A_{21} \]
\[ BN = B_{11} - A_{12} A_{22} B_{21} \]
\[ CN = C_{11} - C_{12} A_{22} A_{21} \]
\[ DN = D_{11} - C_{12} A_{22} B_{21} \]

Return

Figure 47. Subroutine REDUCE Flow Chart
Figure 48. Subroutine REDUCE Program Listing
240 ER(I,J)=ER(I,J)+C(I,K)*ES(K,J)
IF(IP=INT,GE,4) CALL MORS(E,ES,XM,NUM+NX+NY+T+4*ES)
IF(IP=INT,LE,4) CALL MORS(E,ES,XM,NUM+NX+NY+T+4*ES)
260 CONTINUE

CC

REDUCE65
REDUCE66
REDUCE67
REDUCE68

REDUCE69
REDUCE70

REDUCE71
REDUCE72
REDUCE73
REDUCE74

REDUCE75
REDUCE76

REDUCE77
REDUCE78
REDUCE79
REDUCE80

REDUCE81
REDUCE82

REDUCE83
REDUCE84
REDUCE85

REDUCE86
REDUCE87
REDUCE88

REDUCE89
REDUCE90
REDUCE91

REDUCE92
REDUCE93
REDUCE94

REDUCE95
REDUCE96
REDUCE97

REDUCE98
REDUCE99
REDUCE100

REDUCE101
REDUCE102
REDUCE103

REDUCE104
REDUCE105
REDUCE106

REDUCE107
REDUCE108

REDUCE109
REDUCE110
REDUCE111

REDUCE112
REDUCE113
REDUCE114

REDUCE115
REDUCE116
REDUCE117

REDUCE118
REDUCE119
REDUCE120

REDUCE121
REDUCE122
REDUCE123

REDUCE124
REDUCE125
REDUCE126

REDUCE127
REDUCE128
REDUCE129
REDUCE130

Figure 48. Subroutine REDUCE Program Listing (Continued)
DO 480 I=1,NXR
  KK=NN+K
  460 C(I,J)=C(I,J)*DUMMY2(1,1)*DUMMY1(K,J)
  DO 480 I=1,NXR
  DO 480 J=1,NUN
  JJ=NN+J
  DO 480 K=1,NXR
  KK=NN+K
  480 D(I,J)=D(I,J)*DUMMY2(1,1)*DUMMY1(K,J)

C.C. COMPILE SS VALUE OF STATE AND OUTPUT FOR REDUCED SYSTEM
AND SUBTRACT IT FROM SS VALUE OBTAINED EARLIER TO GET
THE ERROR OF RESIDUALIZATION

IF (PRINT.LT.3) GO TO 600
  DO 580 I=1,NXR
  DO 580 J=1,NUN
  JJ=NN+J
  580 DUMMY(I,J)=B(I,J)
  NDR=NXN
  NDC=NXN+NUN
  IF (PRINT.LT.6) CALL MPSRES(DUMMY1,1,NOM11,NOM12,NDR,NDC,T4DNUMY1)
  CALL TDINVRS(ISOL,ISOL+1,NOR+-NDC*DUMMY1+4DNUMY1+DUMMY2+DET)
  IF (ISOL.GT.1) OR (ISOL.GT.1) GO TO 620

C.C. COMPUTE RESIDUALIZATION ERROR

DO 540 I=1,NXR
  DO 540 J=1,NUN
  JJ=NN+J
  540 ES(I,J)=ES(I,J)*DUMMY1(I,J)
  DO 540 I=1,NXR
  DO 540 J=1,NUN
  ER(I,J)=D(I,J)*ER(I,J)
  JJ=NN+J
  DO 540 K=1,NXR
  KK=NN+K
  540 ER(I,J)=ER(I,J)*DUMMY2(I,K) DUMMY1(K,J)
  WRITE(1W,580)
  580 FORMAT(/1X,48HRESIDUALIZATION ERROR MATRICES ES AND ER/)
  CALL MPSRES(ES+NXN+NUN+NUN+NUN+NUN+T4MES)
  CALL MPSRES(ER+NXN+NUN+NUN+NUN+NUN+T4MER)

680 CONTINUE
  NX=NUN
  NR=NUN
  RETURN

420 CONTINUE
  STOP 0838
END

Figure 48. Subroutine REDUCE Program Listing (Concluded)
Figure 49. Subroutine SCAL Flow Chart
SUBROUTINE SCAL(INX,NX,NUN,NMM,NUM)
   SCAL 2
   SCAL 3
   SCAL 4
   SCAL 5
   SCAL 6
   SCAL 7
   SCAL 8
   SCAL 9
   SCAL 10
   SCAL 11
   SCAL 12
   SCAL 13
   SCAL 14
   SCAL 15
   SCAL 16
   SCAL 17
   SCAL 18
   SCAL 19
   SCAL 20
   SCAL 21
   SCAL 22
   SCAL 23
   SCAL 24
   SCAL 25
   SCAL 26
   SCAL 27
   SCAL 28
   SCAL 29
   SCAL 30
   SCAL 31
   SCAL 32
   SCAL 33

C PURPOSE - TO COMPUTE SCALED QUADRUPLES
C ANALYSTS - A.K.KOHUL / J.K.VANESN - THE HONEYWELL INC
C DATE WRITTEN - 1976
C
C ARGUMENT LIST
C SCFS   INPUT  SCALING ARRAY FOR STATE
C SCFD   INPUT  SCALING ARRAY FOR OUTPUT
C NUN   INPUT  NUMBER OF REDUCED STATES
C NMR   INPUT  NUMBER OF REDUCED OUTPUTS
C OTHER PARAMETERS ARE Defined IN CALLING PROGRAM
C
C DIMENSION A(NX,NUN),H(NX,NUN),C(NMM,NMM),D(NMM,NMM)
C DIMENSION SCFS(NX,NUN),SCFD(NMM),SCFI(NMM)
C DO 15 I=1,NX
C DO 16 J=1,NUN
C 15 A(I,J)=SCFS(I)*SCF(J)/SCF(J)
C DO 17 I=1,NX
C DO 18 J=1,NUN
C 17 A(I,J)=SCFD(I)*SCF(J)/SCF(J)
C DO 19 I=1,NMR
C DO 20 J=1,NMR
C 19 C(I,J)=SCFO(I)*SCF(J)/SCF(J)
C DO 21 I=1,NMR
C DO 22 J=1,NMR
C 20 D(I,J)=SCFD(I)*SCF(J)/SCF(J)
RETURN
END

Figure 50. Subroutine SCAL Program Listing

173
Figure 51. Subroutine SHUFF Flow Chart
Figure 52. Subroutine SHUFF Program Listing
Figure 53. Subroutine SHUFF1 Flow Chart
SUBROUTINE SHUF1(ARCD,NSHUF,SHUF,DIHY,VR4,NCM,MRM,NR,NC,NDM,NDM1),

PURPOSE - TO SHUFFLE THE MATRIX ARCD

ANALYSTS - A R KONAR / J K MARESH - THE HONEYWELL INC

DATE WRITTEN - 1976

ARGUMENT LIST

ARCD IN/OUT MATRIX TO BE SHUFFLED
NSHUF INPUT ROW SHUFFLING ARRAY
SHUF INPUT COLUMN SHUFFLING ARRAY
NR INPUT MAXIMUM NUMBER OF ROWS
NCM INPUT MAXIMUM NUMBER OF COLUMNS
NR INPUT NUMBER OF ROWS
NC INPUT NUMBER OF COLUMNS

OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM

DIMENSION ARCD(NR,NCM),DIHY(NDM,NDM1),NSHUF(1NR,1NCM)

DO 12 I=1,NR
11=NSHUF(I)

120 DUMMY(I,J)=ARCD(I,J)

DO 14 J=1,NC

JJ=NSHUF(J)

140 ARCD(I,J)=DIHY(I,J)

RETURN

END

Figure 54. Subroutine SHUF1 Program Listing
Figure 55. Subroutine SHUF2 Flow Chart
SUBROUTINE SHUF2(NN, VN, DES, IJIT, NSHF, DUMMY3, DUMMY1)

DIMENSION VN(NM+1), DES(NM+1), IJIT(1), NSHF(1)
DIMENSION DUMMY1(NDM1+1), DUMMY2(NDM2+1)

INTEGER DUMMY1

DO 14 JJ=1, N
   NN(JJ) = NN(JJ)
   DO 22 J=1, N
      V(J) = V(J)
   DO 22 J=1, N
      DES(J) = DES(J)
   DO 22 J=1, N
      IJIT(J) = IJIT(J)
   DO 22 J=1, N
      NSHF(J) = NSHF(J)
   DO 22 J=1, N
      DUMMY1(J) = DUMMY1(J)
   DO 22 J=1, N
      DUMMY2(J) = DUMMY2(J)
   RETURN
END

Figure 56. Subroutine SHUF2 Program Listing
Figure 57. Subroutine QDIO Flow Chart
Figure 58. Subroutine QDIO Program Listing
CALL IPRS(R,NK,NUM,NX,NU,T,640)  Q010 65
CALL IPRS(C,NK,NUM,NX,NU,T,464)  Q010 66
CALL IPRS(D,NK,NUM,NX,NU,T,46E)  Q010 67
160 CONTINUE
IF(IO,NE,1) GO TO 270  Q010 69
C
C PRINT WEIGHTING MATRIX Q
C
WRITE(1,180)  Q010 71
180 FORMAT(1X,'29x,47x*** STARTING WEIGHTS FOR OPTIMAL CONTROL DESIGN.***')  Q010 74
14H ****/1  Q010 75
CALL IPRS(Q,NUM,NUM,NU,T,640)  Q010 76
200 CONTINUE
CALL FILE(J0,INSERT+1,REL)  Q010 77
IF(IO,NE,1) GO TO 210  Q010 79
C
C WRITE QUADRUPE DATA AND WEIGHTING MATRIX Q ON FILE QDATA
C
WRITE(J0+1,NX+NU,NU)  Q010 81
1((A(I,J)=[1]NXJ+J=1NX),  Q010 82
2((R(I,J)=[1]NXJ+J=1NU),  Q010 83
3((C(I,J)=[1]NPJ+J=1NX),  Q010 84
4((D(I,J)=[1]NRJ+J=1NU),  Q010 85
SNXA,NUA+NUA,MR+N3,NU1+NU2,NU3,  Q010 86
6((Q(I,J)=[1]NRJ+J=1NR1)  Q010 87
CALL FILE(J0+INSERT+MARK)  Q010 88
RETURN;  Q010 89
210 CONTINUE
C
C WRITE QUADRUPE DATA ON FILE QDATA
C
WRITE(J0+1,NX+NU,NU)  Q010 94
1((A(I,J)=[1]NXJ+J=1NX),  Q010 95
2((R(I,J)=[1]NXJ+J=1NU),  Q010 96
3((C(I,J)=[1]NPJ+J=1NX),  Q010 97
4((D(I,J)=[1]NRJ+J=1NU),  Q010 98
SNXA,NUA+NUA,MR+N3,NU1+NU2,NU3,  Q010 99
CALL FILE(J0+INSERT+MARK)  Q010 100
RETURN;  Q010 101
220 CONTINUE
IF(FLAG,NE,1) CALL ERRM(1,4+QDIO+4H,0,0,1M)  Q010 104
IF(I PRINT.EQ.0) WRITE(1,181)  Q010 105
181 FORMAT(1X,'QDIO PROGRAM LISTING (Continued)')  Q010 106
CALL FILE(J0+LOCATE+1,REL)  Q010 107
IF(IO,NE,1) GO TO 270  Q010 108
C
C READ QUADRUPE DATA AND WEIGHTING MATRIX Q FROM FILE QDATA
C
READ(J0+1,NX,NU,NU)  Q010 110
1((A(I,J)=[1]NXJ+J=1NX),  Q010 111
2((B(I,J)=[1]NXJ+J=1NU),  Q010 112
3((C(I,J)=[1]NPJ+J=1NX),  Q010 113
4((D(I,J)=[1]NRJ+J=1NU),  Q010 114
SNXA,NUA+NUA,MR+N3,NU1+NU2+NU3,  Q010 115
6((Q(I,J)=[1]NRJ+J=1NR1)  Q010 116
RETURN;  Q010 117
C
C READ QUADRUPE DATA FROM FILE QDATA
C
230 CONTINUE
READ(J0+1,NX+NU,NU)  Q010 121
1((A(I,J)=[1]NXJ+J=1NX),  Q010 122
2((B(I,J)=[1]NXJ+J=1NU),  Q010 123
3((C(I,J)=[1]NPJ+J=1NX),  Q010 124
4((D(I,J)=[1]NRJ+J=1NU),  Q010 125
SNXA,NUA+NUA,MR+N3,NU1+NU2+NU3  Q010 126
RETURN;  Q010 127

Figure 58. Subroutine QDIO Program Listing (Continued)
Figure 58. Subroutine QDIO Program Listing (Concluded)
Figure 59. Subroutine IDRO Flow Chart
SUBROUTINE IDRO(IN, IN, JS)

PURPOSE: TO FORM A LIST FILE INPUT DATA
ANALYSIS: A F THERA / J R VAIHSM - THE HONEYWELL INC
WRITE - 1974

ARGUMENT LIST
IP  INPUT  FILE NO FOR CARD HEADER BUFFER
IW  INPUT  FILE NO FOR LIST PRINTER
JS  INPUT  FILE NO FOR SCRATCH FILE

DIMENSION CARD(21), CARD2(1), CARD3(17), CARD4(20), CHEAD(6)
DATA CHEAD /4HCONS.WHSELE.4HR.TA.4HTRUN.4HRS.T1.4HSCAL/
DATA -HRHS.HNR.T4HMR.T4HM +HM +HM +HM +HM /
DATA +J5+M15+JNT +I5+H15+I5+I5+I5+I5/
DATA -X+M1+HR+M1+M1+M1+I5/
DATA -H+M1+I5+H+I5+I5+I5/
NID=H
IF(WIN = 1R
IF(WIN = JS
CARD3(2)=HEAP
CARD3(4)=HEP
NO 10 [I=5,7A
100 CARD3(1)=HR
CARD4(1)=HEMPA
NO 11 [I=2,2A
110 CARD4(1)=HHPA
C
READ CARD IMAGES FROM FILE IP AND IGNORE THE COMMENT CARDS
C
120 CONTINUE
READ(IP+14)(CARD)
140 FORMAT(2G44)
IF(IPF(IP)1560,160
160 CONTINUE
DECODE(*+17)(CARD11)CCONMUR
170 FORMAT(12+AP)
IF(CC.EQ.M则GO TO 12)
WRITE(JS+140)(CARD)
190 CONTINUE
GO TO 12)
C
READ RESPONSE SPECIFICATION DATA AND ENCODE
C INTO SIMPLER RESPONSE SPECIFICATIONS
C
200 CONTINUE
READ(10,22)(CARD)
220 FORMAT(10A1)
100
240 CONTINUE
I=1
IF(1.E810 TO 21)
IF(CARD2(1),EQ,HEMP TO 57)
IF(CARD2(1),EQ,M15GO TO 24)
IF(CARD2(1),EQ,M15GO TO 24)
IF(CARD2(1),EQ,M15GO TO 4,5
IF(CARD2(1),EQ,M15GO TO 57)
IF(CARD2(1),EQ,M15GO TO 24)
10 TO 21
I=1
IF(CARD2(1),EQ,M15GO TO 24)

Figure 60. Subroutine IDRO Program Listing
Figure 60. Subroutine IDRO Program Listing (Continued)
540 CONTINUE
READ(5,140)CARD
IF(EOF(J51).EQ.590.590)
580 CONTINUE
WRITE(19,140)CARD
GO TO 560
600 CONTINUE
ENDFILE IR
REWIND IR
REWIND JS
RETURN
C
C PRINTER ERROR MESSAGE
C
620 CONTINUE
WRITE(1W,640)CARD
640 FORMAT(1H4,1X,32HERROR IN ORGANIZING INPUT DATA//IX*80A1)
STOP 111
END

Figure 60. Subroutine IDRO Program Listing (Concluded)
Enter

Read Response Specification Data

Response Specification = Retain, Residualize, Truncate or End?

Print Error Message and Stop (Call ERPM)

Retain

Fill Up the New Order Array JNEW for State, Output and Input Variables

A

Residualize

Fill Up the New Order Array JNEW for State, and Output Variables

A

Truncate

Fill Up the New Order Array JNEW for Output Variables

A

End

Form NSHUF Array for Shuffling the State Space Data

Return

Figure 61. Subroutine RSDRD Flow Chart
SUBROUTINE RSDRD(JNEW,NSHUF,JNEWO,NSHUF0,JNEWI,NSHFUFI)
INAX,NUMAXJAX1JAX1JAX1JAX1JAX1JAX1JAX1JAX1JAX1JAX1JAX1
Z1,J,PRINT,195)

C
C PURPOSE - TO READ REDUCTION AND SHUFFLING DATA
C ANALYSIS - A F KONAR / J K MAHESHWAR / THE MONEYGALL INC
C DATE WRITTEN - 1973
C
C SUBPROGRAMS CALLED
C DE-UGR
C ERGM

C ARGUMENTS LIST
C JNEW OUTPUT ARRAY FOR NEW ORDER OF STATES
C JNEWO OUTPUT ARRAY FOR NEW ORDER OF OUTPUTS
C JNEWI OUTPUT ARRAY FOR NEW ORDER OF INPUTS
C NSHUF OUTPUT SHUFFLING ARRAY FOR STATE
C NSHUF0 OUTPUT SHUFFLING ARRAY FOR OUTPUT
C NSHFUFI OUTPUT SHUFFLING ARRAY FOR INPUT
C NX INPUT NUMBER OF STATES
C NR INPUT NUMBER OF OUTPUTS
C NU INPUT NUMBER OF INPUTS
C NXN OUTPUT NO OF STATES TO BE RETAINED AND RESIDUAL
C NXU OUTPUT NUMBER OF REDUCED STATES
C NRN OUTPUT NUMBER OF REDUCED OUTPUTS
C NR OUTPUT NO OF OUTPUTS TO BE RESIDUALIZED
C NU OUTPUT NUMBER OF REDUCED INPUTS
C IR INPUT FILE NO FOR INPUT DATA BUFFER
C IW INPUT FILE NO FOR LIVE PRINTER
C IPROTR INPUT PRINT CONTROL FLAG
C ISV OUTPUT RESIDUALIZATION FLAG
C
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C DIMENSION JNEWS(INX)+NSHUF(NX)
C DIMENSION JNEWO(INR)+NSHUF0(NX)
C DIMENSION JNEWI(INU)+NSHUFI(NX)
C DIMENSION CARD(120)
C DATA (HENDR,HTATA,HSTA,4HEND,4HTATA,4MX)
C DATA (HENDR,HTATA,HSTA,4HEND,4HTATA,4MX)
C IN=O
C IR=0
C KS=0 KR=O KU=0
C NXN=O NX=0 NXR=O NU=O NR=O NUN=O
C IF (IPRINT.EQ.6) CALL DEBUG1,4HRSRD,4MD,50,W)
C CONTINUE
C READ(1,R,40)CAR
C FORMAT(20A4)
C IF (CARD(1).EQ.HENDR) GO TO 420
C IS=1
C IF (CARD(1).EQ.HENDR) GO TO 420
C IF (CARD(3).EQ.HENDR) GO TO 420
C 1(CARD(3).EQ.HENDR) GO TO 420
C CONTINUE
C READ SHUFFLE DATA FOR THE RETAINED SYSTEM VARIABLES
C READ(1,R,180)HON
C FORMAT(14*(1X,12)
C IF (HO.EQ.HENDR) GO TO 220
C IF (HO.EQ.HENDR) GO TO 220
C CONTINUE
C FORM NEWS ARRAY FOR STATES
K=KS+1

Figure 62. Subroutine RSDRD Program Listing
Figure 62. Subroutine RSDRD Program Listing (Continued)
Figure 62. Subroutine RSDRD Program Listing (Continued)
Figure 62. Subroutine RSDRD Program Listing (Concluded)
Figure 63. Subroutine SDRD Flow Chart
SUBROUTINE SDRD(SCFS,UNITN,UNITS,SCFO,UNITNO,UNITW,SCFI,UNITI)

C

PURPOSE - TO READ SCALE DATA

ANALYST - A. F. KORNA / J. R. HANSHEL - THE MONEWALL INC

DATE WRITTEN - 1978

SUBPROGRAMS CALLED

DEBUG

ARGUMENTS LIST

SCFS OUTPUT SCALING ARRAY FOR STATE

SCFO OUTPUT SCALING ARRAY FOR OUTPUT

SCFI OUTPUT SCALING ARRAY FOR INPUT

NX INPUT NUMBER OF STATES

NU INPUT NUMBER OF OUTPUTS

IR INPUT FILE NO FOR INPUT BUFFER

IN INPUT FILE NO FOR LINE PRINTER

PRINT INPUT PRINT CONTROL FLAG

OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM

DIMENSION SCFS(NX),UNITN(NM),UNITS(NM)

DIMENSION SCFO(NP),UNITO(NP)

DIMENSION SCFI(NM),UNITI(NM)

DIMENSION UNITN(UN)

DIMENSION SCFO(UN)

DATA (ENDB<1),XBMK<1),HBMK<1),HEND<1),HMX<1),HM<1),HMU<1)

DATA S<SCAL/4HSCAL/1001

IF(IPrint.EQ.6) CALL DEBUG(1,4,SDRD,4H +50+1W)

INITIALIZE SCF ARRAY

DO 140 I=1,NX

140 SCFS(I)=1.0

DO 160 I=1,NM

160 SCFO(I)=1.0

DO 180 I=1,UN

180 SCFI(I)=1.0

READ 'VIEW SCALE AND UNIT DATA AND UPDATE SCF AND UNIT ARRAYS

READ (IR=280)HD,N,SC<(UNITN(J)+J=1)+,UNITW(J)+J=1)+

IF(IPrint.EQ.6) CALL DEBUG(2,4,SDRD,4H +50+1W)

IF(MD.EQ.ONE) RETURN

ISC=1

IF(MD,NE,XBMK) GO TO 320

FOR STATES

SCFS(N)=SC

DO 300 J=1,NM

UNITN(N+/J)=UNITW(J)

300 UNITN(N+/J)=UNITW(J)

GO TO 260

FOR OUTPUTS

SCFO(N)=SC

GO TO 320

Figure 64. Subroutine SDRD Program Listing

194
Figure 64. Subroutine SDRD Program Listing (Concluded)
Figure 65. Subroutine FILE Flow Chart
SUBROUTINE FILE (INFILF, NOME, NAME)  
FILE  2
FILE  3
FILE  4
FILE  5
FILE  6
FILE  7
FILE  8
FILE  9
FILE 10
FILE 11
FILE 12
FILE 13
FILE 14
FILE 15
FILE 16
FILE 17
FILE 18
FILE 19
FILE 20
FILE 21
FILE 22
FILE 23
FILE 24
FILE 25
FILE 26
FILE 27
FILE 28
FILE 29
FILE 30
FILE 31
FILE 32
FILE 33
FILE 34
FILE 35
FILE 36
FILE 37
FILE 38
FILE 39
FILE 40
FILE 41
FILE 42
FILE 43
FILE 44
FILE 45
FILE 46
FILE 47
FILE 48
FILE 49
FILE 50
FILE 51
FILE 52
FILE 53
FILE 54
FILE 55
FILE 56
FILE 57
FILE 58
FILE 59
FILE 60
FILE 61
FILE 62
FILE 63
FILE 64

Figure 66. Subroutine FILE Program Listing
WRITE(1W*190)NAME,FILE
190 FORMAT(1W///,///,1X,20A4,///,1X,20A)CANNOT BE FOUND ON DATA FILE *12*)
STOP 111
C
C POSITION THE FILE TO THE BEGINNING OF NEXT RECORD
C
200 CONTINUE
READ(FILE)
GO TO 153
C
C WRITE NAME ON THE FILE
C
210 REALSPACE NFILE
WRITE(FILE)NAME(1=1,20)
RETURN
220 CONTINUE
IF(HOME.EQ.,INSERTING)GO TO 230
IF(HOME.EQ.,LOCATE)RETURN
C
C WRITE END OF DATA MARK
C
220 CONTINUE
REALSPACE NFILE
WRITE(FILE)LAST(1=1,20)
RETURN
C
C PRINT ERROR MESSAGE
C
230 CONTINUE
WRITE(1W*240)NAME,NFILE
240 FORMAT(1W///,///,1X,20A4,///,1X,21A)ALREADY ON DATA FILE *12*)
STOP 111
END

Figure 66. Subroutine FILE Program Listing (Concluded)
Enter

Fill the Array IDES with the Appropriate Description for the Elements of the Transfer Function

Print the Name of the Transfer fn

Print the Transfer fn Data

Return

Figure 67. Subroutine TPR Flow Chart
SUBROUTINE TPR(H,N,F,NM,NAME,T+I)
  TPR 2
C PURPOSE - TO PRINT TRANSFER FUNCTION DATA
C PUBLISHER - A F KONAR / J K MAHESHW - THE HONEYWELL TNC
C DATE WRITTEN - 1975
C TPR 3
C ARGUMENTS LIST
C M  INPUT  TRANSFER FUNCTION
C N  INPUT  NO. OF ELEMENTS OF THE TRANSFER FN
C NM  INPUT  MAXIMUM NO. OF ELEMENTS OF THE TRANSFER FN
C NAME  INPUT  NAME OF THE TRANSFER FN
C T  INPUT  SAMPLE TIME
C I+1  INPUT  FILE NO. FOR LINE PRINTER
C TPR 4
C LABELLED COMMON LIST
C DES  LOCAL  ARRAY FOR DESCRIPTION OF THE TRANSFER FN
C M1  LOCAL  CONSTANT
C C  LOCAL  INDEX
C K  LOCAL  INDEX
C TPR 5
C DIMENSION H2+NM),NAME(3))
C COMMON /SCI/ DES(6)M1+1,K
C TPR 6
C CHECK FOR DIMENSION ERROR
C IF(NM.NE.6,OR.(M1.GT.NM))GO TO 260
C TPR 7
C FILL THE ARRAY IDES WITH THE APPROPRIATE DESCRIPTION FOR THE
C ELEMENTS OF THE TRANSFER FUNCTION
C TPR 8
C IF(T.EQ.0.0)GO TO 120
C IDES(1+1)=H5**5  IDES(1+2)=H4M TER $ IDES(1+3)=H4M
C IDES(2+1)=H5**4  IDES(2+2)=H4M TER $ IDES(2+3)=H4M
C IDES(3+1)=H5**3  IDES(3+2)=H4M TER $ IDES(3+3)=H4M
C IDES(4+1)=H5**2  IDES(4+2)=H4M TER $ IDES(4+3)=H4M
C IDES(5+1)=H5**1  IDES(5+2)=H4M TER $ IDES(5+3)=H4M
C IDES(6+1)=H5**0  IDES(6+2)=H4M TER $ IDES(6+3)=H4M
C GO TO 140
C TPR 9
C END CONTINUE
C TPR 10
C PRINT THE NAME OF THE TRANSFER FN
C IF(T.EQ.0.0)WRITE(1W+160)NAME
C TPR 11
160 FORMAT(/X,3A4)
C TPR 12
IF(T.EQ.0.0)WRITE(1W+160)NAME,T
C TPR 13
180 FORMAT(/X,3A43H(T=..,6+1M))
C TPR 14
C END CONTINUE
C TPR 15
C PRINT THE TRANSFER FN
C TPR 16
C M=6+I+1
C WRITE(1W+200)((IDES(I+1)+K+1+3)+I=M+6)
C TPR 17
200 FORMAT(/X,3A443H(T=..,6+5G14,6))
C TPR 18
WRITE(1W+200)H(I+1+1)=M+NE
C TPR 19
220 FORMAT(/X,3A443H(T=..,6+5G14,6))
C TPR 20
WRITE(1W+200)H(I+2+1)=M+NE
C TPR 21
240 FORMAT(/X,3A443H(T=..,6+5G14,6))
C TPR 22
C TPR 23
C TPR 24
C TPR 25
C TPR 26
C TPR 27
C TPR 28
C TPR 29
C TPR 30
C TPR 31
C TPR 32
C TPR 33
C TPR 34
C TPR 35
C TPR 36
C TPR 37
C TPR 38
C TPR 39
C TPR 40
C TPR 41
C TPR 42
C TPR 43
C TPR 44
C TPR 45
C TPR 46
C TPR 47
C TPR 48
C TPR 49
C TPR 50
C TPR 51
C TPR 52
C TPR 53
C TPR 54
C TPR 55
C TPR 56
C TPR 57
C TPR 58
C TPR 59
C TPR 60
C TPR 61
C TPR 62
C TPR 63
C TPR 64

Figure 68. Subroutine TPR Program Listing
RETURN.

C PRINT ERROR MESSAGE
C
250 CONTINUE
WRITE(*,260)
260 FORMAT(1X,'DIMENSION ERROR DETECTED BY SUBROUTINE TPR')
STOP 111
END

Figure 68. Subroutine TPR Program Listing (Concluded)
Figure 69. Subroutine HPR Program Listing
SUBROUTINE IDPR(IR, IW)

PURPOSE: TO PRINT INPUT DATA

ANALYSIS: A F KONAR / J K YAHEM - THE MONEYWELL INC

DATE WRITTEN: 1976

ARGUMENTS LIST

IR: INPUT
IW: INPUT

FILE NO FOR INPUT DATA BUFFER
FILE NO FOR LINE PRINTER

DIMENSION CARD(70)

REWIN IR

120 CONTINUE
READ(IR+143)CARD

140 FORMAT(28A4)

160 CONTINUE
WRITE(IW+180)CARD

180 FORMAT(1X+2)A4

GO TO 120

200 CONTINUE
REWIN IR

RETURN

END

Figure 70. Subroutine IDPR Program Listing
SUBROUTINE MPRS(A,H,INDEX,C,TITLE)

DIMENSION A(H),INDEX(H)

! --- T --- PRINT MATRIX NAME
! --- A --- AN ARRAY / J < H / MFSM - THE MESSYWELL INC

DATA TITLE / 'MFSM' /

! --- ARGUMENTS LIST
! --- A --- ARRAY
! --- H --- NUMBER OF ROWS
! --- INDEX --- INDEX TO ARRAY
! --- C --- COLUMN NAME
! --- TITLE --- TITLE FOR THE MATRIX

IF (NCR .LT. NC) THEN
    CALL MPRS(T,INDEX,A,H,C,TITLE)
ELSE
    CALL MPRS(T,INDEX,A,H,C,TITLE)
ENDIF

RETURN TO CALLING PROGRAM

Figure 71. Subroutine MPRS Program Listing
Figure 71. Subroutine MPRS Program Listing (Concluded)
SUBROUTINE MPRSI(A,NRM,NCM,NRN,NCM,NHEAD)

C PURPOSE - TO PRINT LIST MATRIX DATA
C ANALYST - A E KONAG / J K JAMES - THE HONEYWELL INC
C DATE WRITTEN - 1972
C
C ARGUMENTS LIST
C A INPUT MATRIX DATA
C NRM INPUT MAXIMUM NUMBER OF ROWS
C NCM INPUT NUMBER OF COLUMNS
C NR INPUT NUMBER OF ROWS
C NC INPUT NUMBER OF COLUMNS
C NHEAD INPUT MATRIX TITLE OR NAME
C
DIMENSION A(NRM,NCM)
COMMON /NRM/ IP,ITU,PRINT,INC,INSERT,LOCATE,NULL,MARK(20)
IF (INC.EQ.1.NEQ.7) .AND. ((PRINT.NE.9)) RETURN

C WRITE NAME AND SIZE OF THE MATRIX
WRITE(14,80)NRM,NCM,NHEAD
80 FORMAT (/A80/N80/N80)
JC=0
100 IF (JC.GE.NC) RETURN
JC=JC+7
IF (JC.GT.NC) JC=NC
JC=JC-1C+1

C WRITE COLUMN HEADINGS
WRITE(15,160)JC
160 FORMAT (/A15)WRITE(15,170)
170 FORMAT (/)
DO 18 IC=1,NC

C WRITE ROW HEADINGS
WRITE(15,180)JC
180 FORMAT (/A15)WRITE(15,190)
190 FORMAT (/A15)
RETURN
END

Figure 72. Subroutine MPRSI Program Listing
SUBROUTINE ZERO (A, NRM, NCM)

PURPOSE: TO ZERO THE ELEMENTS OF A MATRIX
ANALYSIS: A F KONAR / J R WAKESH - THE HONEYWELL INC
DATE WRITTEN: 1975

ARGUMENTS LIST
- A: OUTPUT MATRIX DATA
- NRM: INPUT MAXIMUM NUMBER OF ROWS
- NCM: INPUT MAXIMUM NUMBER OF COLUMNS

DIMENSION A(NRM, NCM)
DO 12 I = 1, NRM
DO 12 J = 1, NCM
120 A(I, J) = 0
RETURN
END

Figure 73. Subroutine ZERO Program Listing
SUBROUTINE INPT(A+II+JJ)

PURPOSE - TO READ NON ZERO ELEMENTS OF A MATRIX

ARGUMENTS LIST

A    OUTPUT     MATRIX DATA
II   INPUT      MAXIMUM NO OF ROWS
JJ   INPUT      MAXIMUM NO OF COLUMNS

DIMENSION A(I+II),I0(5),J0(5),Y0(5)
1 READ(5,2)(ID(1),J0(1),Y0(1),1=1+5)
2 FORMAT(5(252+EI2,5))
   IF(ID(1)<10+10+4)
3   DO 6 I=1+5
   IF(ID(L))4+1+4
   CONTINUE
   I=ID(L)
   J=J0(L)
   A(I+J)=Y0(L)
4 CONTINUE
6 CONTINUE
   GO TO 1
10 CONTINUE
RETURN
END

Figure 74. Subroutine INPT Program Listing
SUBROUTINE INPT1(R,NR,NC,IR)
C PURPOSE - TO READ ISA MATRIX DATA
C ANALYSTS - A F KOSAR / J K YAMASH - THE HONEYWELL INC
C DATE - WRITTEN - 1976
C ARGUMENTS LIST
C A OUTPUT MATRIX DATA
C NR INPUT MAXIMUM NUMBER OF ROWS
C NC INPUT MAXIMUM NUMBER OF COLUMNS
C IR INPUT NUMBER OF COLUMNS
C
DIMENSION A(NR,NC)
READ(1)(A(I,1) = I = 1 + NC)*1 = 1 + NR)
120 FORMAT(AG10.3)
RETURN
END

Figure 75. Subroutine INPT1 Program Listing
Figure 76. Subroutine DEBUG Program Listing
SUBROUTINE ERM(N+1,A2+1,N1+N2+1)

PURPOSE - TO PRINT ERROR MESSAGE
ANALYSIS - A F KONAR / J K MAHESHW - THE HONEYWELL INC
DATE WRITTEN - 1975

ARGUMENTS LIST
C N INPUT POSITION OF EXECUTION
C A1 INPUT NAME OF THE SUBROUTINE
C A2 INPUT NAME OF THE SUBROUTINE (CONTINUED)
C N1 INPUT PRIMARY OVERLAY NO
C N2 INPUT SECONDARY OVERLAY NO
C IW INPUT FILE NO. FOR LINE PRINTER

WRITE(IW+120)N,A1,A2,N1,N2
120 FORMAT(I11//A1X,7HERROR DETECTED AT POSITION +I2+1X+
111HSUBROUTINE +PA+1X,12HLINE OVERLAY (+I1+1H++1I11)
STOP 111
END

Figure 77. Subroutine ERM Program Listing
SUBROUTINE DERRM(M1,M2,M3,M4,M51,M52,M53,M54,N1,N2,A1,A2,IM)

PURPOSE: TO PRINT ERROR MESSAGE WHEN DIMENSIONS FOR
SCRATCH ARRAYS IS NOT SUFFICIENT

ANALYST: A F KONAR / J K NAMSH - THE HONEYWELL INC

DATE WRITTEN - 1975

ARGUMENTS LIST

M1 INPUT ACTUAL DIMENSION FOR SCRATCH ARRAY S1
M2 INPUT ACTUAL DIMENSION FOR SCRATCH ARRAY S2
M3 INPUT ACTUAL DIMENSION FOR SCRATCH ARRAY S3
M4 INPUT ACTUAL DIMENSION FOR SCRATCH ARRAY S4
M51 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
M52 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
M53 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
M54 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
N1 INPUT PRIMARY OVERLAY NO
N2 INPUT SECONDARY OVERLAY NO
A1 INPUT NAME OF THE SUBROUTINE (CONTINUED)
A2 INPUT NAME OF THE SUBROUTINE
IM INPUT FILE NO FOR LINE PRINTER

DIMENSION M(4),M5(4)
M(1)=N1 M(2)=N2 M(3)=N3 M(4)=N4
M5(1)=M51 M5(2)=M52 M5(3)=M53 M5(4)=M54
WRITE(14*,240),N1,N2,A1,A2

240 FORMAT(1H1,1/1X,2AH1DIMENSION ERROR IN OVERLAY (11111111111M)
113HIN SUBROUTINE*X2X244)
241 CONTINUE
WRITE(14*,250),M5(1),M5(1)

250 FORMAT(1X,15H1DIMENSION FOR S*11,2X7ACTUAL=,15,2X,
15HREQUIRED=,15)
260 CONTINUE
STOP 111
END

Figure 78. Subroutine DERRM Program Listing
SUBROUTINE DERRMS(1,2,3,4,MS1,MS2,MS3,MS4,N1,N2,A1,A2,IN)

PURPOSE - TO PRINT ERROR MESSAGE WHEN SYSTEM DIMENSION ARE NOT SUFFICIENT

AUTHORS - A K KONAR / J K YAMSH - THF HONEYWELL INC

DATE WRITTEN - 1976

ARGUMENTS LIST

M1     INPUT ACTUAL DIMENSION
M2     INPUT ACTUAL DIMENSION
M3     INPUT ACTUAL DIMENSION
M4     INPUT ACTUAL DIMENSION
MS1    INPUT MAXIMUM DIMENSION
MS2    INPUT MAXIMUM DIMENSION
MS3    INPUT MAXIMUM DIMENSION
MS4    INPUT MAXIMUM DIMENSION
N1     INPUT PRIMARY OVERLAY NO
N2     INPUT SECONDARY OVERLAY NO
A1     INPUT NAME OF THE SUBROUTINE
A2     INPUT NAME OF THE SUBROUTINE (CONTINUED)
IN     INPUT FILE NO FOR LINE PRINTED

DIMENSION M(4);MS(4);M(4)
DATA I/4M1,4M2,4M3,4M4,4MS1,4MS2,4MS3,4MS4 /
M(1)=41 M(2)=42 M(3)=43 M(4)=44
MS(1)=MS1 MS(2)=MS2 MS(3)=MS3 MS(4)=MS4
WRITE(IW,240)N1,N2,A1,A2
240 FORMAT(1X,'ERROR IN OVERLAY',1X,'DIMENSION ERROR AT OVERLAY',1X,'IN SUBROUTINE DERRMS')
DO 26 1=1,4
WRITE(IW,250)A1,MS(1:4)
250 FORMAT(1X,'DIMENSION',1X,'ACTUAL=',15X,'DIMENSION=')
26 CONTINUE
STOP 111
END

Figure 79. Subroutine DERRMS Program Listing
SUBROUTINE SHIFT(NN, VN, DES, UNIT, NN, VN, DES, UNIT, NN, VN, IW, IPrint)
C
C PURPOSE - TO SHIFT CONTENTS OF OLD ARRAYS NN, VN, DES, UNIT
C INTO NEW ARRAYS NN, VN, DES, UNIT
C ANALYSIS - A F KONAR / J K NAEMSH - THE HONEYWELL INC
C DATE WRITTEN - 1976
C
C SUBPROGRAMS CALLED
C DEBUG
C
C ARGUMENTS LIST
C NN INPUT OLD NUMBER ARRAY
C VN INPUT OLD VARIABLE NAME ARRAY
C DES INPUT OLD DESCRIPTION ARRAY
C UNIT INPUT OLD UNIT ARRAY
C NN OUTPUT NEW NUMBER ARRAY
C VN OUTPUT NEW VARIABLE NAME ARRAY
C DES OUTPUT NEW DESCRIPTION ARRAY
C UNIT OUTPUT NEW UNIT ARRAY
C NM INPUT MAX NO OF SYSTEM VARIABLES
C IW INPUT FILE NO FOR LINE PRINTER
C IPrint INPUT PRINT CONTROL FLAG
C
C DIMENSION NN(NM), VN(NM+2), DES(NM+10), UNIT(NM+4)
C IF(IPrint.EQ.6) CALL DEBUG(1,4,SHIF,4HT,5,0,IW)
C DO 140 I=1,NM
C NN(I)=NN(I)
C DO 120 J=1,2
C VNN(I,J)=VN(I,J)
C DO 130 J=1,10
C DESN(I,J)=DES(I,J)
C DO 140 J=1,4
C UNITN(I,J)=UNIT(I,J)
C IF(IPrint.EQ.6) CALL DEBUG(2,4,SHIF,4HT,5,0,IW)
C RETURN
C END

Figure 80. Subroutine SHIFT Program Listing
SUBROUTINE TDINVR(ISOL, IDSOL, NR, NC, A, WRA, KWA, DET)

C PURPOSE - TO INVERT A NONSINGULAR MATRIX OR TO SOLVE LINEAR EQUATIONS

ARGUMENTS LIST

C ISOL TDINVR SOLUTION INDICATOR
C IDSOL = 1 IF INVERSE FOUND OR EQUATIONS SOLVED
C ID = 0 IF DETERMINANT OVERFLOW INDICATOR
C IDSOL = 1 IF DETERMINANT CALCULATION DID NOT OVERFLOW
C NR NUMBER OF ROWS
C NC NUMBER OF COLUMNS
C A MATRIX TO BE INVERTED OR CONTAINING THE LINEAR EQUATION COEFFICIENTS
C MR0 MAXIMUM NUMBER OF ROWS OF A
C KWA SCRATCH ARRAY WHEN INVERTING
C DET VALUE OF THE DETERMINANT

DIMENSION A(11), KWA(1)

C IF NR ISOL = 1 IDSOL = 1
IF(NR) 61, 61, 11
IF(Id - 1, 12, 12)
1 IC = ID - (NC)
IF((IC - IR) 1516, 15, 16)
1 IC = IR
14 IR = (IR - MSER)
GO TO 17
16 MDIV = 1
17 MADM = DIV
MSER = 1
KSER = IR
MZ = 1
DET = 1, 0
18 PIV = 0, 0
19 IMSE = 1
20 IF (IV - MSER) 20, 20, 23
21 PIV = AMS(A(11))
C
22 IP = 1
23 IF (PIV) 24, 62, 24
24 IF (NC) 26, 25, 25
25 IF (IP - 1, 1, JHMP) 27, 27, 27
JHMP = (IP - 1, JHMP) * JHMP
JMSER = (KSER - 1) / JHMP * JHMP
11 IF (IP - MSER) 20, 21, 21
22 JHMP = (IP - MSER)
KWA(J) = 1
GO TO 27
26 IP = 1
27 IF (IP - MSER) 61, 31, 28

Figure 81. Subroutine TDINVR Program Listing
Figure 81. Subroutine TDINVR Program Listing (Continued)
ISOL = ?
IDSOL = 1
GO TO 65
63  ISOL = 2
    IDSOL = 2
65  RETURN
END

Figure 81. Subroutine TDINVR Program Listing (Concluded)
Figure 82. Overlay Structure and Subroutines in KONPACT-2
OVERLAY(KON?00)
PROGRAM MAIN(MINPUT,INPUT,TAPE7=MINPUT,TAPE4=INPUT,
I0DATA,OUTPUT,TAPER=0DATA,TAPE4=OUTPUT,SCRATCH,TAPES=SCRATCH,
2FDATA,0DATA,TAPEF=0DATA,TAPES=0DATA,TAPEF=0DATA,TAPEP=0DATA)

PURPOSE - TO SET UP MAXIMUM DIMENSIONS
ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
DATE WRITTEN - 1975

SURROGAM CALLED
KON/P

LABELLED COMMON LIST
NX1  MAXIMUM NUMBER OF STATES
NY1  MAXIMUM NUMBER OF OUTPUTS
NU1  MAXIMUM NUMBER OF INPUTS
CODE PROGRAM CODE WORD (DIAG,FFOC,LSA)
MS1  MAXIMUM DIMENSION FOR SCRATCH ARRAY 51
MS2  MAXIMUM DIMENSION FOR SCRATCH ARRAY 52
MS3  MAXIMUM DIMENSION FOR SCRATCH ARRAY 53
MS4  MAXIMUM DIMENSION FOR SCRATCH ARRAY 54

COMMON /INF/ NXM,NRM,NUM,CODE,MS1,MS2,MS3,MS4

MAXIMUM SYSTEM DIMENSIONS
NXM=51 % NRM=70 % NUM=70

MAXIMUM SCRATCH ARRAY DIMENSIONS
MS1=0500 $ MS2=17000 $ MS3=000001 $ MS4=000001

*** NOTE *** SCRATCH ARRAY DIMENSIONS IN PROGRAM DATAK
SHOULD BE CHANGED

CALL KON/P
CALL <ORG2
STOP
END

Figure 83. Program MAIN Program Listing
OVERLAY(KON2+10)  DIAK 2
PROGRAM DIAK  DIAK 3
C DOUBLY-ITERATIVE ALGORITHM FOR SOLVING ALGEBRAIC RICCATI EQUATION DIAK 4
C THIS PROGRAM COMPUTES QUADRATIC CONTROLLERS AND/OR COMPUTES COVARIANCE DIAK 5
C TIME RESPONSES FOR SYSTEMS MODELED AS
C
C XDOT = F*X + G1*U + G2*ETA
C AND
C R = H*X + N*U
C WITH
C J = E(R*Q*R)
C
C DIMENSION F(40,40),G1(40,6),G2(40,2),A(40,40),AN(40,40),E(40,40) DIAK 6
C DIMENSION G(40,40),MR(40,40),EP(40,40),P(40,40),H(40,40),D(40,6) DIAK 7
C DIMENSION AK(6,40),PI(40,40),DD(6,6),W(16,40),N(16,40) DIAK 8
C DIMENSION QQ(40,40),RR(90),AM(40,40),MK(6,40),X(40),DX(40),DLX(40) DIAK 9
C DIMENSION XI(40,2),XLX(40,2),GM(40,2),GS(40,2),R(8000),IPLR(80) DIAK 10
C DIMENSION IITL(80),YMIN(80),YMAX(80),CL(12+1),SCAL(80),NEWY(80) DIAK 11
C DIMENSION NORD(480),QR(40,40),UNIT(80) DIAK 12
C COMMON A,E,Q,AN,WR,OR,EP,P,P
C EQUIVALENCE (F(I),P(N),A(I),PI(I))- DIAK 13
C EQUIVALENCE (P(I),E(N))
C DIMENSIONS OF THE ABOVE ARRAYS ARE DEFINED BELOW, CHANGE BOTH SIMULTANEOUSLY
C SEE DOCUMENTATION FOR DEFINITIONS OF ARRAY DIMENSIONS
C
C M>40
C MX>>MX
C MR>>MR
C MU>>MU
C MXR=MR
C NPO=NX+1
C CONVERGENCE TEST FACTOR
C EE=0.01
C ITERATION COUNTER
C IRUN=0
C READ AND PRINT ID
C READ(*,1274) DATE,NAMES1,NAMES2
C 1274 FORMAT(3A10) DIAK 16
C WRITE(*,1275) DATE,NAMES1,NAMES2
C 1275 FORMAT(1H1,TX,13H**DATE** DATE =A10.5X,16MIDENTIFICATION +2A10//) DIAK 17
C READ NUMBER OF VARIABLES BEING PLOTTED
C READ(*,20) NOP
C 20 FORMAT(4B2) DIAK 18
C IF NOP = 0; SKIP TO STATEMENT 70
C IF(NOP,EQ,0) GO TO 70 DIAK 19
C IF NOT PRINTED, PRINT WHETHER FIXED ALONG WITH RUNS
C IF TYPICAL, PRINT FOR ALL RUNS
C IF NOT PRINTED, PRINT WHETHER FIXED
C IF SCAL = 0, USE COMPUTED MAX AND MIN (NEWY=1)
C IF SCAL = 0, USE SCALE FACTOR OF 1
C
C

Figure 84. Program DIAK Program Listing
DO 1 I=1,NOP
NEWY(I)=0
IF(YMIN(I)*EQ.0.0. AND YMAX(I)*EQ.0.0.) NEWY(I)=1
IF(SCAL(I)*EQ.0.0.) SCAL(I)=1.
1 CONTINUE
C READ AND PRINT PLOTTING TIME PARAMETERS - FIXED FOR ALL RUNS
C T = TOTAL PLOTTING TIME
C DT = SAMPLING INTERVAL
C ST = PLOTTING SAMPLING INTERVAL
C T1 = FIRST DELAY IN GUST PROFILE
C T2 = SECOND DELAY IN GUST PROFILE
READ(4,1278) T,DT,ST,T1,T2
1278 FORMAT(5G12.4)
C PRINT PLOTTING PARAMETERS
WRITE(4,1279) T,DT,ST,T1,T2
1279 FORMAT(1H0/7X,3H TIME RESPONSES PLOTTING TIME = G12.4/22X+18H SAMPLIK
C PLOTTING INTERVAL = G12.4/22X+27H PLOTTING SAMPLE INTERVAL = G12.4/
22X+19H FIRST DELAY TIME = G12.4/22X+20H SECOND DELAY TIME = G12.4/
3/*)
WRITE(4,1280) ((LPLR(I),ITITL(I),UNIT(I),YMIN(I),YMAX(I),SCAL(I))IDIAK
1),ID=1,NOP)
1280 FORMAT(17X+8HRESPONSE NUMBER RESPONSE VALUES
1VARIABLE RESPONSE UNITS MIN SCALE MAX SCALE SCALE FACTOR//12X, IDAK
2110X,A10,9X,A10,3X,G11,3X,G11,3X,G11,3X,G11,3X,G11))
C READ AND PRINT MAX NO. OF INNER AND OUTER LOOP ITERATIONS
READ(5,28) IMAX,ITER0
WRITE(9,882) IMAX,ITER0
4002 FORMAT(///7X,3H MAX NUMBER OF INNER-LOOP ITERATIONS I3,3H MAX NO
NUMBER OF OUTER-LOOP ITERATIONS I3,7X,6H MAX NUMBER OF ITERATIONS IDAK
20H ELIMINATING CONTROL SURFACE FEEDBACKS I3//)
C DEFINITION OF PROGRAM OPTIONS
C INPD=1 COMPLETELY NEW DATA
C INPD=2 CHANGE SELECTED QUADRATIC WEIGHTS ONLY - USE SOME GAINS IN SIDIAK
C INPD=3 CHANGE SELECTED QUADRATIC WEIGHTS ONLY WITH OPTION FOR NEW GADIK
C INPD=4 CHANGE SELECTED DATA IDAK
C INPD=5 CHANGE SELECTED DATA IN MEASUREMENT MATRIX+ QUADRATIC WEIGHTSIDIAK
C INPD=6 OPTION FOR NEW GAINS IDAK
C INPK=1 NEW INPUT GAINS IDAK
C INPK=2 NEW STARTING ROUTINE GAINS IDAK
C INPK=3 USE GAINS IN STORAGE IDAK
C INPK=4 USE INPUT GAINS IN STORAGE IDAK
C INCONT=0 DON'T COMPUTE OPTIMAL GAINS - USE INPUT GAINS AND DATA IN CORR
C AND TIME RESPONSE ANALYSIS ONLY IDAK
C INCONT=1 COMPUTE OPTIMAL GAINS IDAK
C INCONT=2 DO AUTOMATIC SELECTION OF Q ON CONTROL RATES IDAK
C INCONT=3 DO AUTOMATIC SELECTION FOR PLOTTING OPTIONS USING NPLOT, NPRIN
C NSTEP, IDAK
C NOCOV=1 NO COVARIANCE ANALYSIS IDAK
C NOCOV=2 COVARIANCE ANALYSIS IDAK
C NOCOV=3 SKIP CORRELATION ANALYSIS IDAK
C READ AND PRINT PROGRAM OPTIONS
READ(5,28) NOCOV,NSTEP,NRAND,NPRIN,NPLOT
READ(5,28) INPK
INPD=1
READ(5,28) NCONT
WRITE(9,37) INPD,INPK,NCONT,NOCOV,NSTEP,NRAND,NPRIN,NPLOT
37 FORMAT(1H0/7X,7H NEW PROBLEM WITH INPD =I3,2X,6H INPK =I3,2X,
2H NCONT =I3,7H NOCOV =I3,7H NSTEP =I3,7H NRAND =I3,7H,
7H NPRIN =I3,7H NPLOT =I7//)
1210 CONTINUE
C READ FLIGHT CONDITION ID
READ(5,1278) IFLT
1278 FORMAT(A10)
C PRINT FLIGHT CONDITION ID AND RUN NO.
WRITE(9,1271) IFLT, IRUN
1271 FORMAT(1H0/7X,18H FLIGHT CONDITION A10+5X+3HRUN+13)
C READ ANY PRINT SYSTEM PARAMETERS

Figure 84. Program Dlak Program Listing (Continued)
Figure 84. Program DIAK Program Listing (Continued)
CALL INPT(F,MX,MX)
CALL INPT(G1,MX,MU)
CALL INPT(G2,MX,NN)
IF(INPD GT 1) GO TO 54
C IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)
CALL SHUFL(F,MX,MX,NX,NX,1,1,NORD,0,MX)
CALL SHUFL(G1,MX,MN,NX,NU,1,0,NORD,0,MX)
CALL SHUFL(G2,MX,NN,NX,NN,1,0,NORD,0,MX)
54 CONTINUE
C PRINT F, G1, G2
WRITE(9,20)
CALL HP(MX,MX,NX,NX,F)
WRITE(9,21)
CALL HP(MX,MU,NX,NN,G1)
WRITE(9,22)
22 FORMAT(1H1/7X,10H MATRIX//)
CALL HP(MX,NN,NX,NN,G2)
20 FORMAT(1H1/7X,10H MATRIX//)
21 FORMAT(1H1/7X,10H MATRIX//)
C READ CHANGES IN XI AND XLDAL
C XI = INITIAL STATE VALUES IN SIMULATION
C XLDAL = STATE AND STATE RATE LIMITS
CALL INPT(XI,NX,MX)
CALL INPT(XLDAL,MX,2)
IF(INPD GT 1) GO TO 55
C IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)
CALL SHUFL(XI,MX,MN)
CALL SHUFL(XLDAL,MX,2,XI)
CONTINUE
C IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)
CALL SHUFL(H,MX,MX,MR,NX,MR,PN,MR,0)
C READ CHANGES IN M (AM = MEASUREMENT MATRIX - USED FOR RESPONSE ANALYSIS ONLY
WHERE
C AM = MEASUREMENT MATRIX - USED FOR RESPONSE ANALYSIS ONLY
WHERE
C READ IN CHANGES IN M AND D
AM = STATE-RESPONSE OUTPUT MATRIX
D = CONTROL-RESPONSE OUTPUT MATRIX
R = H*M + D*U
3 AM = MEASUREMENT MATRIX - USED FOR RESPONSE ANALYSIS ONLY
WHERE
Y = M*U
C 73 CALL INPT(H,MR,MX)
CALL INPT(D,MR,MU)
IF(INPD GT 1) GO TO 1250
C IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)
CALL SHUFL(H,MR,MR,NX,NX,0,1,NORD,0,MX)
C READ CHANGES IN M (AM)
3 AM = MEASUREMENT MATRIX - USED FOR RESPONSE ANALYSIS ONLY
WHERE
Y = M*U
C 1250 CALL INPT(AM,MX,MX)
IF(INPD GT 1) GO TO 56
C IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)
CALL SHUFL(AM,MX,MX,NX,NX,0,1,NORD,0,MX)
56 CONTINUE
C PRINT M, D, AM
WRITE(9,23)
CALL HP(MR,MX,MR,NX,AM)
Figure 84. Program DIAK Program Listing (Continued)
WRITE(9,24)
CALL MP(MR,NR,NU,NF,MU)
WRITE(9,29)
29 FORMAT(1H1/7X,1OH M MATRIX///)
CALL MP(MX,NX,NM,NAM)
23 FORMAT(1H1/7X,1OH M MATRIX///)
24 FORMAT(1H1/7X,1OH N MATRIX///)
C CHECK GAINS INPUT OPTION
1230 GO TO (3001,3002,3003,3004,3005,3006,3007,3008,3009)
C NEW INPUT GAINS
C R = INPUT GAINS MATRIX
C WHERE
U = BK*X (WHEN COMPUTING OPTIMAL GAINS)
C OR
U = BK*Y = RK*AM*X (WHEN COMPUTING RESPONSES ONLY)
C ZERO AND READ BK
3000 DO 3001 J=1,NU
   DO 3003 I=1,NF
      ! IF NCONT<0 RE-ORDER STATES (BECAUSE U = BK*X)
      CONTINUE
      ! PRINT BK
      WRITE(9,31)
      31 FORMAT(1H1/7X,3IH STARTING GAINS MATRIX///)
      CALL MP(MU,NX,NU,NF,BK)
      SKIP TO STATEMENT 1230 TO READ QUADRATIC WEIGHTS
      GO TO 1220
   C USE STARTING ROUTINE (STRIC) TO COMPUTE STARTING GAINS - AS A LAST RESORT
      BK = -G1*(W(T)) (AI MEANS INVERSE OF MATRIX A)
      C WHERE
      W(T) = INTEGRAL(0,T)OF(EXP(F*T)+G1*G1*EXP(F*T))DT
      C FOR AN ARBITRARY TIME AT
      3002 CALL STRIC(F,G1,A+AN+AP+NP+NP,NF,NU+NX,NM)
      CALL TDINVR(ISOL,1DSOL+NF+NF+AN,1NX+NX+XM,1NX+NX+XM+DET)
      IF(ISOL-1DSOL<2) GO TO 3004
      3003 CALL TDINVR(ISOL,1DSOL+NF+NF+AN,1NX+NX+XM,1NX+NX+XM+DET)
      IF(ISOL-1DSOL<2) GO TO 3004
      C IF W(T) IS NO GOOD - GO TO NEXT RUN - BUT FIRST READ REMAINING DATA
      C THIS RUN AND CHECK TO SEE IF THE NEXT RUN IS SOLVABLE - THE STARTING GAINS MAY NOT BE GOOD - IF SO STOP
      3005 WRITE(9,3006)
      3006 FORMAT(1H1/7X,3IH INVERSE OF W(T) DOES NOT EXIST/TX=10H CHECK NEXT)
      I PROBLEM///
      I PROBLEM///
      CALL MP(QD,MR,MR)
      READ(5,1215) IDUM
      IF(IDUM) STOP 77
      READ(5,20) INPD,INPK
      IF(INPK,EQ.1) GO TO 1216
      IF(INPK,EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.2)) GO TO 1216
      WRITE(9,3008)
      3008 FORMAT(1H1/7X,3IH NEW PROBLEM NOT SOLVABLE WITHOUT NEW STARTING GAINS)
      I PROBLEM///
      STOP 11
      C DEFINE BK
      C AN = (NCT)**1
      C
      3004 DO 3009 I=1,NU
         DO 3009 J=1,NF
            BK(I,J)=0.
         DO 3009 K=1,NF
            BK(I,J)=BK(I,J)-G1(KI)*AN(K,J)
         C PRINT BK
         WRITE(9,31)
         31 FORMAT(1H1/7X,22H STARTING GAINS MATRIX///)
      STOP 11
CALL MP(MU,NK,NMR,8K)
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
GO TO 1220
C USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS
C DEFINE RK = AK
3010 WRITE(9,33)
33 FORMAT(1H1/7X,28H USE GAINS MATRIX IN STORAGE//)
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
GO TO 1220
C USE INPUT GAINS IN STORAGE = AK = BK
3011 WRITE(9,34)
34 FORMAT(1H1/7X,24H USE INPUT GAINS MATRIX IN STORAGE//)
C READ CHANGES IN QUADRATIC WEIGHTS FOR PERFORMANCE INDEX
C WHERE: 0 IS THE MATRIX OF QUADRATIC WEIGHTS
C 00 = 0
1220 CONTINUE
CALL INPT(00,MR,9K)
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
GO TO 1220
C USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS
C USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS
C 3010 WRITE(9,33)
33 FORMAT(1H1/7X,28H USE GAINS MATRIX IN STORAGE//)
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
GO TO 1220
C USE INPUT GAINS IN STORAGE = BK
C 3011 WRITE(9,34)
34 FORMAT(1H1/7X,24H USE INPUT GAINS MATRIX IN STORAGE//)
C READ CHANGES IN QUADRATIC WEIGHTS FOR PERFORMANCE INDEX
C WHERE: 0 IS THE MATRIX OF QUADRATIC WEIGHTS
C 00 = 0
1220 CONTINUE
CALL INPT(00,MR,9K)
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
GO TO 1220
C USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS
C USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS
C 3010 WRITE(9,33)
33 FORMAT(1H1/7X,28H USE GAINS MATRIX IN STORAGE//)
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
GO TO 1220
C USE INPUT GAINS IN STORAGE = BK
C 3011 WRITE(9,34)
34 FORMAT(1H1/7X,24H USE INPUT GAINS MATRIX IN STORAGE//)
C READ CHANGES IN QUADRATIC WEIGHTS FOR PERFORMANCE INDEX
C WHERE: 0 IS THE MATRIX OF QUADRATIC WEIGHTS
C 00 = 0
1220 CONTINUE
CALL INPT(00,MR,9K)
C PRINT 00
WRITE(9,36)
36 FORMAT(1H1/7X,28H QUADRATIC WEIGHTING MATRIX//)
36 FORMAT(1H1/7X,28H QUADRATIC WEIGHTING MATRIX//)
C IF NCONT = 0 (NO OPTIMAL CONTROL COMPUTATIONS), SKIP TO STATEMENT 800
C RESPONSE COMPUTATIONS
IF(NCONT,EO.O) GO TO 903
C CALCULATE A+E+0 FOR PICCATTI EQUATION 0 = PA + A&P + Q - PEP
C W = D#0
DO 4 I=1,NU
DO 4 J=1,NR
W(I,J)=0.
DO 4 K=1,NR
W(I,J)=W(I,J)+D(K)*Q0(K,J)
C DOD = D#0
DO 5 I=1,NU
DO 5 J=1,NR
DOD(I,J)=0.
DO 5 K=1,NR
DOD(I,J)=DOD(I,J)+W(I,K)*Q0(K,J)
C INVERT MOD = DOD = (n#n#0)
IF(NU-1302+302,301)
302 DOO(I,J)=1./DOD(I,J)
GOTO 303
301 CONTINUE
CALL TINVN(1501,1501,NU,NU,DOD,NU+MK+4,DET)
C DO#Q#D) DOES NOT EXIST - GO TO NEXT RUN
7 WRITE(9,35)
35 FORMAT(1H1/7X,30H INVERSE OF DOD DOES NOT EXIST//7X,19H CHECK NEXT DOD)
1 PROBLEM//)
GO TO 1220
5 CONTINUE
303 CONTINUE
C W = D#Q#H
DO 8 I=1,NU
DO 8 J=1,NX
W(I,J)=0.
DO 8 K=1,NR
C W = (D#Q#D)+D#Q#H
DOD(I,J)=DOD(I,J)+W(I,J)
C STORE W FOR OPTIMAL CONTROL COMPUTATION
DO 9 I=1,NU
DO 9 J=1,NX
W(I,J)=0.
Figure 84. Program DIAK Program Listing (Continued)
DO 9 I=1,NX  
9 W(I,J)=W(I,J)+QDO(I,K)*W(K,J)  
C AN = A OF EQUATION $ O = A*P + PA + Q - PEP  
DO 10 I=1,NX  
10 AN(I,J)=F(I,J)  
DO 10 K=1,NJ  
10 AN(I,J)=AN(I,J)+G(I,K)*W(K,J)  
C 0 = M*Q*O*D*(O*Q*O*D):*D*Q*H  
DO 12 I=1,NX  
12 J=1,NX  
12 G(I,J)=0  
DO 12 K=1,NJ  
12 G(I,J)=G(I,J)-V(K,I)*W(K,J)  
C E = O*H  
DO 13 I=1,NR  
13 J=1,NX  
E(I,J)=0  
DO 13 K=1,NR  
13 E(I,J)=E(I,J)+D0(I,K)*H(K,J)  
C 0 = N OF EQUATION $ O = A*P + PA + O - PEP  
DO 14 I=1,NX  
14 J=1,NX  
14 K=1,NR  
15 O(I,J)=Q(I,J)+(K,I)*F(K,J)  
14 Q(I,J)=Q(I,J)  
C W(I,J)=D*Q*O*D):*G*#  
DO 16 I=1,NU  
16 J=1,NX  
W(I,J)=0  
DO 16 K=1,NU  
16 W(I,J)=W(I,J)+QDO(I,K)*G(J,K)  
C E = E OF EQUATION $ O = A*P + PA + Q - PEP  
DO 17 I=1,NX  
17 J=1,NX  
E(I,J)=0  
DO 17 K=1,NR  
17 E(I,J)=E(I,J)+G(I,K)*W(K,J)  
C PRINT AN=E*Q  
WRITE(9,32)  
32 FORMAT(I1,'7X+36HSTARTING MATRICES FOR PA*A*P+Q-PEP=Q//')  
WRITE(9,25)  
25 CALL'H(MX+NX+NX+NX+NX+AN)  
WRITE(9,26)  
26 CALL'H(MX+NX+NX+NX+AN)  
WRITE(9,27)  
27 CALL'H(MX+NX+NX+NX+Q)  
25 FORMAT('/7X+10H A MATRIX//')  
26 FORMAT('/7X+10H E MATRIX//')  
27 FORMAT('/7X+10H O MATRIX//')  
C DUMP F+, AND AM ON DISC TO CONserve STORAGE  
C MP, PI, AND EQ USE STORAGE EQUIVALENT TO THESE MATRICES  
REWIN=2  
WRITE(2) F  
WRITE(2) H+AM  
ITERC=0  
C CHECKS INPUT OPTION  
GO TO (3800,3800,P50,3000),INPK  
C FOR ALL OPTIONS EXCEPT INPK = 3 (USE AM IN STORAGE), AK = BK  
3800 DO 7010 I=1,NU  
7010 AN(I,J)=AM(I,J)  
3800 DO 7010 J=1,NU  
7010 AN(I,J)=AM(I,J)  
3800 DO 7010 J=1,NU  
7010 AN(I,J)=AM(I,J)  
7010 AN(I,J)=AM(I,J)  
Figure 84. Program DIAK Program Listing (Continued)
DO 7011 I=1,NX
DO 7011 J=1,NX
C A(I,J)=F(I,J)
DO 7011 K=1,NU
7011 A(I,J)=A(I,J)+G(I,K)*AK(K,J)
C H=H+D*K
C H IS NOW CLOSED LOOP STATE-RESPONSE OUTPUT MATRIX
DO 7012 I=1,NR
DO 7012 J=1,NX
DO 7012 K=1,NU
7012 H(I,J)=H(I,J)+H(I,K)*AK(K,J)
C COMPUTE (H+D*K)#Q*(H+D*K)
DO 7013 I=1,NR
DO 7013 J=1,NX
DO 7013 K=1,NU
P(I,J)=P(I,J)+Q(I,K)*H(K,J)
C EP = (H+D*K) Commentary
DO 7014 I=1,NX
DO 7014 J=1,NX
DO 7014 K=1,NR
C SOLVE FOR INITIAL RICCATI MATRIX P FROM
C 0 = A*EP + P*A + Q + (H+D*K) Commentary
C VIA SUBROUTINE CAL
3 P IS WORKING MATRIX HERE = RICCATI MATRIX RETURNS IN EP
CALL CAL(A,EP,P,KMA=NX,MA=IMAX+1,IER=EE)
IF(IER<0) GO TO 875
C ERROR ENCOUNTERED IN CAL = GO TO NEXT RUN
WRITE(9,'(E13.8)')
38 FORMAT(1H1/77x,27H INITIAL GAINS ARE UNSTABLE//77x,19H CHECK NEXT PROJ)
100 FORMAT(I2)
READ(1215) IDUM
IF(1044,GT,0) STOP 77
READ(5,28) INPD,INPK
IF(INPK.EQ.1) GO TO 1216
IF(INPK.EQ.2.AND.(INPD.EQ.1).OR.INPD.EQ.2) GO TO 1216
C NEXT RUN NOT SOLVABLE WITH PRESENT STARTING GAINS = STOP
WRITE(9,3008)
STOP 1
C SET P = EP; INITIALIZE PI = 0
875 DO 876 I=1,NX
DO 876 J=1,NX
PI(I,J)=0.
876 P(I,J)=EP(I,J)
C UPDATE A AND G MATRICES FOR NEXT IterATION
C A = AN - EP
C Q = 0 + P*EP
C TO SOLVE FOR P FROM
C 0 = A*EP + P*A + 0
C VIA SUBROUTINE CAL
C AFTER SOLVING FOR SECOND P, SOLVE FOR DIFFERENCES IN P BETWEEN ITERATIONAL
C THIS INITIALIZE DIFFERENCES AND CONVERGENCE CRITERIA
C PI IS DIFFERENCE AND PT IS THE TOTAL RICCATI MATRIX
C INITIALLY PI IS ZERO
DO 106 I=1,NX
DO 106 J=1,NX
EP(I,J)=0.
Figure 84. Program DIAK Program Listing (Continued)
DO 101 I=1,NX
101 F(I,J)=F(I,J)+F(I,K)*P(K,J)
DO 102 J=1,NX
DO 103 K=1,NX
103 (1,J)=G(I,J)+P(I,K)*F(K,J)
102 G(I,J)=F(I,J)
100 CONTINUE
DO 201 I=1,NX
DO 202 J=1,NX
201 P(I,J)*P(I,J)*P(I,J)
CALL 'SECOND(T)' 
WRITE(4,355) IT
3055 FORMAT('/%X,S/'),IT=INT(5/7)
C CALL CALP AGAIN WORKING MATRIX -- PICCATI MATRIX RETURNS IN Q
CALL CALA(J,P,K,NX,NA,NA,1,NA,TERR,EE)
FEE=EE*10.
CALL 'SECOND(T)
WRITE(4,355) IT
IF(EE.EQ.0.) GO TO 874
C ERROR -- COUNTERFER IN CAL -- GO TO NEXT RUN
WRITE(4,349)
39 FORMAT('/%X,S/'),IT=INT(5/7)
C PICCATI SOLUTION IS DIVERGING/%X,S/IT=INT(5/7) CHECK NEXT
IF(IT.P.EQ.0.) STOP 77
IF(IT.EQ.28) INPJ,INDK
IF(INPJ,INDK).EQ.1216
GO TO 1216
IF(INPJ,INDK).EQ.1216
GO TO 1216
C NEXT RUN NOT SOLVABLE WITH PRESENT STARTING GAINS = 50 STOP
WRITE(4,309A)
STOP 11
C SET P = 0
874 DO 877 I=1,NX
875 P(I,J)=0.0
877 DO 874 J=1,NX
876 P(I,J)=0.0
C ON SECOND ITERATION -- SOLVE FOR DIFFERENCE P = P - P
DO 3054 J=1,NX
3054 P(I,J)*P(I,J)+P(I,J)
3057 CONTINUE
ITERC=ITERC+1
C UPDATE A AND Q FOR NEXT ITERATION WHERE
A = A + E(P,P) - (P*P) IS TOTAL PICCATI MATRIX
Q = D*P**P
C TO SOLVE FOR THE DIFFERENCE P FROM
C D*P**P = A + E(P,P) - (P*P)
DO 3050 I=1,NX
3050 A(I,J)*A(I,J)+E(I,K)*P(K,J)
DO 3051 K=1,NX
3051 (I,J)=P(I,K)*F(K,J)
3052 (I,J)=0.0
C BEFORE GOING TO THE NEXT ITERATION, CHECK FOR CONVERGENCE

Figure 94. Program DIAK Program Listing (Continued)
CONVERGENCE IS WHEN THE ABSOLUTE CHANGE IN THE ELEMENTS OF THE DIAK 593
MATRIX BETWEEN ITERATIONS IS LESS THAN THE ABSOLUTE VALUE OF THE DIAK 594
TIME EE
C ONLY CHECK THE UPPER TRIANGULAR ELEMENTS
C
DO 105 ICT=0
DO 405 J(ICT, ICT)
IF(A(ICT,J(ICT)) < 1.E-10) GO TO 135
C IF THE ELEMENTS ARE SMALL, CONSIDER THEM AS ZERO AND COUNT THEM AS DIAK 602
WRITE(9,391)
READ(5,1215) ICT, J
IF(IDM, GT, 0) STOP 77
READ(4, 28) INPD, INDP
IF(INDP, EQ, 1) GO TO 136
IF(INPD, EQ, 1, 0, INPD, EQ, 2) GO TO 136
C IF THE ELEMENTS ARE LARGE, CONSIDER THEM AS DIVERGING AND GO TO NEXT DIAK 404
WRITE(9,3008)
STOP 11
888 IF(A(ICT,J(ICT)) < 1.E-10) GO TO 105
106 RAT=A(ICT,J(ICT))
RAT=A(ICT,J(ICT))
IF(RAT, EQ, 1.E-10) GO TO 107
C COUNT CONVERGED ELEMENTS
105 ICT=ICT+1
107 CONTINUE
C IF ICT DOES NOT EQUAL NC, THE NUMBER OF ELEMENs, AND THE NUMBER OF DIAK 622
TIONS DOES NOT EQUAL ITER, GO TO NEXT ITERATION
108 IF(ITS-ICT, GT, 109) STOP 109
109 IF(IT(ICT, EQ, 100)) STOP 109
C IF ITER EQUALS ITER, NO CONVERGENCE - PRINT LAST TWO RICCATi MATRICES
C AND GO TO NEXT RUN
1001 WRITE(9,1211) ITER, ICT
120 FORMAT(1H1/7X, 1AH NOT CONVERGED IN 17, 1AH ITERATIONS-FIRST TERM TO DIAK 629
FAIL =16/)
121 FORMAT(1H1/7X, 1AH NOT CONVERGED IN 17, 1AH ITERATIONS-FIRST TERM TO DIAK 629
FAIL =16/)
121 FORMAT(1H1/7X, 1AH NOT CONVERGED IN 17, 1AH ITERATIONS-FIRST TERM TO DIAK 629
FAIL =16/)
WRITE(9, 1211) ITER, ICT
WRITE(9, 1211) ITER, ICT
WRITE(9, 1211) ITER, ICT
3054 P(I,J)=P(I,J) + P(I,J)
WRITE(9,1211) ITER
WRITE(9,1211) ITER
WRITE(9,1211) ITER
121 FORMAT(3P4H MATRIX AT ITERATION 13/)
CALL *PMX, MX, INPD, INDP
CALL *PMX, MX, INPD, INDP
CALL *PMX, MX, INPD, INDP
WRITE(9,39)
WRITE(9,39)
WRITE(9,39)
C MODIFICATIONS
C CR READ(5,1215) IDM
C CR IF(IDM,GT,0) STOP 77
C CR READ(4,28) INPD, INDP
C CR IF(INPD, EQ, 1) GO TO 1716
C CR IF(INPD, EQ, 1, 0, INPD, EQ, 2) GO TO 1716
C NEXT RUN IS NOT SOLVABLE WITH PRESENT STARTING GAINS - STOP
C CR WRITE(9,3008)
C CR STOP 11
C CR MODIFICATIONS
122 CONTINUE
C COMPUTE OPTIMAL GAINS
DO 120 I=1, NX
DO 120 J=1, NX
3056 P(I,J)=P(I,J)+P(I,J)
DO 120 I=1, NX
DO 120 J=1, NX

Figure 84. Program DIAK Program Listing (Continued)
AK(I,J) = -w(I,J)
DO 125 K = 1, NX
125 AK(I,J) = AK(I,J) + W(I,J) * P(K,J)
C SET COMMAND FEEDFORWARD GAINS TO ZERO
NXMNC = NX - NCS + 1
DO 86 I = 1, NU
J = I * I = F - NU
DO 86 J = NXMNC + 1
C *** MODIFICATIONS
C A(I,I) = 0.
C *** MODIFICATIONS
86 AK(I,J) = 0.
C *** MODIFICATIONS
RECOMPUTE A - CLOSED LOOP STABILITY MATRIX
REWIN 2
READ(2) A
DO 88 I = 1, NX
DO 88 J = 1, NX
DO 88 K = 1, NU
88 A(I,J) = A(I,J) + G(I,K) * AK(K,J)
C *** MODIFICATIONS
C PRINT GAINS MATRIX AND PCCATI MATRIX
4004 FORMAT(1H1/TX=1H) MATIX/1)
4010 WRITE(9,4005)
4005 FORMAT(1H1/TX=1H) PCCATI MATIX/1)
CALL M P (MX, MX, NU, NX, AK)
WRITE(9, 4004)
CALL HP (NU, MX, NX, NU, AK)
C RE-READ M AND N MATRICES FROM DISC
REWIN 2
READ(2) M+4M
IF (NCUR.LT.2) GO TO 82
C RECOMPUTE QUADRATICS WEIGHTS ON CONTROL RATES
NSCSS = NF + NU + 1
DO 80 I = NSCSS, NF
J = I - JCSS + NF
J = J + JCSS + 1
DO 80 R = JNSCSS, NF
DD = (PI(I,J) * R(I,J) / (H(I,J) * D(I,J)))
REWIN 2
READ(2) F
NU = N + NNU
NCU = 0
DO 84 I = 1, NU
DO 84 J = NSCSS, NF
IF (ABST (AK(I,J) GT .05) GO TO 84
NCU = NCU + 1
84 CONTINUE
IF (NCU.EQ. NNU) GO TO 85
IF (NU .GT. ITERO) GO TO 85
NO = NO + 1
INPK = 1
INPD = 2
GO TO 81
85 CONTINUE
WRITE (6, 83) IRUN
83 FORMAT (17H0 MATIX FOR CASE, I3)
CALL OUTP(NR, NR, NR, 00, 46)
82 CONTINUE
WRITE (6, 7776)
7776 FORMAT (20(4M 1))
C PUNCH IDENTIFICATION
WRITE (6, 9010) IRUN
9010 FORMAT (20GAINS MATIX FOR CASE, I3)

Figure 84. Program DIAK Program Listing (Continued)
C PUNCH OPTIMAL GAINS
CALL OUTP(MU,NX+NU,NX,AK+S)
WRITE(6,7776) DIAK 725
DIAK 726
DIAK 727
C INVERT X MATRIX (IN P) FOR COMPIUTION OF KSTAR = K*M
NO 992 I=1,NX
NO 993 J=1,NX
992 P(I,J)=AM(I,J)
CALL TDINV((ISOL+INSOL,NX,NX,P,MX,KW,DET)
IF((ISOL+INSOL)-2) AA4,AA9,AA6 DIAK 728
DIAK 729
DIAK 730
DIAK 731
DIAK 732
DIAK 733
C IF X MATRIX DOESN'T INVERT, FORGET COMPUTATION OF KSTAR - SKIP TO
C RESPONSE CALCULATIONS (STATEMENT 894)
A90 WRITE(9,1) DIAK 734
DIAK 735
DIAK 736
DIAK 737
40 FORMAT(1H/7X,3H = MATRIX INVERSE DOES NOT EXIST/7X,10H IGNORE)
GO TO 494 DIAK 738
DIAK 739
DIAK 740
DIAK 741
DIAK 742
DIAK 743
DIAK 744
DIAK 745
DIAK 746
DIAK 747
DIAK 748
DIAK 749
DIAK 750
DIAK 751
DIAK 752
DIAK 753
DIAK 754
DIAK 755
DIAK 756
DIAK 757
DIAK 758
DIAK 759
DIAK 760
DIAK 761
DIAK 762
DIAK 763
DIAK 764
DIAK 765
DIAK 766
DIAK 767
DIAK 768
DIAK 769
DIAK 770
DIAK 771
DIAK 772
DIAK 773
DIAK 774
DIAK 775
DIAK 776
DIAK 777
DIAK 778
DIAK 779
DIAK 780
DIAK 781
DIAK 782
DIAK 783
DIAK 784
DIAK 785
DIAK 786
DIAK 787
DIAK 788
DIAK 789
DIAK 790
C COMPUTE KSTAR (IN AN)
A90 NO 120 I=1,NU
NO 1210 J=1,NX
AN(I,1)=0.
NO 1220 K=1,NX
1280 AN(I,J)+AK(I,J)*AK(I,K)*P(K,J)
C STORE K<STAR IN W1
NO 58 I=1,NU
NO 58 J=1,NX
58 W(I,J)=AN(I,J)
C PRINT AND PUNCH KST</STAR
WRITE(9,1291) DIAK 791
1291 FORMAT(1H/7X,13H = KSTAR MATRIX/) DIAK 792
CALL DP(MX,NU,NX,NU,AN) DIAK 793
WRITE(6,7776) DIAK 794
DIAK 795
DIAK 796
DIAK 797
DIAK 798
DIAK 799
DIAK 800
DIAK 801
DIAK 802
DIAK 803
DIAK 804
DIAK 805
DIAK 806
DIAK 807
DIAK 808
DIAK 809
DIAK 810
DIAK 811
DIAK 812
DIAK 813
DIAK 814
DIAK 815
DIAK 816
DIAK 817
DIAK 818
DIAK 819
DIAK 820
DIAK 821
DIAK 822
DIAK 823
DIAK 824
DIAK 825
DIAK 826
DIAK 827
DIAK 828
DIAK 829
DIAK 830
DIAK 831
DIAK 832
DIAK 833
DIAK 834
DIAK 835
DIAK 836
DIAK 837
DIAK 838
DIAK 839
DIAK 840
DIAK 841
DIAK 842
DIAK 843
DIAK 844
DIAK 845
DIAK 846
DIAK 847
DIAK 848
DIAK 849
DIAK 850
DIAK 851
DIAK 852
DIAK 853
DIAK 854
DIAK 855
DIAK 856
DIAK 857
DIAK 858
DIAK 859
DIAK 860
DIAK 861
DIAK 862
DIAK 863
DIAK 864
DIAK 865
DIAK 866
DIAK 867
DIAK 868
DIAK 869
DIAK 870
DIAK 871
DIAK 872
DIAK 873
DIAK 874
DIAK 875
DIAK 876
DIAK 877
DIAK 878
DIAK 879
DIAK 880
DIAK 881
DIAK 882
DIAK 883
DIAK 884
DIAK 885
DIAK 886
DIAK 887
DIAK 888
DIAK 889
DIAK 890
C GO TO RESPONSE CALCULATIONS
GO TO 94 DIAK 891
GO TO 94
893 NO 894 I=1,NU
NO 894 J=1,NX
W(I,J)=HK(I,J)
AK(I,J)=0.
NO 896 K=1,NX
898 AK(I,J)=AK(I,J)+BK(I,K)*AM(K,J)
NO 899 I=1,NU
NO 899 J=1,NX
A(I,J)=AF(I,J)
NO 899 K=1,NX
895 A(I,J)=A(I,J)+AM(I,K)*AK(K,J)
WRITE(9,42)
42 FORMAT(1H/7X,41H = AIRCRAFT RESPONSES WITH PRESCRIBED GAINS//)
942 NO 4042 I=1,NU
NO 4042 J=1,NX
NO 4042 K=1,NU
4052 H(I,J)=H(I,J)+C(I,K)*AK(K,J)
GO TO (H50,R51,R51),NOCOV
851 CONTINUE
NO 600 I=1,NU
NO 600 J=1,NU
NO 600 K=1,NU
6080 Q(I,J)=0.
AJQ=0.
KCOM=0.
6076 KCOM=KCOM+1
WRITE(9,441)
41 FORMAT(1H/7X,36H = COVARIANCE ANALYSIS FOR DISTURRANCE,13//)
DO 4020 I=1,NX
DO 4020 J=1,NX
4020 F(I,J)=G(I,J)*KCOM*KCOM*A(I,J,KCOM)
DO 4020 K=1,NX
Figure 84. Program DIAK Program Listing (Continued)
Figure 84. Program DIAK Program Listing (Continued)
DO 111 J=1,NX
P(I,J)=0.
P(I,J)=AT(I,J)*AM(I,J)
111 CONTINUE
WRITE(*,37)
47 FORMAT(1H1/7X,*1H CONTROL-STATE HOW-SUM CORRELATION MATRIX//)
CALL "P(MX,MX,NH,NX,P)
DO 1114 I=1,NR
DO 1114 J=1,NX
P(I,J)=0.
P(I,J)=AM(I,J)*P(I,J)
1114 CONTINUE
WRITE(*,48)
48 FORMAT(1H1/7X,*1H RESPONSE-STATE HOW-SUM CORRELATION MATRIX//)
CALL "P(MX,MX,NH,NX,P)
DO 1115 I=1,NX
DO 1115 J=1,NX
IF((I-1,LT,I,F-2)) GO TO 1115
IF((J-1,JL,T,F-2)) GO TO 1115
P(I,J)=Q(I,J)/SORT((I.I)*(J.J))
1115 CONTINUE
WRITE(*,49)
49 FORMAT(1H1/7X,*1H MEASUREMENT CROSS-CORRELATION MATRIX//)
CALL "P(MX,MX,NH,NX,P)
DO 1130 I=1,NX
DO 1130 J=1,NX
P(I,J)=0.
P(I,J)=AM(I,J)*P(I,J)
1130 CONTINUE
WRITE(*,50)
50 FORMAT(1H1/7X,*1H CONTROL-RESPONSE POUR-SUM CORRELATION MATRIX//)
DIAGRAPH FULL
Figure 94. Program DIAK Program Listing (Continued)
IF(IN[1:1].LT.1.F-20) GO TO 11
IF(IN[1:1].LT.1.F-20) GO TO 11
P(I,J)=UP(I,J)/SORT(SQRT(I,J)=UP(J,J))
61 CONTINUE
WRITE(9,62)
62 FORMAT((1X)/7X.4IN TOTAL RESPONSE CROSS-CORRELATION MATRIX/)
CALL PMX,PMX,NNMPN)
DO 699 1=1,6
699 FORMAT(9.99))
6882 WRITE(11),INT((1.J))
WRITE(9,63) (1,J)=1,NM)
6881 FORMAT(1H)/7X,2MTOTAL RESPONSES/(7X,13,3X,E15.11)
WRITE(9,64)) A)
6578 FORMAT(1H)/7X,70DIETIC COST = +EI5,81)
A59 CONTINUE
IF(NOC=0,1) GO TO 997
IF(NSTEP,0.0,NO1,2-2.0) GO TO 997
CALL NERV(A2O3X4,1,A1x11x1X2X2X13,1X3,1X5,1x6.1x7.1X8.1X9.1X10,1+11.1+121+131+14.1+151+16.1+17.1+118.1+119.1+201+21.1+221+231+241+251+261+271+281+291+30)
WRITE(8,9) NAMF1,NAMF2,SCALE,E1Y,T1,E1OG)
997 CONTINUE
CALL COLES(INX,AHM,21,2)
IFEMP3
1200 READ(1,12) INHM
IF(INHM,61.1.1) GO TO 7777
1215 FORMAT(31))
READ(24,16) inpJ,inpK
1216 INHM=RUN+1
REWIN: 2
READ(1,F)
READ(8,1AAM
READ(2811 CONT
READ(1,281 NOCOVSTEP1+RANV1+PRIN+PLOT
WRITE(9,37)) INPM+CONT2+NOCOVSTEP1+RANV1+PRIN+PLOT
IF(INHM,61.1.1) GO TO 1216
WRITE(9,171) IFLM,INHM
GO TO (121.1,123=1.125=1.126=1.127=11PM
7777 CONTINUE
END IF 6
END

Figure 84. Program DIAK Program Listing (Concluded)
Figure 85. Program FFOC Program Listing
Figure 85. Program FFOC Program Listing (Continued)
1502 DO 14 5 I=1,NI
DO 14 6 J=1,NX
AK(I,J)=.
DO 14 6 K=1,NX
1405 AK(I,J)=AK(I,J)+AKC(I,K)*GS(K+J)
C
C DEFINE K2
CALL INPT(AK,MU1,MM)
DO 14 4 I=1,NI
DO 14 4 J=1,NX
AK(I,J)=AK(I,J)+AKP(I,J)
1404 WRITE(9,126)
1018 FORMAT(1H1/7X,3H INITIAL GAINS -- I(I//)
CALL 'P(MU1,MX,UX,NU,AK )
WRITE(9,126)
1019 FORMAT(///7X,10H K2 UMTPIX//)
CALL 'P(MU1,MX,UX,NU,AK)
IF(ALM+LT,.99) GO TO 140A
DO 14 7 I=1,NI
DO 14 7 J=1,NX
1407 DELK(I,J)=0.
GO TO 140H
C
C INPUT P-SENT FIXED GAINS -- I(I(LAMDA)
1406 CALL 'PEHO(AK,MU1,MM)
CALL INPT(AK,MU1,MM)
WRITE(9,126)
1020 FORMAT(///7X,3H PRESENT FIXED GAINS -- I(I(LAMDA)//)
CALL 'P(MU1,MX,UX,NU,AK )
C
C INPUT FIXED PREDICTOR -- DELK(I(LAMDA)A
CALL 'PEOI(DELK,MU1,MM)
CALL INPT(DELK,MU1,MM)
1409 WRITE(9,121)
1021 FORMAT(///7X,3H PRESENT PREDICTOR -- DELK(I(LAMDA)A//)
CALL 'P(MU1,MX,UX,NU,DELK)
C
C TAKE STEP IN LAMDA
172 ALAM=XM-ALAM
WRITE(9,123)
173 FORMAT(///7X,9H LAMDA = .F4.3)
C
C PREDICT GAINS FOR NEW LAMDA
DO 31 I=1,NI
DO 31 J=1,NM
DK(I,J)=DELK(I,J)
310 AK(I,J)=AK(I,J)+DK(I,J)
C
C INITIAL CONDITIONS
C
NL=1
LAST=0
NC=0
NO=6
NIT=5
EPS=EPS1
AjI=1.*AJT
AJL=AJT
WRITE(9,125) NIT
4051 FORMAT(///7X,9H ITERATION,11)
WRITE(9,125) EPS

Figure 85. Program FFOC Program Listing (Continued)
C PRINT *, INS
C WRITE (4,6005)
C 6005 FORMAT (7X,13H1M GAINS MATRICE)
C CALL IP (MUM, MM, S11N, NX, AK)
C C INITIALIZE GRADIENT PROJECTION
C IF (NG.NE.1) GO TO 5
C DO 16 J=1, NX
C 16 AK(I, J) = AK(I, J)
C C INITIALIZE ARRAYS
C COMPUTE FGK*(LAMBD,
C C 5 NO 12 J=1, NX
C NO 12 I=1, NU
C CI(I, J) = CI(I, J)
C NO 12 K=1, NM
C CI(I, J) = CI(I, J) + (AK(I,K) + RK(I,K)*ALAM)*AM(K, J)
C CONTINUE
C IF (LAST, NE, 1) GO TO 66
C DO 15 I=1, NU
C 15 J=1, NM
C 151 DJK(I, J) = AK(I, J) + ALAM*AK(I, J)
C WRITE (4, 152)
C 1022 FORMAT (7X, 37H K* (LAMBD, A) FOR RESPONSE CALCULATIONS/)
C CALL IP (MU, M, MM, MU, NM, DJK)
C 66 CONTINUE
C DO 4 I=1, NX
C 4 NO 4  J=1, NX
C 4 NO 4 J=1, NX
C AI(I, J) = F(I, J)
C DO 4 K=1, NU
C 4 AI(I, J) = AI(I, J) + (AK(I,K) + RK(I,K)*C(K, J))
C IF (IN, NE, 1) GO TO 144
C C CHECK FOR STABILITY (IF A)
C CALL DOLS (IX, DX, XM, RR, M)
C KK = 0
C II = 1
C 921 IF (RR(II), LT, 0.) GO TO 185
C C IF UNSTABLE -- HALVE DELTA LAMBD A AND PREDICTOR
C 69 IF (NU, NE 1) WRITE (9, 184)
C 184 FORMAT (7X, 14H1M INSTABILITY -- CHANGE GAINS//7X, 13, 14M INSTABILITY/)
C IF (NU, NE, 1) GO TO 5071
C ALAM = ALAM + DELT
C IF (NU, NE, 1) GO TO 5071
C C FIRST OR THIRD INSTABILITY -- HALVE PREDICTOR
C 5070 DELT(I, J) = D(K(I, J))
C AIJ = ISTE(I, J)
C NUST + NUST 1
C GO TO 172
C C SECOND OR FOURTH INSTABILITY -- HALVE DELTA LAMBD A
C C ** MODIFICATIONS
C IF (DELT.LE.1.0E-36) WRITE (9, 7740) DELT
C 7740 FORMAT (7H1M //7X, 13, 14M** EXIT IN DETECTING VERY SMALL VALUE FOR DELT/)
C IT **, G18, 4)
C C Figure 85. Program FFOC Program Listing (Continued)
Figure 85. Program FFOC Program Listing (Continued)
GO TO 63
IF(NG,NE,1) GO TO 43
GO TO 1756

C CALCULATE COST
C AJT=1.
DO 74 J=1,NX
DO 74 I=1,NM
AJT=AJT*(1+X(I,J))
WRITE(9,52) AJT
127 FORMAT//7X,9H COST = X(E,I,J)
IF(NG,NE,1) GO TO 10
IF(AJT.LT.1.0) GO TO 1510
IF(AJT.LT.1.0) GO TO 14

1510 WRITE(9,55)
1402 FORMAT//7X,9H COST EXCEEDS 10 TIMES LOWEST COST EXPECTED//
WRITE(9,56)
14 FORMAT(1H/7X,9H COVARIANCE MATRIX//)
CALL SMTA3749
STOP 11
16 IF(NG,NE,1) GO TO 56

C COMPUTE XM, XM INVERSE
C DO 12 I=1,NX
C DO 12 J=1,NI
C C(I,J)=...
125 C(I,J)=C(I,J)+(I*J)*AM(I,J)
CALL TRANSAMT(AMT,X,DOUMX,MX,MX,F,M,FM,FM,FM,F)
C COSTATE CALCULATION
C CALL COSTAT (R+S,X+ES,Y+Z+E,U,V,KW,ND,TF,TF,TF,TF,TF,TF,TF,TF,TF,TF)
IF(ENP,ENP,1) GO TO 1400
WRITE(9,1401)
1401 FORMAT//7X,9H COST MATRIX UNSTABLE WHEN STATE MATRIX IS//1
STOP 11
C GRADIENT CALCULATION
C DO 19 J=1,NX
RI(J)=...
19 RI(J)=RI(J)+NI(I,J)+NI(I,J)*HIK(I,J)
DO 30 I=1,NI
DO 30 J=1,NX
XI(I,J)=...
30 XI(I,J)=XI(I,J)+G(I,J)*S(K,J)
DO 129 I=1,NI
DO 129 J=1,NX
DJK(I,J)=...
DO 129 K=1,NI
DO 129 X=1,NX
JJK(I,J)=JJK(I,J)+NI(I,J)+NI(I,J)*C(K,J)*S(K,J)
DO 129 X=1,NX

Figure 85. Program FFOC Program Listing (Continued)
CONTINUE
C PROJECTED GRADIENT
DO 14 I=1,NU
DO 14 J=1,NM
L4 DJK(1,J)=AKP(1,J)-AK(1,J)
GO TO 154
C CALCULATE RESPONSES
C CONTINUE
GO TO (155, 156, 157) NODC
151 CALL GFSPCA(A, VM, DI, K, Y, NS, ES, F, VR, K0, AKG, D00, HK, OA, FOC)
GO TO 152
152 FORMAT (6X, 93H RELATED NO FLOW, NODC)
CONTINUE
153 CALL HOLES (X, A, M, DR, M)
150 CONTINUE
IF (LAST.EQ.1) GO TO 57
IF (NL.EQ.1) AJLAT=AJT
NREGI = 0
IF (NL.EQ.1) GO TO 43
IF (NL.EQ.2) GO TO 1746
CALL TINV (ISOL, INSL, NF, T, MF, W, DET)
IF (NC.EQ.16) LE, 2) GO TO 442
WRITE (10, 123)
123 FORMAT (A10, 1X, 7X, 18H GRADIENT TRANSFORMATION NOT INVERTIBLE )
STOP 2
125 FORMAT (1X, 7X, 18H GRADIENT TRANSFORMATION MATRIX )
CALL INCHR ( X, VM, JD, NW, T, MF, W )
CALL HOLES ( H, T, MF, PR, M )
1756 IF (NL.GT.1) GO TO 560
IF (NL.GT.0) GO TO 560
C CORRECT FOR INSTABILITY WHILE COMPUTING GRADIENT
NGH=1
IF (AJT.GT.AJM) NGH=0
IF (AJL.GT.ALM) NGH=0
IF (NL.EQ.1) GO TO 500
NGRA=1
EPS=E-50
DO 52 I=1,NU
DO 52 J=1,NM
AK(I,J)=AK(I,J)-DK(I,J)
DK(I,J)=EPS*DK(I,J)/FPS
IF (NC.EQ.2) AK(I,J)=AK(I,J)+2.*DK(I,J)
IF (NC.EQ.3) AK(I,J)=AK(I,J)+.5*DK(I,J)
IF (NC.EQ.5) AK(I,J)=AK(I,J)+DK(I,J)
51 CONTINUE
GO TO 7
500 IF (NC.EQ.1) GO TO 50A
C COMPUTE RATIO OF COSTS
ROC=AJT/AJLAT
WRITE (9, 530) ROC
5930 FORMAT (1X, 15H RATIO OF COSTS = , F10.4)
IF (ROC.GT.DROC) LAST=1
AJLAT=AJT
IF (INIT.GT.NTIM) LAST=1
C NORMALIZE GRADIENT AND COMPUTE DELTA GAINS
C SUM=0
500A SUM=SUM+DJK(I,J)*DK(I,J)
Figure 85. Program FFOC Program Listing (Continued)
SUM=SUM(T(SUM))
WRITE(9,5031) SUM
5031 FORMAT(/X,7X,1AH,GRADIENT NORM = x(15.4))
5009 DO 10 J=1,NU
DO 11 J=1,NM
10 DJOK(J,J)=DJOK(J,J)/SUM
11 WRITE(4,34)
39 FORMAT(/X,7X,1AH,NORMALIZED GRADIENT/)
38 CALL IP(MU,NN,N1,N,J,DK)
5010 NL=0
IF(LA;TE,1.E0,1) GO TO 5
C COUNT GRADIENT DIRECTIONS
NGRD=NGD+1
NGD=NGD-1
NGHD=NGD
NGHD=NGD+1
NGM=NGM+1
NGM=NGM-1
GO TO 102
102 IF(NGHD, EQ.2) NCH =1
IF(NGHD, EQ.2) NGD =1
110 AJO =AJL
AJQ =AJL
AJQ =AJL
DO 42 I=1,NU
DO 42 J=1,NM
AKG(I,J)=AKG(I,J)
42 DK(I,J)=EPS*0.10*(I,J)
NGD=0
NC=1
NCD=0
GO TO 44
C STEP SIZE LOGIC
43 IF(AJL.LT.0.1) GO TO 301
IF(AJL.LT.AJL) GO TO 41
C UNSTABLE --HALVE STEP SIZE
301 NC=NC+1
NIT=NIT+1
IF(NC,GT.1) NIT=NIT-1
AJL=AJL
AJL=AJL
EPS=EPS/2.
C *** MODIFICATIONS
7720 FORMAT(1H1,/X,5X,2H*** EXIT ON DETECTING VERY SMALL VALUE FOR EPSFFOC 500
1 ***/12,4,)
IF(EPS.LE.1.0E-06) WRITE(9,7721) EPS
7721 FORMAT(1H1,/X,15X,2H*** EXIT ON DETECTING VERY SMALL VALUE FOR EPSFFOC 500
1 ***/12,4,)
C *** MODIFICATIONS
5012 DO 12 I=1,NU
DO 12 J=1,NM
AK(I,J)=AKG(I,J)
123 DK(I,J)=EPS*DK(I,J)
GO TO 44
41 IF(NC,GT.1) GO TO 45
IF(AJL,GT.AJL) GO TO 47
IF(NC,GT.1) GO TO 47
C DOUBLE STEP SIZE
5013 DO 46 I=1,NU
DO 46 J=1,NM
46 AK(I,J)=AKG(I,J)+DK(I,J)
NIT=NIT+1

Figure 85. Program FFOC Program Listing (Continued)
Figure 85. Program FFOC Program Listing (Continued)
Figure 85. Program FFOC Program Listing (Concluded)
Figure 86. Program DATAK Program Listing

DATAK 65


DATAK 66


DATAK 67

DATAK 68

DATAK 69

DATAK 70

DATAK 71

DATAK 72

DATAK 73

Figure 86. Program DATAK Program Listing (Concluded)
Figure 87. Subroutine KORG2 Flow Chart
Figure 87. subroutine KORG2 Flow Chart (Concluded)
Figure 88. Subroutine KORG2 Program Listing
Figure 88. Subroutine KORG2 Program Listing (Continued)
Figure 88. Subroutine KORG2 Program Listing (Concluded)
Figure 89. Subroutine TIMER Program Listing
DO 3 J=1,NN
3 GS(I,J)=0.
NFG=N
IF(NSTEP.EQ.4) GO TO 7
IF(NR,N0,EQ.0) GO TO 4
NFG=N+NC
DO 5 I=1,NFG
5 Q(I,J)=0.05(J+1)/S(N0(T))
DO 4 J=1,NN
4 NFG=N
10 IF(NSTEP.EQ.2) GO TO 4
IF(NSTEP.EQ.0) GO TO 7
DO A =1,NF
DO B =1,NC
JG(J)=J
JG(J+1)=J-NC
8 GS(I,J)=A(I,J)*CL(J,G+1)
IF(NSTEP.EQ.3) GO TO 7
6 NFG=N+NLG
7 CONTINUE
WRITE(9,161) NFP
101 FORMAT (7X, "99THRESH ARE [13,21]H RESPONSES TO COMPUTE//")
NT=T/IT
S=NT
S=S+DT
IF(S.EQ.1.0) NT=NT+1
NT=NT+1
NTP=S*2
S=S+NTP
S=S+DT
IF(S.EQ.1.0) NTP=NTP+1
IF(NT.EQ.1.0) NTP=1
NTP=S*2
NPTOT=N
N=1
DO 51 J=1,NN
IF(NSTEP.EQ.4) GO TO 51
IF(J.NE.1) GO TO 41
IF(NR+N0,EQ.0,RAND,STEP.EQ.0)) GO TO 12
GO TO 51
41 IF(NSTEP.EQ.0,EQ.0,STEP.EQ.2) GO TO 12
51 DO 11 I=1,II
X(I,J)=X(I,J)
11 DO(I,J)=X(I,J)
IF(J.NE.1) GO TO 56
IF(NSTEP.EQ.1) GO TO 56
IF(NSTEP.EQ.2) GO TO 56
IF(NSTEP.EQ.0) GO TO 56
JG(J)=J
56 CONTINUE
WRITE(9,162)
24 DO 17 J=1,NT
17 IF(J.NE.1) GO TO 12
IF(NSTEP.EQ.0) GO TO 13
IF(NSTEP.EQ.1) GO TO 13
12 CALL GUST(XG,CLX,QT,IT,JN,F=NG,IT=NM+1)
CONTINUE
12 IF(NR+N0.EQ.0) GO TO 13
ETA=AN(J)
Figure 89. Subroutine TIMER Program Listing (Continued)
GO TO 14
13 FTA=1.
14 GO TO 15 IF(K=GFG)

OX(I)=GNI(I)+J)•FTA•GS(I+J)
10 K=K+1=NFG
52 OX(I)=OX(I)+X(I)•K+X(K)
16 CONTINUE IF(FA(I)=X(I)) GO TO 15
15 CONTINUE IF
16 CONTINUE IF(NT=EQ.) GO TO 19
17 IF(I=NE.) GO TO 17
18 IF(NT=EQ.) ITT=1T
19 CONTINUE IF
20 R(I)=R(I)+HDF(I,K)*X(K)
20 CONTINUE IF
21 CONTINUE IF
22 CONTINUE IF
23 CONTINUE IF
24 CONTINUE IF
25 CONTINUE IF
26 CONTINUE IF

Figure 89. Subroutine TIMER Program Listing (Continued)
201 FORMAT(1X:)
17 CONTINUE
IF(NPLOT.EQ.0) GO TO 17
IF(NPLOT.EQ.1) GO TO 18
WRITE(9,19) J
DO 32 I=1,IND
DO 31 J=1,15
31 IF(RUFT(I)=BLANK)
32 IF(YMIN(I) .LT. 1.E-2)
WRITE(9,16) (1=1,1,5)
33 CONTINUE
34 IF(YMAX(I) .LT. 1.E-2)
WRITE(9,16) (TRUF(I),I=1,5)
104 FORMAT(1X/SA10/)
105 FORMAT(6F14.3)
WRITE(9,16)
106 FORMAT(7X*H+1*5(14H+------------))
DO 37 L=1,IND
IF(L(K)=NTDP) GO TO 37
WRITE(9,16) (1=1,1,5)
107 FORMAT(1X/)
Figure 89. Subroutine TIMER Program Listing (Continued)
DO 47 I = 1, 15
   47 I RUF(I) = BLANK
    CONTINUE
32 CONTINUE
30 CONTINUE
IF (NPTOT.EQ.2) GO TO 12
FJ = J
   CALL SYMBOL(0, 0, 14, TIME RESPONSE for DISTURBANCE = 0, 30)
   CALL SYMBOL(0, 0, 14, FJ = 90, 360)
   CALL PLOT(1, 0, 0)
   IARX = IOP * (NTP - 1)
   NO 48 X = 1, NTP
   IF (AUX(K), EQ, 6) GO TO 49
   IARX(K) = NTP - 1
   CALL SCALE(R(IARX), R(NTP), 1)
   CALL SCALE(R(IARX) + 10, R(NTP), 1)
   GO TO 50
49 IARX(K) = (NTP - 1)
   IF (YMIN(K), EQ, YMAX(K)) GO TO 48
   R(IARX) = YMIN(K)
   R(IARX + 1) = (YM(X) - YMIN(K)) / R.
   IARX(K) = NTP
   R(IARX) = 0.
   R(IARX + 1) = T / 10.
50 I RUF(I) = TITLE(K)
   IBUF(1) = BLANK
   IBUF(2) = UNIT(K)
   IBUF(3) = UNIT(K)
   I RUF(I) = TITLE(K)
   IBUF(1) = UNIT(K)
   IBUF(2) = UNIT(K)
   CALL AXIS(0, 0, 1, IBUF(I), 30; R, 90, R(IARX), R(IARX + 1))
   IARX = IARX + NTP
   IBUF(3) = BLANK
   I RUF(I) = IBUF(1)
   CALL AXIS(0, 0, 14, IBUF(I), -20, 10, 0, R(IARX), R(IARX + 1))
   IARX(K) = NTP + 1
   CALL LINE(R(IARX), P(IARX), NTCP, 0, 0)
   CALL PLOT(1, 5, 0)
   NPTOT = NPTOT + 1
   IF (NPTOT.LT, 5) GO TO 48
   NPTOT = 0
   CALL DSP(2)
   CALL PLOT(0, 0, 13, 3)
   CALL PLOT(0, 0, 13, 3)
   I RUF(I) = DATE
   IBUF(1) = BLANK
   IBUF(4) = NAME1
   IBUF(5) = NAME2
   I RUF(1) = MOVEMENT
   I RUF(2) = MOVEMENT
   I RUF(3) = IARX
   I RUF(4) = IARX
   CALL SYMBOL(0, 0, 14, IBUF(I), 0, 0)
   CALL SYMBOL(0, 0, 14, IBUF(5), 0, 0)
   CALL SYMBOL(0, 0, 14, IBUF(1), 0, 0)
   CALL SYMBOL(0, 0, 14, IBUF(2), 0, 0)
   CALL SYMBOL(0, 0, 14, IBUF(3), 0, 0)
   CALL SYMBOL(0, 0, 14, IBUF(4), 0, 0)
   CALL SYMBOL(0, 0, 14, IBUF(5), 0, 0)
   CALL PLOT(1, 0, 0)
   48 CONTINUE
12 CONTINUE
RETURN
END

Figure 89. Subroutine TIMER Program Listing (Concluded)
SUBROUTINE SGUST(A,GS,CL,X,N,T,TT,T2,J,NF,NT,T,MX,N,NN)

DIMENSION A(MX,NX),GS(MX,NN),CL(MN+1),X(NX)

IF(IT,GT,1) GO TO 5

JJ=J*J

DO 5 T=1,NF

5 GS(I,J)*A(I,J)*CL(J+1)

X(J,J)=CL(J+1)

ND1=TI/DT

NDP=TP/DT

S=ND1*NT

IF(S,T,T1) ND1=ND1+1

S=NDP*DT

IF(S,T,T2) NDP=NDP+1

ND1=ND1+1

NDP=NDP+1

3 IF(IT,LT,ND1) RETURN

IF(IT,GT,ND1) GO TO 1

JJ=J*J+NF

DO 1 T=1,NF

1 GS(I,J)=GS(I,J)*A(I,J)*CL(J+1)

X(J,J)=CL(J+1)

2 IF(IT,NF,NDP) RETURN

JJ=J*J+NF

RETURN

END

Figure 90. Subroutine SGUST Program Listing
SUBROUTINE Call

DIMENSION A(N,N),X(N+1),Y(N+1),Z(N+1),W(N+1)

IF(RE=.T.) THEN
TR=1,
DO 30 I=1,N
TR=TR+A(I+1)
FN=TR
IF(TR=3.1+2.) THEN
2 ERR=.T.
GO TO 60
30 ALF=A:5(TR)/FN
NC=N*(N+1)
NC=NC/I
DO 51 J=1,N
DO 50 I=1,N
0=ALFA(I,J)
50 CONTINUE
CALL INVRR(ISOL+ISOL, N+1, P=NR, KN+KWA+DET)
IF(ISOL+ISOL, LE, 10) GO TO 20
51 CONTINUE
CALL INVRR(ISOL+ISOL, N+1, P=NR, KN+KWA+DET)
NS=NS(I)
IF(NS(N+1)) GO TO 20
20 CONTINUE
DO 70 I=1,N
A(I,J)=
70 CONTINUE
DO 110 J=1,N
DO 100 I=1,N
4 A(I,J)=A(I,J)*P(K+1)*XN(K,J)+ALF
100 CONTINUE
DO 110 J=1,N
DO 90 I=1,N
A(I,J)=
90 CONTINUE
DO 90 J=1,N
DO 110 I=1,N
9 A(I,J)=A(I,J)*P(K+1)*XN(K,J)
DO 110 I=1,N
111 DZ1=DX1/A(I+1)*P(K+1)
11 DZ1=DX1/A(I+1)*P(K+1)
DO 300 I=1,N
DO 200 J=1,N
DXIJ=.
DO 200 K=1,N
11 IF(FX*LT,1.0-E-29) GO TO 14
14 ERR=.T.
GO TO 60
201 RAT=A(A*DX1/XN(I,J))
IF(RAT=EE) GO TO 60

Figure 91. Subroutine CALL Program Listing

258
14 ICOT=ICOT+1
70 CONTINUE
10 CONTINUE
18 ITER=ITER+1
IF(ICOT=NC115.5J.15
15 CONTINUE
DO 20 I=1,N
DO 20 J=1,N
10 A(I,J)=P(I,J)
16 DO 17 I=1,N
DO 17 J=1,N
P(I,J)=0.
DO 17 K=1,N
17 P(I,J)=P(I,J)+A(I,K)*A(K,J)
40 IF(ITER=IMAX)(93,56,50
50 CONTINUE
WRITE(9,600) ITER
600 FORMAT(9,6H ITER=12)
RETURN
601 WRITE(9,602) IERR
602 FORMAT(9,6H IERR=12)
RETURN
END

Figure 91. Subroutine CAL1 Program Listing (Concluded)
SUBROUTINE STRIC(A,H,P,W,S,TFF,NX,NU,XMX,W)
DIMENSION A(NMX,NMX),H(NMX,NMX),P(NMX,NMX),S(NMX,NMX)
DIMENSION TFF(NMX,NMX),W(J),X(NMX,2)
OT=0.
DO I=1,NX
DO J=1,NX
W(I,J)=0.
DO 1,1=1,NX
1 W(I,J)=W(I,J)+(I*Y)*S(J,K)*DT
NT=10.
KT=10.
T=T+DT
DO 3 T=1,NX
DO 4 J=1,NX
S(I,J)=S(I,J)
4 TPF(I,J)=S(I,J)
   S(I,J)=S(I,J)
   TPF(I,J)=TPF(I,J)
   A(I,J)*S(I,J)*FAC*T
   S(I,J)=S(I,J)+PS(I,J)
   DO 7 J=1,NX
   TPF(I,J)=TPF(I,J)
   100 CONTINUE
   DO 9 J=1,NX
   TPF(I,J)=TPF(I,J)+S(I,J)*R(I,J)
   DO 11 J=1,NX
   R(I,J)=R(I,J)
   4 TPF(I,J)=TPF(I,J)+S(I,J)*R(I,J)
   DO 9 J=1,NX
   9 W(I,J)=W(I,J)+TPF(I,J)*TPF(I,J)*DT
   200 CONTINUE
WRITE(*,300)
300 FORMAT(1H/7X,12H W(T,I) MATRIX/)
CALL DP(NMX,NMX,NX,NU,W)
RETURN
END
Figure 92. Subroutine STRIC Program Listing
SUBROUTINE SHUFL(A,MM,NN,N,NCM,NORD,B,MAX)
DIMENSION A(MM,NN),NORD(MX),B(MX,MAX)
IF(MC.EQ.0) GO TO 1
DO 2 T=1,N
   1 IF(NOOD(I))
   DO 2 J=1,N
2   A(I+J)=A(I+J)
   NO 3  T=1,N
   NO 3  A(I+J)=A(I+J)
   CONTINUE
   IF(NC.EQ.0) RETURN
   DO 4 I=1,N
   JJ=NOOD(J)
4   DO 4 I=1,N
   DO 5 J=I+1,N
5   A(I+J)=A(I+J)
   DO 5 J=1+I
   RETURN
END

Figure 93. Subroutine SHUFL Program Listing
SUBROUTINE SHUF(F,G1,G2,H,AN,AKG,Y,ND,MAX,NX,NR,NN,NM,NU)
DIMENSION F(MAX),G1(MAX,NU),G2(MAX,NV),H(MAX,NX),AN(MAX,NX)
DIMENSION Y(MAX,NX)
DO 1 I=1,NX
 1 Y(I,J)=F(I,J)
   JJ=NORD(J)
DO 2 I=1,NX
 2 Y(I,J)=Y(I,J)
   II=NORD(I)
DO 4 J=1,NU
 4 Y(I,J)=G1(I,J)
DO 3 J=1,NN
 3 Y(I,J)=G2(I,J)
   JJ=NORD(J)
DO 5 I=1,NR
 5 Y(I,J)=H(I,J)
   II=NORD(I)
DO 8 J=1,NU
 8 Y(I,J)=Y(I,J)
   JJ=NORD(J)
DO 7 J=1,NN
 7 Y(I,J)=H(I,J)
   II=NORD(I)
DO 10 I=1,NM
10 Y(I,J)=AN(I,J)
   JJ=NORD(J)
DO 9 I=1,NN
 9 Y(I,J)=AN(I,J)
   II=NORD(I)
DO 11 J=1,NU
11 Y(I,J)=AKG(I,J)
   JJ=NORD(J)
DO 12 J=1,NN
12 Y(I,J)=Y(I,J)
RETURN
END
F

wiaw'mmmmm

""" -

~^

—

>
——-P———.

i»«I»«MSäS»^!*-W»l>W«IK?msM"p-i'«i:.J.v,.'.«s.-m»

,

-

^

r ,

SUBROUTINE RESP(C<A<G?«AM«AK>XtYfZ«St4«ES>EtU*V«XIf00«*KG«0a0fH0KtRESP
|KW*.N«tNfT.NN>NMfNMtNR.MX,MFF.HFe»MN.'«M,MU,MRtlTER»IM*Ä.IE«R»NCOV>R£SP
DIMENSION XKMFFtMrF), X (MX .MX ) «R (MX« <4X) «HOK <MP«HX ) «C (MX tMX) t
RESP
IG2(MX.MN)*A(MX«MX),AM(MM«MX)t«K(MO«MMtfY(«X«MAt«Z(MX«MX)«S(MX«MX)«RCSP
2ES(MX.MX)«U(MFR«MrR)«V(MrF«MFn«E(MFF>Mrri«D0(MU*MM)«
RESP
3KMA(M«)«AKG(MU<HX).nOO(MU.MUl
RESP
NFB-Nx-NFr
RESP

RESP
RESP
RESP
RESP
RESP
RESP
KCOMx
RESP
6076 KCOMxCOM*!
RESP
00 *0'0 I«I»NFF
RESP
DO «O'O J«1«NFF
RESP
II«I»MFH
RESP
JJ«J»-4FB
<»02o c(i.J'=r.2(ii
<KCnH)*G?(jj
«KCOM)
RESP
RESP
WRITER.M» KCOM
RESP
«1 FORMAT(1H1/7X«36H COVARIANCE ANALYSIS FOR DISTURBANCE«13//)
RESP
ITER«.
RESP
1ITER«'«IERR«KWA»
RESP
IF(IEPR.EQ.O) r,0 TO 89ft
RESP
RESP
MRITE(9«0)
43 FORMAT(|Hl/7X«?SH COVARIANCE MATRIX UWEFINED//7X.27H IGNORE COVARRESP
RESP
1IANCE ANALYSIS//)
RESP
RETURN
RESP
896 MRITE(9«<i»0Sl)
RESP
<.05l FORMAT(//7X«IRH COVARIANCE MATRIX//)
CALL MP(MX«MX«NX«NX«X)
RESP
RESP
c
COMPUTE (H»OKM)X(H*OfM)
RESP
c
RESP
DO *0S3 I»l«NR
00 *0S3 J>1«NX
RESP
RESP
C(I«JI«0.
RESP
DO <»0S3 Kal«NX
4053 C(I«J)>C(I«J)*H0K(I«K)«X(K«J)
RESP
RESP
DO 40S4 I»1«NR
RESP
DO 40^4 J>I«NR
RESP
S(I«J)«0*
RESP
DO *OSfc K3|«NX
4054 S(I«J)aS(I«J)*C(I«K)*HDK(J*K)
RESP
RESP
IF(NC0V.GT.2) GO TO 2
MRITE(9«42)
RESP
RESP
42
FORMAT(»Hl/7X.?7H RESPONSE COVARIANCE MATRIX//1
RESP
CALL MP(MX«MX«NR«NR*St
RESP
DO 701? I>I«NX
RESP
DO 70\5a J>1«NM
RESP
ES(J«n O.
RESP
DO 7015 K«I«NX
7015 ES(J«n*ES(J«I)*XII«Kt««M(J.K)
RESP
RESP
00 701ft I>1«NM
RESP
00 7016 J»1.NM
Y(I«Jtx«.
RESP
RESP
DO 7016 KM.NX
RESP
7016 V(I«J><Y(I«J)*AM(I,K)*ES(J«)()
RESP
«RITE(9.44)
RESP
44 FORMAT(IH1/7X«30H MEASUREMENT COVARIANCE MATRIX//)
CALL MP(MX«MX*NM«NM«Y)
RESP
RESP
DO 1112 I«!.NU
DO III? J»I.NM
RESP

C

COMPUTE COVARIANCE MATRIX FOR DISTURBANCE «COM
DO f>0A0 I>1«NR
DO «0*t J'ltNR
6080 R(I«Jl«0.

...»»»^^

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
5«
59
60
61
62
63
64

Figure 95. Subroutine RESP Program Listint

263
.
.. ■■.J-.^L^g'^»^


**Figure 95. Subroutine RESP Program Listing (Continued)**

```fortran
DO (i = 1, NX)
  DO (j = 1, NX)
    CALL YP(MX, MX, NX, NU, Z)
    DO (i = 1, NU)
      DO (j = 1, NX)
        AKG(i, j) = 0.
        IF (X(i, j) .LT. 1.E-2(1)) GO TO 1113
        IF (X(i, j) .LT. 1.E-2(0)) GO TO 1113
        Z(i, j) = Z(i, j) .LT. X(i, j) / SQRT(Y(i, i) * X(j, j))
    CONTINUE
  CONTINUE
END
1113 CONTINUE
WRITE (9, 47)
1114 CONTINUE
WRITE (9, 48)
1115 CONTINUE
WRITE (9, 49)
1116 CONTINUE
WRITE (9, 50)
1117 CONTINUE
WRITE (9, 51)
1118 CONTINUE
WRITE (9, 52)
1119 CONTINUE
WRITE (9, 53)
1120 CONTINUE
WRITE (9, 54)
1121 CONTINUE
WRITE (9, 55)
1122 CONTINUE
WRITE (9, 56)
1123 CONTINUE
WRITE (9, 57)
1124 CONTINUE
WRITE (9, 58)
1125 CONTINUE
WRITE (9, 59)
1126 CONTINUE
WRITE (9, 60)
1127 CONTINUE
WRITE (9, 61)
1128 CONTINUE
WRITE (9, 62)
1129 CONTINUE
WRITE (9, 63)
1130 CONTINUE
WRITE (9, 64)
END
```

Figure 95. Subroutine RESP Program Listing (Continued)
1300 CONTINUE
00 WRITE(9,1302)
1302 FORMAT('1/1/7X,43H MEASUREMENT-STATE CROSS-CORRELATION MATRIX/"
00 CALL MP(MX,MX,NX,Z)
00 DO 1116 I=1,NU
00 DO 1117 J=1,NN
00 Z(I,J)=0.
00 IF(D00(I+1),LT,1.E-20) GO TO 1116
00 IF(Y(I,J),LT,1.E-20) GO TO 1116
00 Z(I,J)=D00(I,J)/SORT(D00(I,J)*Y(J,J))
1116 CONTINUE
00 WRITE(9,50)
50 FORMAT('1/1/7X,43H CONTROL-MEASUREMENT CROSS-CORRELATION MATRIX/"
00 CALL MP(MX,NU,NV,NM,Z)
00 DO 1118 I=1,NR
00 DO 1119 J=1,NN
00 Z(I,J)=0.
00 IF(D00(I+1),LT,1.E-20) GO TO 1118
00 IF(Y(I,J),LT,1.E-20) GO TO 1118
00 DO 1119 K=1,NX
00 Z(I,J,K)=0.
00 IF(D00(I+1),LT,1.E-20) GO TO 1119
00 IF(Y(I,J,K),LT,1.E-20) GO TO 1119
00 CALL MP(MX,NUM,NP,Z)
1119 CONTINUE
00 WRITE(9,51)
51 FORMAT('1/1/7X,43H RESPONSE-MEASUREMENT CROSS-CORRELATION MATRIX/"
00 CALL MP(MX,NU,NP,NM,Z)
00 DO 1120 I=1,NP
00 DO 1121 J=1,NR
00 Z(I,J)=0.
00 IF(D00(I+1),LT,1.E-20) GO TO 1120
00 IF(Y(I,J),LT,1.E-20) GO TO 1120
00 DO 1121 K=1,NPM
00 Z(I,J,K)=0.
00 IF(D00(I+1),LT,1.E-20) GO TO 1121
00 IF(Y(I,J,K),LT,1.E-20) GO TO 1121
00 CALL MP(MX,NPM,NR,Z)
1121 CONTINUE
00 WRITE(9,52)
52 FORMAT('1/1/7X,43H CONTROL-RESPONSE CROSS-CORRELATION MATRIX/"
00 CALL MP(MX,NP,NM,NR,Z)
00 2 DO 4056 I=1,NR
00 IF(S(I+1),LT,0.) S(I+1)=0.
00 R(I+1)=R(I+1)/S(I+1)
4056 S(I+1)=SORT(S(I+1))
00 CALL MP(MX,NPM,NR,Z)
4057 FORMAT('//20X,17H RESPONSES/1AX,13,E16.8)
00 IF(6802.1,GT,0.1) GO TO 6070
00 DO 6042 I=1,NR
00 IF(S(I+1),LT,0.) S(I+1)=0.
00 IF(6082.1,GT,0.1) GO TO 6081
00 RETURN
6081 FORMAT('//7X,22HTOTAL R.M.S. RESPONSES/1AX,13,E16.8)
RETURN
END

Figure 95. Subroutine RESP Program Listing (Concluded)

265
SUHRUITINE COVAR(X1,C1,X2,C2,E,V,N1,N2,M1,M2,F,M,F1,F2) COVAR 2
DIMENSION C(MX+NX),X(MX+NX),G2(MX+NX),S(MX+NX),N1(NF),N2(NF)
X(MX+NX)=X(MX+NX)+1 IF(I1(I1(E),COVAR 4
N2=N2-N1*G2 SUMI=1 I2=1 G2
IF(I1(R=N2)) GO TO 150 COVAR 6
C COVARANCE CALCULATION
C COVARANCE CALCULATION
C COMPUTE A22 FROM G=A22*A22 X2?+X2?+A22*G22*G22
DO 11 I=1,NFF COVAR 11
DO 11 J=1,NFF COVAR 12
II=I+0FH COVAR 13
JJ=J+0FH COVAR 14
S1[I,J]=S1[I,J]+1 IF(I1(R,S2) GO TO 11 COVAR 15
C1[I,J]=0 COVAR 16
DO 12 K=1,NFF COVAR 17
CONTINUE COVAR 19
CALL CALL(S1,C1,X2,NFF,MX,M1,M2,IE) COVAR 20
IF(IE-P,GT.0)) RETURN COVAR 21
DO 151 I=1,NFF COVAR 22
DO 151 J=1,NFF COVAR 23
CONTINUE COVAR 25
C COMPUTE X12 FROM G=A11*X12*X12+A12*X12
DO 150 I=1,NFF COVAR 26
DO 150 J=1,NFF COVAR 27
II=I+0FH COVAR 28
J=J+0FH COVAR 29
DO 152 K=1,NFF COVAR 31
DO 152 J=1,NFF COVAR 32
II=I+0FH COVAR 33
J=J+0FH COVAR 34
S1[I,J]=S1[I,J]+1 IF(I1(R,S2) GO TO 153 COVAR 35
CALL CALL(S1,C1,X12,NFF,M1,M2,IE) COVAR 36
IF(IE-P,GT.0)) RETURN COVAR 37
DO 154 I=1,NFF COVAR 38
DO 154 J=1,NFF COVAR 39
II=I+0FH COVAR 40
J=J+0FH COVAR 41
S1[I,J]=S1[I,J]+1 IF(I1(R,S2) GO TO 155 COVAR 42
CALL CALL(S1,C1,X12,NFF,M1,M2,IE) COVAR 43
IF(IE-P,GT.0)) RETURN COVAR 44
DO 154 I=1,NFF COVAR 45
DO 154 J=1,NFF COVAR 46
II=I+0FH COVAR 47
J=J+0FH COVAR 48
S1[I,J]=S1[I,J]+1 IF(I1(R,S2) GO TO 155 COVAR 49
CALL CALL(S1,C1,X12,NFF,M1,M2,IE) COVAR 50
IF(IE-P,GT.0)) RETURN COVAR 51
DO 155 I=1,NFF COVAR 52
DO 155 J=1,NFF COVAR 53
II=I+0FH COVAR 54
J=J+0FH COVAR 55
X[I,J]=X[I,J]+1 IF(I1(R,S2) GO TO 155 COVAR 56
DO 155 I=1,NFF COVAR 57
DO 155 J=1,NFF COVAR 58
II=I+0FH COVAR 59
J=J+0FH COVAR 60
CONTINUE COVAR 61
CALL CALL(S1,C1,X12,NFF,M1,M2,IE) COVAR 62
IF(IE-P,GT.0)) RETURN COVAR 63
DO 155 I=1,NFF COVAR 64
DO 155 J=1,NFF COVAR 65
II=I+0FH COVAR 66
J=J+0FH COVAR 67
CONTINUE COVAR 68

Figure 96. Subroutine COVAR Program Listing
Figure 96. Subroutine COVAR Program Listing (Concluded)
SUBROUTINE COSTAT(P*X*X*ES*Y*Z+U*V+U*V+KWA*NX*4NF+NX+NFF+1MAX+KOSTAT 2
IERR)
DIMENSION Y(MX,MAX),X(MX,MAX),S(MX,MAX),ES(MX,MAX),S(MX,MAX),IxF(MX,MAX),X(MX,MAX)
COSTAT 4
1V(NFF,NFF)+N(MX,MAX),V(NFF,NFF),ES(MX,MAX),E(NFF,NFF)
COSTAT 5
NFB=N1=NFF
DO 1 I=1,NX
DO 1 J=1,NX
S(I+J)=R(I,J)
1 X(I,J)=X(I+J)
C
C COMPUTE S11 FROM 0=S11*A11+A11*S11+R11
CALL CAL(X+S*ES+KWA+NFF*MX+MAX+1)IERR)
IF(IERR,GT,0) RETURN
C
C COMPUTE S12 FROM 0=S12*A12+A22*S12+R12
DO 15 A=1,NFF
DO 15 J=1,NFF
ES(I+J)=R(I,J)
DO 15 K=1,NFF
158 ES(I+J)+ES(I+J)+S(K+1)*A(K,J)
DO 15 A=1,NFF
DO 15 J=1,NFF
159 X(I,J)=A(J+1)
DO 16 A=1,NFF
DO 16 J=1,NFF
J=J+1
160 V(I+J)=X(I+J)
CALL GCAL(Y*X*ES+NFF+NFF*MX+MAX+ES+Z+V+U+KWA+MAX+IERR)
IF(IERR,GT,0) RETURN
DO 16 A=1,NFF
DO 16 J=1,NFF
J=J+1
160 S(J+1)=ES(I+J)
162 S(I+J)+ES(I+J)
C
C COMPUTE S22 FROM 0=S22*A22+A22*S22+R22
DO 167 E=1,NFF
DO 167 J=1,NFF
I=I+1
167 J=J+1
J=J+1
168 Y(I+J)=X(I+J)
CALL GCAL(Y*X*ES+NFF+NFF*MX+MAX+ES+Z+V+U+KWA+MAX+IERR)
IF(IERR,GT,0) RETURN
DO 167 E=1,NFF
DO 167 J=1,NFF
J=J+1
163 X(I,J)=X(I+J)+A(K+1)*S(K+1)+S(K+1)*A(K,J)
CALL CAL(Y*X*R+KWA*NFF*MX+MAX+1)IERR)
IF(IERR,GT,0) RETURN
DO 167 E=1,NFF
DO 167 J=1,NFF
J=J+1
164 S(I+J)=X(I+J)
RETURN
END

Figure 97. Subroutine COSTAT Program Listing
Figure 98. Subroutine TRANS Program Listing

SURROUN1E TRANS(AUT, T, DUN, NX, MF, R(F, 9, IF)

DIMENSION AMT(MX, MF, NX, MF, R(MF, MF, IF, MF)

DO 1=1, NX
1 R(I, J)=0

DO 1=1, NX
2 T(I, J)=T(I, J) + AMT(I, K) * X(K, J)

RETURN
END
SUBROUTINE UNSCR(T,J,JMN)JVT*JF*NU*N*M(MM,MF)
DIMENSION IF(MF),JF(MF)
DIMENSION IF(NF),JF(NF),JVT(NF),JVT(NF)
L=1
DO 1 I=1,NU
DO 1 J=1,NM
IF(L,LT,NF) GO TO 1
IF(J,LT,JF(L)) GO TO 1
IF(J,LT,JF(L)) GO TO 1
JVT(L)=JVT(L)+JVT(L)
L=L+1
1 CONTINUE
DO 3 I=1,IF
DJVT(:)=0,
DO 3 J=1,IF
DJVT(J)=DJVT(J)+T(J,K)*DJVT(J)
L=1
DO 4 I=1,NU
DO 4 J=1,NM
IF(L,LT,NF) GO TO 5
IF(J,LT,JF(L)) GO TO 5
IF(J,LT,JF(L)) GO TO 5
JJK(1+J)=JVT(L)
L=L+1
5 GO TO 4
4 CONTINUE
RETURN
END

Figure 99. Subroutine UNSCR Program Listing
Figure 100. Subroutine GCAL Program Listing
DO 31 I=1,N
DO 31 J=1,N
0=0.
00 32 K=1,N
32 DO X=X+R(I,K)*V(K,J)
X(I,J) = X(I,J) + DX
AX=AR+(X(I,J))
IF(AX,LT,1.E-20) GO TO 42
IF(AX,LT,1.E-20) GO TO 41
ENDIF.
GO TO 1,1
41 PAT=A+S(DA/X(I,J))
IF(RA=EF)42,43,44
42 IDC=ICOT+1
43 CONTINUE
31 CONTINUE
ITER=ITER+1
IF(IDC-NC)44,45,46
44 CONTINUE
DG 33 I=1,N
DO 33 J=1,N
33 F(I,J)=V(I,J)
DO 34 I=1,N
DO 34 J=1,N
34 F(I,J)=F(I,J)+F(R,K)*ES(K,J)
DO 45 I=1,N
DO 45 J=1,N
V(I,J)=X
DO 45 K=1,N
DO 45 K=1,N
45 U(I,J) = U(I,J) + ES(I,K)*ES(K,J)
IF(UP,LT,IMAX) GO TO 100
WRITE(9,A(10))
600 FORMAT(7X,12H ITER = 1MAX)
50 CONTINUE
RETURN
601 WRITE (9,601) IFRR
602 FORMAT(7X,7H IFRR = 'I2)
RETURN
END

Figure 100. Subroutine GCAL Program Listing (Concluded)
Figure 101. Subroutine CAL Program Listing
14 ICOT=ICOT+1
15 CONTINUE
16 ITER=ITER+1
    IF (ICOT-NC) 15,56,15
15 CONTINUE
    DO 20 I=1,N
    DO 20 J=1,N
20   A(I,J)=P(I,J)
16   DO 17 I=1,N
    DO 17 J=1,N
    P(I,J)=0.
    DO 17 K=1,N
17   P(I,J)=P(I,J)+A(I,K)*A(K,J)
40   IF (ITFR,LT,IMAX) GO TO 160
    WRITE(9,600)
600   FORMAT(2/1X,3H IERR=12)
50 CONTINUE
    RETURN
601   WRITE(9,602) IERR
602   FORMAT(2/1X,3H IERR=12)
    RETURN
END

Figure 101. Subroutine CAL Program Listing (Concluded)
Enter

Read State Space Quadruple Data from DDATA File and the Remaining Data from Cards

Prepare the Data for DIAK Program and Write on the Scratch File JS and Write an End of File Mark on it

Read and Print the Data Written on File JS

Return

Figure 102. Subroutine DDIAK Flow Chart
Figure 103: Subroutine DDIAK Program Listing
Figure 103. Subroutine DDIAK Program Listing (Continued)
CALL -TP(D11,NP1,NH1,NRM,NUM,JS)  
GO TO 100  
280 CONTINUE  
CALL -TP(C1,NR1,NX,NRM,NUM,JS)  
GO TO 140  
285 CONTINUE  
C READ MAIN MATRIX FROM DATA FILE  
C READ(M,200)HEAD  
290 CONTINUE  
READ(M+200)CARD  
DO 295 J=1,20  
IF(CARD(J).NE.,READ(J))GO TO 200  
295 CONTINUE  
CALL -TP(M+M,NUM,NRM,NRJ)  
REWIND JO  
CALL -TP(M+M,NUM,NRM,JS)  
GO TO 190  
300 CONTINUE  
END FILE JS  
REWIND JS  
IF(I(PRINT,LT,5),AND.(I(PRINT,NE,3)))GO TO 400  
C READ END PRINT TAPSE  
C WRITE(IW,310)  
310 FORMAT(I11,I1,23H*** DIAK INPUT DATA ***/I)  
320 CONTINUE  
READ(J,120)CARD  
IF(E0F(JS))330,340  
330 WRITE(IW,330)CARD  
340 FORMAT(I11,I2,2CA4)  
GO TO 120  
360 CONTINUE  
REWIND JS  
WRITE(IW,380)  
380 FORMAT(/1X,23H END OF DIAK INPUT DATA ***)  
490 CONTINUE  
RETURN  
END

Figure 103. Subroutine DDIAK Program Listing (Concluded)
Enter

Read State Space Quadruple Data from QDATA File, Gain Data from DDATA or FDATA File and the Remaining Data from Cards

Prepare the Data for FFOC Program and Write on the Scratch File JS and Write an End of File Mark on it

Read and Print the Data Written on File JS

Return

Figure 104. Subroutine DFFOC Flow Chart
SUBROUTINE DFFOC(A,R,C,N,N,H,R,P,C1,C3,D11,RK,NAM,NRM,NUM) DFFOC 2

PURPOSE - TO PREPARE DATA FOR FFOC PROGRAM DFFOC 3

ANALYSIS - A F KONAP / J W MAHESW - THE MONTREAL INC DFFOC 4

DATE WRITTEN - 1975 DFFOC 5

PROGRAMS CALLED DFFOC 6
ZEO DFFOC 7
FILE DFFOC 8
MPS DFFOC 9
WTP DFFOC 10
INHM DFFOC 11

ARGUMENTS LIST DFFOC 12

A STATE TRANSITION MATRIX DFFOC 13
B CONTROL INPUT MATRIX DFFOC 14
C STATE OUTPUT MATRIX DFFOC 15
D CONTROL OUTPUT MATRIX DFFOC 16
R1 INPUT MATRIX FOR CONTROL INPUTS - G1 DFFOC 17
R2 INPUT MATRIX FOR GUST INPUTS - G2 DFFOC 18
C1 STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H DFFOC 19
C3 STATE OUTPUT MATRIX FOR MEASUREMENTS - M DFFOC 20
D11 OUTPUT MATRIX FOR DESIGN OUTPUTS - D DFFOC 21
RK FEEDBACK GAIN MATRIX DFFOC 22
NUM INPUT MAXIMUM NO OF STATES DFFOC 23
NRM INPUT MAXIMUM NO OF OUTPUTS DFFOC 24
NOMUX INPUT MAXIMUM NO OF INPUTS DFFOC 25

DIMENSION A(NM,NM),R(NM,NUM),C(NRM,NXM),D(NRM,NRM) DFFOC 26
DIMENSION B1(NM,NUM),B2(NM,NUM) DFFOC 27
DIMENSION C1(NUM,NM),C3(NRM,NXM) DFFOC 28
DIMENSION D11(NRM,NM),RK(NRM,NRM) DFFOC 29
DIMENSION HEAD(20),CARD(20) DFFOC 30
COMMON/INOUT/IK,IW,IPRINT,IFORT,LOCATE,NUM,MARK(20) DFFOC 31
1,J0,J+,J5,J+,JF,JD DFFOC 32
DATA -RFR4=HRBG,HRG2=HRHB+4H F +4H G1 +4H G2 +4H H / DFFOC 33
DATA -HRHB=HRHB+4H+HRCA+4H+HRG2+4H+HRCA+4H+HRHB+4H+HRCA+4H+HRHB+4H+HRCA/ DFFOC 34
DATA -HTAP=HEND+4H T+4HEND / DFFOC 35
DATA -HRHB+HRCA+4H / DFFOC 36
DATA -RHRB=HRHB+HRCA+4H+4H AK+4H+AK+4H+AK / DFFOC 37

READ IF DATA IS ON CARDS ONLY DFFOC 38
READ(K+20),CARD DFFOC 39
IF(CARD(6),EQ,'MARRH') GO TO 40 DFFOC 40
IF(CARD(6),NE,'MARRH') GO TO 100 DFFOC 41
CALL ZERO(A,NM,NUM) DFFOC 42
CALL ZERO(B,NM,NUM) DFFOC 43
CALL ZERO(C,NRM,NXM) DFFOC 44
CALL ZERO(D,NRM,NUM) DFFOC 45
CALL ZERO(D11,NRM,NM) DFFOC 46
READ(I1,X+20),HEAD DFFOC 47
20 FORMAT(20A1) DFFOC 48
CALL FILE(JO,LOCATE,HEAD) DFFOC 49
READ(10) T+NX,NU,NR+NU,(11+J)+J=1+NX,J=1+NX) DFFOC 50
1(11+J)+J=1+NX,J=1+NU,(11+J)+J=1+NR,J=1+NX) DFFOC 51
2(11+J)+J=1+NU,J=1+NU,(1+NR)+RR+NU,VR+NR3+NU14+NU2+NU3 DFFOC 52

PARTITION MATRICES A+D DFFOC 53

IF(NUX.LE.0)STOP 111 DFFOC 54
IF(NUX.GT.0)STOP 112 DFFOC 55
IF(NUX.LE.0)STOP 110 DFFOC 56

Figure 105. Subroutine DFFOC Program Listing
Figure 105. Subroutine DFFOC Program Listing (Continued)
Figure 105. Subroutine DFFOC Program Listing (Concluded)
Enter

Read State Space Quadruple Data from QDATA File and Gain Matrix Data from DDATA or FDATA Files or from Cards

Gain Matrix Data Is Read?

Yes

Compute Closed Loop State Space Quadruple Data for the System

Return

No

Figure 106. Subroutine DLSA Flow Chart
SUBROUTINE DLSA(A,R,C,D,B1,R2,C1,C3,D1,RK,RK3,NX,NR,NU,
INXM,N=NXM,NUM)

C PURPOSE - TO PREPARE DATA FOR LSA PROGRAM
C ANALYSIS - A F KONAR / J K MAHESW - THF HONEYWELL INC
C DATE WRITTEN - 1975
C
C SUBPROGRAMS CALLED
C ZERO
C FILE
C MP5
C INPUT
C
C ARGUMENTS LIST
A STATE TRANSITION MATRIX
C CONTROL INPUT MATRIX
D STATE OUTPUT MATRIX
H1 CONTROL OUTPUT MATRIX
H2 INPUT MATRIX FOR CONTROL INPUTS - G1
H3 INPUT MATRIX FOR GUST INPUTS - G2
C1 STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H
C3 STATE OUTPUT MATRIX FOR MEASUREMENTS - M
D1 OUTPUT MATRIX FOR DESIGN OUTPUTS - D
H FEEDBACK GAIN MATRIX
RK1 RK*D3
RA1 INPUT MAXIMUM NO OF STATES
NR1 INPUT MAXIMUM NO OF OUTPUTS
NU1 INPUT MAXIMUM NO OF INPUTS

DIMENSION A(NXM,NXM),R(NNM,NUM),C(NRM,NXM),D(NRM,NUM)
DIMENSION A1(NRM,NRM),R2(INM,NUM)
DIMENSION C1(NRM,NXM),C3(NRM,NXM)
DIMENSION D11(NRM,NRM),RK(INM,NRM)
DIMENSION BCK3(NRM,NXM)
DIMENSION HEAD(20),CARD(20)
COMM= INPUT/INOUT/IN/PRINT/INSERT/LOCATE/NULL/MARK(20)
1=JO,J5,JSD,JF,J0
DATA HBFBB,HBR1,HBR2,HBR4/H 44 H 44 H 44 H 44 H / 444 H 44 H 44 H 44 H / 444 H 44 H 44 H 44 H / 444 H 44 H 44 H 44 H / 444 H 44 H 44 H 44 H /
DATA HBBH,HRAK,HMC,HPCAR,HREAD/44 H 44 H 44 H 44 H 44 H 44 H 44 H / 44 H 44 H 44 H 44 H 44 H 44 H 44 H /
DATA HATAP,HEND/44 H 44 H 44 H 44 H / 444 H 44 H 44 H / 444 H 44 H 44 H / 444 H 44 H 44 H /
DATA HAPHP,HERR/44 H 44 H / 444 H 44 H / 444 H 44 H /
IGAIN=Z

C READ QUADRUPE DATA FROM ODATA FILE
C READ(1R,20) CARD
IF(CARD(6).NE.HMPEFR) GO TO 420
CALL ZFRO(A,NXM,NXM)
CALL ZFRO(B,NRM,NRM)
CALL ZFRO(C,NRM,NXM)
CALL ZFRO(D,NRM,NM)
CALL ZFRO(RK,NRM,NRM)
READ(1R,20) HEAD
C
FORMAT(20A4)
CALL FILE(JO,LOCATF,HFAD)
READ(JO,TNX,NU,J1,TA1,NX1,JM,NU1,JN1,NX1,JM1,NX),
1((RI,J)=1,NI1,NU1),((CI,J)=1,NI1,NU1),
2((DI,J)=1,NI1,NU1),((A1,J)=1,NI1,NU1)
00 28 1=1,NX

Figure 107. Subroutine DLSA Program Listing
Figure 107. Subroutine DLSA Program Listing (Continued)
RKC3(I,J)=0.0
DO 70 K=1,NP3
70 RKC3(I,J)=RKC3(I,J)+R(K)*C3(K+J)
DO 80 I=1,NX
DO 80 J=1,NX
DO 80 K=1,NUJ
80 A(I,J)=A(I,J)+R(K)*HKC3(K+J)
DO 90 I=1,NR1
DO 90 J=1,NX
DO 90 K=1,NUJ
90 C(I,J)=C(I,J)+R(K)*HKC3(K+J)
DO 100 I=1,NX
DO 100 J=1,NUJ
100 B(I,J)=H2(I,J)
IF(IPV*INT,LT,6) RETURN
CALL UPRS(A,NM,NX,NX,NX,T4HA)
CALL UPRS(B,NM,NUM,NX,NX,T4HA)
CALL UPRS(C,NM,NX,NM,NX,T4HC)
CALL UPRS(D,NM,NUM,NM,NX,T4HD)
CALL UPRS(E,NM,NM,NM,NX,T4MK)
RETURN
END

Figure 107. Subroutine DLSA Program Listing (Concluded)
Enter

Read the Names of the System Variables and Print Them

Write the Names of the System Variables on SDSTP File

Compute CO Matrix

Write CO Matrix on SDSTP File

Compute Cl Matrix

Write Cl Matrix on SDSTP File

Write End of File Mark on SDSTP File

Return

Figure 108. Subroutine FINK Flow Chart
SUBROUTINE FINK(A,NX,NM1,NM2,NM3,NM4,NM5)
C
C PURPOSE: TO COMPUTE FREQUENCY DOMAIN REPRESENTATION
C OF STATE SPACE QUADRUPOLE DATA
C
C ANALYSIS - A F KONYA / J K NAENS - THE HONEYWELL INC
C DATE WRITTEN - 1976
C
C SUBPROGRAMS CALLED
C
C ARGUMENTS LIST
C
A INPUT STATE TRANSITION MATRIX
C H INPUT CONTROL INPUT MATRICES
C C INPUT STATE OUTPUT MATRIX
C D INPUT CONTROL OUTPUT MATRIX
C NA-C ARRAY FOR SYSTEM VARIABLES NAMES
C NA INPUT NO OF STATES
C NR INPUT NO OF OUTPUTS
C NU INPUT NO OF INPUTS
C NX MAXIMUM NO OF STATES
C NX-MAXIMUM NO OF OUTPUTS
C NX-MAXIMUM NO OF INPUTS
C NX-MAXIMUM ROW DIMENSION FOR CO AND CI
C NX-MIN MAXIMUM COLUMN DIMENSION FOR CO AND CI
C
D I M E N S I O N C C ( N X M + N M 1 ) , N X M 2 , N X M 3 , N X M 4
D I M E N S I O N C A ( N X M ) , N X M 2 , N X M 3 , N X M 4
D I M E N S I O N C A N ( N X M ) , N X M 2 , N X M 3 , N X M 4
C J3,J4,J5,J6,J7 DATA + N M A X 2 N M A X 2 N M A X 2 N M A X 2 N M A X 2
C 120 CONTINUE PEAD(1P*14)*ICAPD
140 FORMAT(2V4.4)
I F I C A D ( 1 ) = 0 . H E N D I T U R N
I F I C A D ( 1 ) = 1 . F O R W R I T E A N T O 2 . 0
W R I T E ( I N X = 1 7 4 )
S T O P 1 1 1
200 CONTINUE
C
C READ AND WRITE NAMES OF THE SYSTEM VARIABLES
C
R E A D ( 1 P * 3 7 7 ) ( N A M E ( 1 ) , 1 , 1 , N X M )
3 7 0 F O R M A T ( H A L U )
W R I T E ( I N X = 3 7 7 )
3 7 5 F O R M A T ( 1 H 1 ) / / 1 X , A 2 ) / N A M E S O F T H E O U T P U T V A R I A B L E S , / /
W R I T E ( 1 N X = 3 8 6 ) ( N A M E ( 1 ) , 1 , 1 , N X M 1
3 8 0 F O R M A T ( 1 X , A 1 )
W R I T E ( I N X = 3 8 6 )
N A R P I N X = N A R P 1
W R I T E ( 1 N X = 3 8 6 ) ( N A M E ( 1 ) , 1 , 1 , N A R P 1 , N X M )
W R I T E ( J S D N A R P 1 ) ( N A M E ( 1 ) , 1 , 1 , N X M )
C
C COMPUTE CC AND WRITE ON STOP FILE
C
C A L L / F N O ( C C ( N X M , N X M ) , N X M )

Figure 109. Subroutine FINK Program Listing
Figure 109. Subroutine FINK Program Listing (Concluded)
SUBROUTINE MP(K,L+1,J;A)
DIMENSION A(K,L)
DO 1 J=1,L
WRITE(9,5)
5 FORMAT(5H ROW I)
1 WRITE(9,2)(A(I+J),J=1,J)
FORMAT(2X,1H17,4)
RETURN
END

Figure 110. Subroutine MP Program Listing
Figure 111. Subroutine OUTP Program Listing

```fortran
SUBROUTINE OUTP(I+1,J,J,J,Y,I=1)
DIMENSION Y(I+1,Y(I+1)10+10+10+10)
50 FORMAT(I2+9F6.1)
III=0
10 K=I+1
10 M=J+J
IF(Y(1,M) .LT. 9) GOTO 10
III=1
YD[I]=Y(K,M)
JD[I]=M
IF(Y[I].LT.516010 1=1)
WRITE(I+1,Y(I+1)+I+1)
III=0
100 CONTINUE
IF(Y[I].EQ.0) RETURN
WRITE(I+1,Y(I+1)+I+1)
RETURN
END
```
SUBROUTINE POLES(NX,A,MX,RO,M)
DIMENSION A(NX+1),RR(1)
CALL HESEN(NX,A,MX)
CALL ORCALL(MX,A,RO,M,NX)
WRITE(9,6087)
6087 FORMAT(1X,17X,HEIGENVALUES/2X+4X,REAL,9X,9H1,9HAGINARY,9X,13HDAMP)
ING RATIO+5X+9H,FREQUENCY/1)
M=M+1
DO 60 1=1,N
1=2*K-1
OMEGA=SORT(RR(I)+RR(I+1)+RR(I+1)+RR(I+1))
IF(ABS(RR(I+1)) GT .000000001) GO TO 1
WRITE(9,6084) DD(I)
GO TO 883
883 CONTINUE
6084 FORMAT(9X,6084) DD(I),RR(I),RR(I),DELTA*OMEGA
CONTINUE
6085 FORMAT(9X,4F15.8)
RETURN
END

Figure 112. Subroutine POLES Program Listing
Figure 113. Subroutine HESSEN Program Listing
IF(JK.LE.NK) GO TO 45
70 CONTINUE
RETURN
END

HESSE46
HESSE46
HESSE67
HESSE68

Figure 113. Subroutine HESSE Program Listing (Concluded)
SUBROUTINE QRCALL(N,A,P,M,NIN)
INTEGER D
DIMENSION A(N+1),R(J)
N = N+1
ACT = 1.E-7
ITER = 0
M = 0
IF(N.E.1) RETURN
IF(N.E.0,2) GO TO 25
15 DELTA = ACT*ABS(A(N,N))
ACC = ABS(A(N,N-1))
IF(ACC.EQ.0.) GO TO 16
IF(ACT.GT.DELTA) GO TO 25
IF(IT.EQ.25) GO TO 16
IF(ACT.GT.DELTA) GO TO 25
16 M = M+1
R(M) = A(N,N)
P(M) = 0.
17 K = NIN-N+1
ITER = !
N = N+1
20 IF(N.E.1) GO TO 15
IF(N.E.0,2) GO TO 25
IF(N.E.0,1) GO TO 16
R(N+1) = ACT
RETURN:
25 R = A(N-N+1).*R(N-N+1)
DAN = ABS(A(N,N-N+1).*R(N-N+1))
SAN = ABS(A(N,N-N+1).*R(N-N+1))
IF(DAN.LE.ACT*SAN) DAN=0.
DAN=DAN+DAN.*25
C=A(N-N+1).*A(N-N+1)
T=DAN*C
IF(ABS(T).LE.ACT) T=0.
C = 0.*P(ABS(TI))
IF(N.E.2) GO TO 30
26 IF(N.E.0,1) GO TO 30
R(N+1) = 1.
R(M) = C
27 N = N+1
GO TO 17
30 M = M+2
R(M) = 0.
R(M) = 0.
K = NIN-N+1
M = M+2
R(M) = B-C
R(M) = 0.
GO TO 27
50 IF(N.E.0,1) GO TO 60
R(M+1) = H
R(M+2) = C
R(M+3) = B
R(M+4) = C
GO TO 70
55 X = H
Y = H
R(M+5) = X
Figure 114. Subroutine QRCALL Program Listing
R(M+7) = Y
IF(ABS(R(M+5)) .GT. ABS(Y)) GO TO 70
R(M+5) = Y
R(M+7) = X
70 IF(ITER .LE. 0) GO TO 130
X = ABS(R(M+5)) + ABS(R(M+1)) + ABS(R(M+1)) + ABS(R(M+1)) + ABS(R(M+1)) + ABS(R(M+1))
ACC = ABS(R(M+5)) + ABS(R(M+1)) + ABS(R(M+1)) + ABS(R(M+1)) + ABS(R(M+1)) + ABS(R(M+1))
IF(ACC .GT. 1.) X = X/ACC
Y = ABS(R(M+7)) + ABS(R(M+3)) - ABS(R(M+3)) - ABS(R(M+4)) - ABS(R(M+4))
ACC = ABS(R(M+7)) + ABS(R(M+3)) + ABS(R(M+3)) + ABS(R(M+4)) + ABS(R(M+4)) + ABS(R(M+4))
IF(ACC .GT. 1.) Y = Y/ACC
ACC = ABS(R(M+7)) + ABS(R(M+3)) + ABS(R(M+3)) + ABS(R(M+4)) + ABS(R(M+4)) + ABS(R(M+4))
DELTA = MAX1(DELTA, (ACT+ABS(A(N-1:1:N-1))
IF(ACC .GT. DELTA) GO TO 80
IF((X .GE. ACT) .AND. (Y .GE. ACT)) GO TO 26
80 IF((X .GE. ACT) .AND. (Y .GE. ACT)) GO TO 130
K = M + 5
IF((X .GE. ACT) .AND. (Y .GE. ACT)) GO TO 130
IF(X .GE. 1.0) GO TO 120
IF(X .GE. 1.0) GO TO 110
RHO = R(M+3) - R(M+1) - R(M+1) - R(M+8)
SIGMA = R(M+5) + R(M+7)
CONTINUE
A(N) = A(N+1)
CALL ORCALL6(N+1, RHO, SIGMA, D, DELTA)
R = ABS(A(N+1))
A(N) = ABS(A(N+1))
IF(R .GE. ACT) A(N) = A(N)/R
ITER = ITER + 1
DO 105 IT = 1, 4
K = M + 1
105 R(K) = R(K+4)
GO TO 15
110 K = M + 7
120 RHO = R(K) * R(K)
SIGMA = R(K) * R(K)
GO TO 100
130 RHO = 0.
SIGMA = 0.
GO TO 100
CONTINUE
WRITE(9, 700)
700 FORMAT(16H1, 25SMALL EIGENVALUES NOT FOUND)
RETURN
END

Figure 114. Subroutine QRCALL Program Listing (Concluded)
SUBROUTINE OR (N,A,PHI,SIGMA,D,DELTA)
DIMENSION A(N)
REAL *APPDA
INTEGER P,Q,R
EQUIVALENCE (P,Q)

10 = 0
N0 = N-1
N1 = N0-1
N2 = N1-1
N3 = N2-1
IF(N, N1, 3) GO TO 5
IF(N, N2, 2) RETURN

2 \text{ D = 1}
3 \text{ G TO 35}

5 \text{ I = N1+1}
7 IF(A(I), I, T, D, P, A) GO TO 10
IF(I, L, E, 2) GO TO 2
1 \text{ J = 1-10}
2 \text{ G TO 7}

10 \text{ G = 1+0}
11 \text{ A(I) = G,}
35 \text{ J = P}
37 \text{ \text{ I = 0}
38 \text{ I0 = 1-0}
40 \text{ I = 1+0}
42 \text{ I = 1+0}
44 \text{ G1 = (I1*(A(I)+\text{S*(PHA)}=A(I)+A(I)+1)+RNI)
46 \text{ G2 = A(I)*A(I)+A(I)+1)+S*(PHA)
48 \text{ G3 = (I1*(A(I)+1)+A(I)+1)+S*(PHA)
50 \text{ A(I+2) = 0}
52 \text{ G0 TO 45}
54 \text{ G1 = (I1}
56 \text{ G2 = A(I+1)
58 \text{ G3 = '}
60 \text{ I0 = 10+0}
62 \text{ IF(I, L, E, N2) G3 = A(I+2)}
64 \text{ KAPPA = SQRT(A(I)+02+G2+G3+G3)}
66 \text{ IF(G1, L, T, 1, 1) KAPPA = -KAPPA}
68 \text{ IF(KAPPA, N, E, 0, 2) G0 TO 47}
70 \text{ ALPHA = 2}
72 \text{ P1 = '}
74 \text{ P2 = '}
76 \text{ G0 TO 48}
78 \text{ ALPHAE = 1+G1/KAPPA}
80 \text{ P1 = 1/G(A,KAPPA)}
82 \text{ P2 = P1*G3}
84 \text{ P3 = P1*G2}
86 \text{ IF(I, L, E, N3) GO TO 49}
88 \text{ A(I) = -A(I)}
90 \text{ IF(I, L, E, P, A)**(I) = -KAPPA}
92 \text{ J = 1-0}
94 \text{ J = J+0}
96 \text{ IF(J, L, E, N3) GO TO 51}
98 \text{ ETA = A(J)+P1*A(J+1)}
100 \text{ IF(I, L, E, N2) ETA = ETA+P2*A(J+2)}
102 \text{ ETA = ALPHAE*ETA}
104 \text{ A(J) = A(J)-ETA}
106 \text{ A(J+1) = A(J+1)-P1*ETA}
108 \text{ IF(I, L, E, N2) A(J+2) = A(J+2)-P2*ETA}
110 \text{ G0 TO 56}
112 \text{ J = P1-1}
114 \text{ JINX = MING(I+1, N1+1)
116 \text{ J = J+1}

Figure 115. Subroutine QR Program Listing
Figure 115. Subroutine QR Program Listing (Concluded)
SUBROUTINE INPTM(A,II,JJ,IR)

PURPOSE - TO READ NONZERO ELEMENTS OF A MATRIX FROM FILE IN
ANALYSIS - A F KONAR / J K VAMESH - TIE HONEYWELL INC
DATE WRITTEN - 1976

ARGUMENTS LIST
A     INPUT  MATRIL DATA
II    INPUT  MAXIMUM NO OF ROWS
JJ    INPUT  MAXIMUM NO OF COLUMNS
IR    INPUT  FILE NO FOR READING MATIRIA DATA

DIMENSION A(II+JJ+1,D(5)+J(5)+YD(9))
2 FORMAT (5(2F2,13,5))
1 READ(II,JJ,1D(L),JM(L)+YD(L)+L=1+5)
IF (EOF(I))10+6
6 CONTINUE
IF (I=11)3+10+3
3 DO 5 I=1+5
IF (I=11)4+1+4
4 I=I+1(L)
J=J+1(L)
5 A(I,J)+YD(L)
GO TO 1
10 CONTINUE
RETURN
END

Figure 116. Subroutine INPTM Program Listing
SUBROUTINE WTP(M,N,A,JM,NJ)

PURPOSE - TO WRITE NUMERICAL ELEMENTS OF A MATRIX ON A FILE

ARGUMENTS LIST

M INPUT MATRIX DATA
N INPUT NO. OF ROWS
A INPUT MAXIMUM NO. OF COLUMNS
J INPUT MAXIMUM NO. OF COLUMNS
JX INPUT FILE NO. FOR WRITING DATA

DIMENSION A(N+1,N+1),JX(NJ+1)
INT HAS CARD
IF (NR,F4,16) GO TO 10
IF (NC,F3,17) GO TO 10
1 I=1
DO (A(I,J)=0, J=1,N) GO TO 20
II=II+1
AN[I]=A(K,M)
JN[I]=K
JQ[I]=M
IF (I.EQ.LT.S) GO TO 30
WRITE (6,2) [11], JD(L) + AD(L) * L = [11]
II=0
2 CONTINUE
10 CONTINUE
WRITE (6,2) [11], JD(L) + AD(L) * L = [11]
100 CONTINUE
WRITE (6,2) [11], ILAN=K
DO 111 [11]=1,2
111 CONTINUE
WRITE (6,12) [11], READ(I1) [11]=1,2
120 CONTINUE
RETURN
END

Figure 117. Subroutine WTP Program Listing
FUNCTION GRAN(N)

K=N
IF (N .EQ. 0) GO TO 1
ISEED = 33973A3900
X=ISEED
TEM=RANF(X)
X=0,
1 TEM = 0.4
DO 2 I = 1,12
2 TEM = TEM*RANF(X)
TEM = TEM - N,0
GRAN = TEM
RETURN
END

Figure 118. Subroutine GRAN Program Listing
Enter

Read the Program Used and the Maximum System Dimensions

Calculate Maximum Scratch Array Dimension Needed

Compute Total Memory (Field Length) Required to Execute KONPACT-1 Program and Print It in Octal

Write Main Program for KONPACT-1 Program on COMP File

Stop

Figure 119. Program PRECOM Flow Chart
APPENDIX

PRECOMPILER PROGRAM FOR KONPACT-1

The precompiler program performs the task of writing the MAIN program for KONPACT-1. A brief description of the precompiler program is presented in this section.

The precompiler program reads the system dimensions and the KONPACT-1 program names and computes the maximum sizes of the scratch arrays. It writes the MAIN program for KONPACT-1 on file COMPIL. The flow chart is given in Figure 119 and the program listing is given in Figure 120.
PROGRAM PRECOM(INPUT=OUTPUT,COMP=TAPES=INPUT,TAPE=OUTPUT)
1:TAPE=COMP)
C
ANALYSIS = A & K KONAR / J R MANESH = THE HONEYWELL INC
C
PURPOSE = TO READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM
C
DIMENSIONS AND SET UP THE MAIN PROGRAM FOR KONPART=1 PROGRAMS
C
DATE = WRITTEN = DECEMBER 1975
C
DIMENSION CARD(20)
DATA NAME,NUMHE,NNAME,NHYME/AMNHE,AMNRM,AMNM,NHNM,ANHYM/
DATA MS5EM,MSHE,MCH,MRAB,AMSH,AMTH,MC,AMH/
DATA MRH,MSK,MRB,MRM,AMK / MS1F0,%MS2F0,%MS3F0,%MS4F0,%MS5F0
C
INITIALIZE MAXIMUM SYSTEM DIMENSIONS
C
NAME % NUMHE % NUMHE % NUMHE % MSTR % MTRK
C
READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM DIMENSIONS
C
CONTINUE
READ(120)CARD
120 CONTINUE
IF (EOF(120)) 140
140 CONTINUE
CODECARD(4,160)CARD(1)CC=Dummy
160 CONTINUE
IF (CC.EQ.MARK) GO TO 100
C
SET THE PROGRAM FLAGS
C
CODECARD(2)
IF (CODE.EQ.MARK) MS1F0
IF (CODE.EQ.MARK) MS2F0
IF (CODE.EQ.MARK) MS3F0
IF (CODE.EQ.MARK) MS4F0
IF (CODE.EQ.MARK) MS5F0
IF (CODE.EQ.MARK) MS6F0
IF (CODE.EQ.MARK) MS7F0
IF (CODE.EQ.MARK) MS8F0
IF (CODE.EQ.MARK) MS9F0
C
SET THE MAXIMUM SYSTEM DIMENSIONS
C
CODECARD(1)
CODECARD(4,180)CARD(2)=MAX=Dummy
180 CONTINUE
IF (CODE.EQ.MARK) NHNM=MAX
IF (CODE.EQ.MARK) NHNM=MAX
IF (CODE.EQ.MARK) NHNM=MAX
IF (CODE.EQ.MARK) NHNM=MAX
IF (CODE.EQ.MARK) NHNM=MAX
IF (CODE.EQ.MARK) NHNM=MAX
IF (CODE.EQ.MARK) NHNM=MAX
IF (CODE.EQ.MARK) NHNM=MAX
C
IF DATA CARD IS IN ERROR PRINT ERROR MESSAGE
C
Figure 120. Program PRECOM Program Listing
C

CALCULATE DIMENSIONS WHICH ARE USEFUL TO COMPUTE
MAXIMUM SCRATCH ARRAY DIMENSIONS REQUIRED

220 CONTINUE

C

CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S1
TO USE THE VARIOUS KONPACT=1 PROGRAMS

C

CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S2
TO USE THE VARIOUS KONPACT=1 PROGRAMS

C

CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S3
TO USE THE VARIOUS KONPACT=1 PROGRAMS

C

IF NO SPECIFIC PROGRAMS ARE READ SET ALL PROGRAM FLAGS TO 1

Figure 120. Program PRECOM Program Listing (Continued)
CALCULATE MAXIMUM SCRATCH ARRAY DIMENSIONS NEEDED

221 CONTINUE
IF(MSF, 4, 1, 1) GO TO 221
MSF=1 $ MSF#1 $ MSF#1 $ MSF#1

222 CONTINUE
IF(MSF, 4, 4, 1) GO TO 222
MSF=1 $ MSF#1 $ MSF#1

224 CONTINUE
IF(MSF, 4, 1, 0) GO TO 224
MSF=1 $ MSF#1 $ MSF#1

226 CONTINUE
IF(MSF, 4, 0, 1) GO TO 226
MSF=1 $ MSF#1 $ MSF#1

228 CONTINUE
IF(MSF, 4, 1, 0) GO TO 228
MSF=1 $ MSF#1 $ MSF#1

230 CONTINUE
MS1=MAXI(MS1, MS2, MS3, MS4, MS5)
MS2=MAXI(MS1, MS2, MS3, MS4, MS5)
MS3=MAXI(MS1, MS2, MS3, MS4, MS5)
MS4=1

COMPUTE MEMORY REQUIRED FOR SCRATCH ARRAYS

MS1=MS1+MS2+MS3+MS4

SET THE MEMORY REQUIRED FOR THE PROGRAM CODE

MPT=30000

COMPUTE TOTAL MEMORY REQUIRED TO EXECUTE KONPACT=1 PROGRAM AND PRINT THE FIELD LENGTH REQUIRED IN OCTAL BASE

MPT=MST+MPT
WRITE(9, 240)
WRITE(9, 260)
WRITE(9, 280)
WRITE(9, 300)
WRITE(9, 320)
WRITE(9, 340)

WRITE MAIN PROGRAM FOR KONPACT=1 PROGRAM ON COMP FILE

WRITE(9, 360)
WRITE(9, 380)
WRITE(9, 400)
WRITE(9, 420)
WRITE(9, 440)
WRITE(9, 460)

Figure 120. Program PRECOM Program Listing (Continued)
Figure 120. Program PRECOM Program Listing (Concluded)
REFERENCES


